ISSN:0975-9646

Data Security through Video Embedding

Arushi Shah[1], Nidhi Kapadia[2], Ashwinee Mehta[3], Mrs Kriti Srivastava[4]

Information Technology Department,
Dwarkadas J. Sanghvi College of Engineering, Mumbai University
Plot No. U-15, J.V.P.D Scheme, Vile Parle, Mumbai, India

***Abstract*— In the world we live in today, all of our essential information, right from our health records to our financials, everything is stored by computers and accessed by the internet. All of this data is a sitting duck for hackers and attackers. We aim to create a system that would make this transmission of data secure. For this purpose, we aim to embed the sensitive data file into a video file and transmitting it. The data file will be encrypted, compressed and embedding into the video file in a secure environment. Authorized users can access this file by extracting it from the video file, decompressing and decrypting it. We will embed the sensitive data text file into a video file and compress it using Shannon-fano algorithm. The use of Shannon-fano algorithm significantly improves the performance of our system.**

This intranet-based application provides the user of the system a centralized view of the things maintained in the software depending on the privileges assigned by the customer accordingly. With the use of advanced computer software, authors of images and software can place a hidden trademark in their product, allowing them to keep a check on piracy. Together, these two are intended to fight piracy. The latter can be used to detect copyright violators and the former can be used to prosecute them. The cover data will not be significantly degraded by the embedded data, and the embedded data will be as imperceptible as possible. The embedded data is as immune as possible to modifications from intelligent attacks or anticipated manipulations. Thus it is necessary that the hidden message should be encrypted. In this way, we aim to make the transmission of data more secure than it currently is.

***Keywords*— Data Security, Embedding, Steganography, Enhanced Tiny Algorithm**

I THE NEED FOR DATA SECURITY

The capability to send a message electronically from one computer system to another computer system so that only the intended recipient receives and reads the message and the message received is identical to the message sent is called transmission security. The message if it was altered in anyway, would not be identical to the original message, whether transmitted over faulty channels or intercepted by an eavesdropper. Transmission security translates into secure networks.

Unfortunately, all transmissions can be interrupted but the cautious user looks at all transmissions as if they will be intercepted. You can minimize the risks of transmission interception, but you can never, under any circumstances, completely rule it out. It is ironically, the people who design and put wires in their place who are the people that can get to them.

Transmission interceptions are an inevitability; it's likely they will occur at times. Designing a 100 percent transmission-secure network is akin to designing a car that can't be broken into; no matter how secure the car is, someone can always break the windows. This doesn't mean that we should sit back and wait for the interception, however; instead, we aim to build this system with the intention of deterring people from attempting to break in, and making it costly for the hacker to enter. [1]

A. Type of Data Security Risks

Any computer with access to a physically connected network or in the vicinity of over-air transmissions could be instructed not to ignore the signals intended for other computers. This is the essence of electronic eavesdropping. Information is considered intercepted when someone other than the intended recipient receives the information. Data can be intercepted in so many ways. Electronic eavesdropping or using the recipient's password are common examples and it can occur anywhere, including in a chat room or through an e-mail exchange.

B. Sniffing Devices

There are a few troubleshooting programs and devices designed to analyse LAN traffic. These are commonly referred to as *packet sniffers*, because they are created to "sniff" packets of data for the network engineer. Unfortunately, there are also users out there maliciously using packet sniffers to read data worldwide, continuously.

C. Snooping devices

Spoofing means making your computer pretend it is a different computer. The user forces the computer to present credentials to the network that are false. To do so, the user doesn't need tools but rather information to make those credentials realistic. By doing so, the user can gain unauthorized access to sensitive data and thus compromise it. These are generally difficult attacks to carry out because of how information is transmitted from computer to computer. [2]

II EXISTING DATA SECURITY MODELS

The information age that we currently belong to relies heavily on the transfer of data between computers, tablets, mobiles and other communication devices. Convenient and effective data transfer relies on standardized data formats, such that different users using very different equipment can communicate with each other. In order to enable accurate data transmission over large distances, data is digitized, text is encoded in ASCII, documents are formatted in rich text format, and other standardized measures are used to ensure maximum efficiency and effectiveness, so that the transmitted data can be fully reproduced between different users, even when they are using different communication equipment.

Some data, like that on many websites, academic databases and libraries are meant to be accessible to anyone, and are considered as being in the public domain, but some access, particularly commercial use, may require payment, such as copyright royalties. Other data is considered private or confidential, and although controlled, easy, cross-platform transmission to specific parties is desirable; it is necessary to protect such data from prying eyes. This may be because of a multitude of reasons; the data having a personal nature, to protect patient privacy, client-attorney privilege or because the data may be pertinent to issues of national security.

A way to protect data files during transmission, is to use some type of encryption. Encryption is the process of changing text in such a way that it is no longer easy to read. Non-encrypted data have been compared to open books, since they may be read by anyone. With encryption however, only the intended recipient will be able to open and read the message, and many types of encryption are known.

Almost all modern encryption methods rely on a 'key', which is a random or fixed number or string of characters used to encrypt, decrypt, or both. One commonly used encryption technique is 'symmetrical' encryption, or 'Private Key' encryption. Both the parties share an encryption key, and the encryption key and the decryption key are identical. The key is used by the sender to lock data prior to its transmission, and the recipient requires knowledge of the key to open the message on its receipt. One difficulty is sharing the key, i.e. safely transmitting it to recipient. Generally, for convenience and to help both sender and recipient remember the encryption key, a meaningful number or letter string is used, such as the name of a relative, a famous person or pet, the title of a song or a phone number. This tendency does however somewhat limit the effectiveness of such symmetrical keys, since easily remembered or meaningful keys are more easily broken.

When each of the parties use a different key, it becomes essential to store the keys in a list or database, which is, itself, a security risk. To overcome the problem of remembering or securing a long list of keys, a group of users, such as all members of a corporation may use the same encryption key. The consequence of grouping users in this manner is that to enable encrypted communication between all group-members, each member is only requires to remember one key. However, grouping users in this manner entails a security risk in that once security is breached all data transfer between all group members is insecure. One threat to data security is gifted computer hackers, but another threat is simply that an individual may simply cease to be a member of the group. If the contract of an employee of a corporation is terminated, for example, to provide adequate protection of data transmission between members of the corporation it may be necessary to change all passwords and encryption keys. This will be critical if such a former employee goes to work for a competitor, for example. Disseminating new encryption keys in a secure manner is itself, not trivial.

A better method is called asymmetrical encryption, otherwise known as 'public key encryption'. It works by using a combination of two keys: a 'private key' and a 'public key', which together form a pair of keys.

The sender requests the intended recipient for the public key, encrypts the message, and sends the encrypted message to the intended recipient. Only the intended recipient can then decrypt the message—even the original sender cannot read the message to be sent once it is encrypted. The private key is kept secret on the recipient's computer since it is used for decryption, whereas the public key, which is used for encryption, is given to anybody who wants to send encrypted mail to the intended recipient. Thus in public key encryption, only the intended recipient's private key can unlock the message encrypted with the corresponding public key thereof. When a sender wishes to share a secret with an intended recipient using public key encryption, he first asks the intended recipient for his public key. Next, sender uses the intended recipient's public key to encrypt the message. The sender sends message to the intended recipient. The intended recipient uses his private key to decrypt sender's message. Public key encryption works if the intended recipient guards his private key very closely and freely distributes the public key.

The sender's encryption program uses the intended recipient's public key in combination with the sender's private key to encipher the message. When recipient receives Public-Key encrypted mail, he uses his Private Key to decipher it. Decryption of a message enciphered with a public key can only be done with the matching private key. The two keys form a pair, and it is most important to keep the private key safe and to make sure it never gets into the wrong hands, that is, any hands other than those of recipient.

Public key encryption is only safe and secure if the sender of an encrypted message can be sure that the public key used for encryption belongs to the intended recipient. A malicious user impersonating the intended recipient can produce a public key with the recipient's name and give it to the sender, who uses the key to send important information in encrypted form. The enciphered message is intercepted by the third party, and since it was produced using their public key they have no problem deciphering it with their private key, and in this manner credit card data may be obtained fraudulently, for example. Therefore, it is mandatory that a public key is either personally given to the sender by the recipient, or is authorized by a certificate authority.

Certification of public keys in this manner requires support resources and is costly. Since the private key of a certified asymmetrical encryption key is typically a long string of random digits or letters, it cannot be remembered by user, and it is impractical to type out each time. Consequently, such private keys are stored on their owner's computer. Computer failure, due to viruses or mechanical failure for example, often results in the private key being irretrievably lost. Since the private key is stored on hard disk of recipient, it is far from immune to hackers. Loss of the private key makes encrypted messages unreadable and is both costly and inconvenient to replace.

Cryptanalysis, or the process of attempting to read the encrypted message without the key, is very much easier

with modern computers than it has ever been before. Modern computers are fast enough to allow for 'brute force' methods of cryptanalysis—or using every possible key in turn until the 'plain text' version of the message is found.
With the advancement in technology, the rate of cybercrime is also rising and so is the increase in threat to personal information. Our system aims to control and keep this threat in check.

A. Disadvantages of the Existing Data Security Models

The people who break cryptographic systems don't follow rules; they cheat. They attack a system using techniques the makers of the system probably never thought of. Art thieves have burgled homes by cutting through the walls with a chain saw. Home security systems, no matter how expensive and sophisticated, won't stand a chance against this attack. Information thieves breach these walls too. They steal technical data, bribe insiders, modify software, and collude. The odds favor the attacker: defenders have to protect against every possible vulnerability, but an attacker only has to find one security flaw to compromise the whole system.
The present-day data security is a house of cards; it may stand for now, but it can't last. Many insecure products have not yet been broken because they are still in their infancy. But when these products are widely used, they will become tempting targets for criminals. The press will publicize the attacks, undermining public confidence in these systems. Ultimately, products will win or lose in the marketplace depending on the strength of their security.
The effectiveness of symmetrical keys, ones that are made easy to be easily memorized, is limited, since easily remembered or meaningful keys are more easily broken.
In case of asymmetrical keys, when multiple keys are used by both the end users, it is a necessity to store the keys in a list or database, which creates a major security loophole. [3]
To overcome the problem of remembering or securing a long list of keys, a group of users, such as all members of a corporation may use the same encryption key. This causes the problem of multiple people, some of who could have malicious intent, having access to the same data, making it unsafe.
There is no proper authentication system that checks the authority of the user before allowing it access to the data.

III THE PROPOSED SYSTEM

Our system aims at providing proper security and authentication for using this application. A user ID and password is given to login and access any feature throughout this application. Authorized users can hide important document inside a video file. The files are also compressed before embedding so as to save the storage space of the system. The main concern of this project is to create an environment for the secure transmission of sensitive information in an efficient manner.

A. Advantages Of The Proposed System

It encrypts, compresses and embeds the sensitive text into a video file. The video file acts as a sealed envelope for the data that will only be read by the intended recipient.
It can provide security to any text document present in our system.
The audio or video can be played just as any other ordinary audio/video file. Thus the attacker would be unaware of the true nature of our video file, and unaware of the data it hides.
Audio can be split or joined in two or more parts.
File's size can be reduced to a great extent, thus making transmission simple.
The system would contain the following modules:

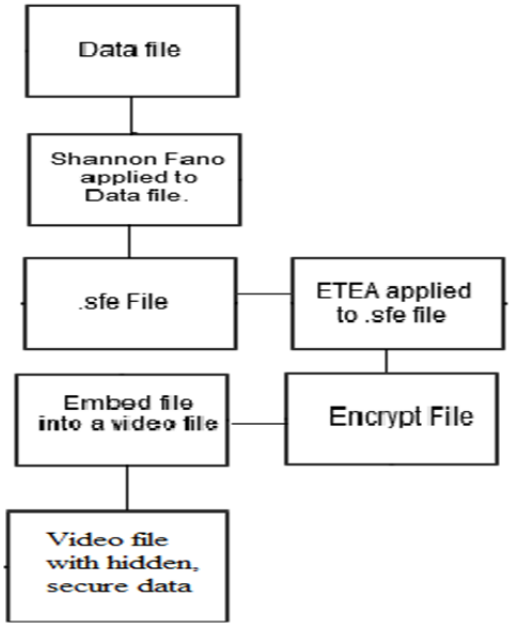

Fig. 1 Model of our System

Module 1: Login Module
In this module, the user has to enter the correct password and name. He will then be authorized to use this application, if his credentials prove false, he will not be granted access. This enhances the security of this project.

Module 2: Audio-Video Module
In this module the new user can view an audio file along with its details such as audio file name, duration, time etc. In this user can also view the video files with their details.

Module 3: Compression-Decompression Module
In this module, the user can compress the file so as to reduce the file size. It also gives the amount of compression done. File is saved with '.sfe' extension. User can also decompress the file to its original form without any loss of data.

Module 4: Embedding De-embedding Module
In this module, user can hide the important document by embedding it into any video. User can also de-embed the file from video back to its original form.

Module 5: Split and Joining Module
In this module the user can split the audio file into maximum 3 parts and can also combine more than one media file together.

IV ALGORITHMS USED

A. *Shannon fano algorithm*

This algorithm is best explained with an example. To create a code tree according to Shannon and Fano an ordered table is required providing the frequency of any symbol. Each part of the table will be divided into two segments. The algorithm has to ensure that the upper and the lower part of the segment have nearly the same sum of frequencies. This procedure will be repeated until only single symbol is left.
Utilization of Shannon-Fano coding makes primarily sense if it is desired to apply a simple algorithm with high performance and minimum requirements for programming. An example is the compression method implode as specified e.g. in the ZIP format. [7]

Symbol Length	Frequency	Code Length	Code	Total
A	24	2	00	48
B	12	2	01	24
C	10	2	10	20
D	8	3	110	24
E	8	3	111	24

Fig. 2 Example of Shannon-Fano Algorithm

Total Symbols: ∑(frequency length) = 62 symbols

SF coded Symbols: ∑(frequency length*code length)=140 Bit
Linear (3 Bit/Symbol): 3*∑(frequency length) = 186 Bit

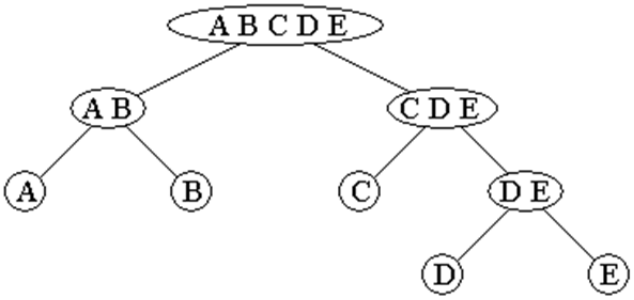

Fig. 3 Shannon Fano Bit Code Tree

The original data can be coded with an average length of 2.26 bit. Linear coding of 5 symbols would require 3 bit per symbol. But before generating a Shannon-Fano code tree the table must be known or it must be derived from preceding data.
In the field of data compression, Shannon–Fano coding, named after Claude Shannon and Robert Fano, is a technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). The algorithm produces fairly efficient variable-length encodings; when the two smaller sets produced by a partitioning are in fact of equal probability, the one bit of information used to distinguish them is used most efficiently. [9]

B. *Enhanced tiny algorithm (ETEA)*

In this paper, we have proposed Enhanced TEA (Tiny Encryption Algorithm) with embedding (ETEA) using input output packages of Java[4][5][6]. In TEA, only encryption was possible. But in our proposed algorithm ETEA, encryption and embedding both are combined to provide high level of security to the data so that it couldn't be hacked.

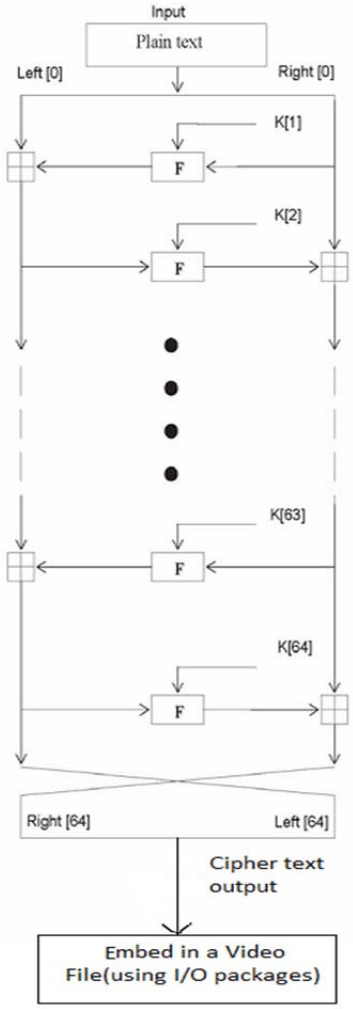

Fig. 4 Encryption and Embedding

The encrypted data using TEA is embedded in a video file using steganography [7] and input/output packages. This file can be transferred through network with high security to another user. Receiver can de-embed the video file and decrypt the original data using same key used at the time of encryption.
The following notations are necessary here:
Hexadecimal numbers will be subscripted as "h" e.g., 10 = 16. Bitwise Shifts: The logical left shift of x by y bits is denoted by x << y. The logical right shift of x by y bits is denoted by x >> y.
Bitwise Rotations: A left rotation of x by y bits is denoted by x <<< y. A right rotation of x by y bits is denoted by x >>> y.

Exclusive-OR: The operation of addition of n-tuples over the field (also known as 2F exclusive-or) is denoted by x⊕y. [8]

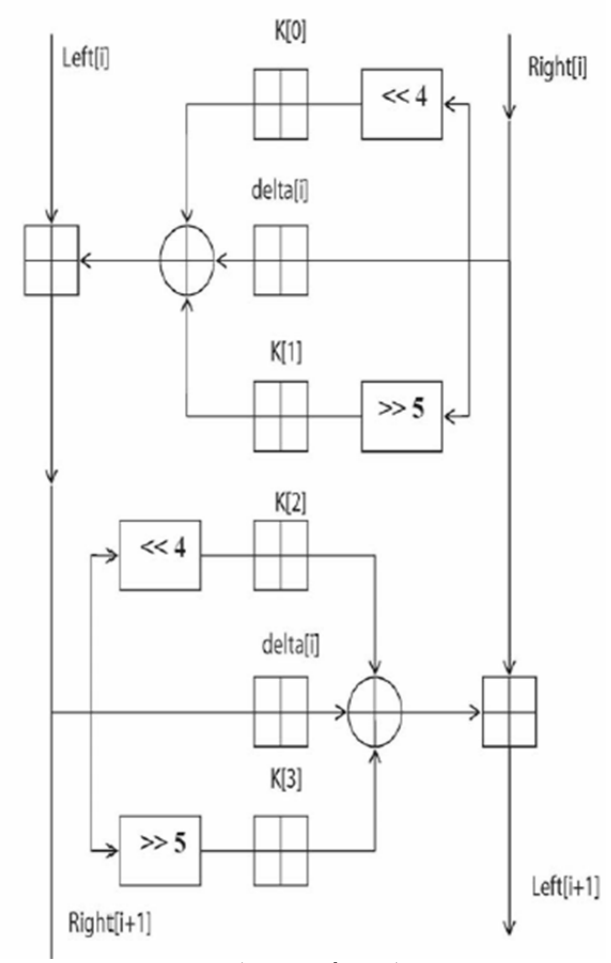

Fig. 5 Round Function

The Enhanced Tiny Encryption Algorithm is a Feistel type cipher that uses operations from mixed algebraic groups. A dual shift causes all bits of the data and key to be mixed repeatedly.

The key schedule algorithm is simple; the 128-bit key K is split into four blocks of 32-bits each K = (K [0], K [1], K [2], K [3]). In a Feistel cipher, the text being encrypted is split into two halves. The round function, F, is applied to one half using a sub key and the output of F is (exclusive-or-ed (XORed)) with the other half. The two halves are then swapped. Each round follows the same pattern except for the last round where there is often no swap.

Each round i has inputs Left[i-1] and Right[i-1], derived from the previous round, as well as a sub key K[i] derived from the 128 bit overall K.

The sub keys K[i] are different from K and from each other. The constant delta = (51/2-1)*231 =9E3779B, is derived from the golden number ratio to ensure that the sub keys are distinct and its precise value has no cryptographic significance.

The round function differs slightly from a classical Fiestel cipher structure in that integer addition modulo 2^{32} is used instead of exclusive-or as the combining operator.

The cipher text as output is then embedded in a video file using Input/output packages of Java.

Fig. 5 presents the internal details of the ith cycle of ETEA. The round function, F, consists of the key addition, bitwise XOR and left and right shift operation. We can describe the output (Left [i +1], Right [i +1]) of the ith cycle of ETEA with the input (Left[i], Right[i]) as follows:

Left [i+1] = Left[i] F (Right[i], K [0, 1], delta[i])

Right [i +1] = Right[i] F (Right [i +1], K [2, 3], delta[i])

Delta[i] = (i +1)/2 * delta

The round function, F, is defined by

F (M, K [j,k], delta[i]) = ((M << 4) K[j]) ⊕ (M delta[i]) ⊕ ((M >> 5) K[k]).

The key schedule algorithm is simple; the 128-bit key K is split into four 32-bit blocks K = (K [0], K [1], K [2], K [3]). The keys K [0] and K [1] are used in the odd rounds and the keys K [2] and K [3] are used in even rounds. [10][11]

The embedded message is de-embedded from video file and then cipher text is taken as input to decryption process. Decryption is essentially the same as the encryption process; in the decode routine the cipher text is used as input to the algorithm, but the sub keys K[i] are used in the reverse order. The intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped.

C. Embedding inside video

Data which hold effective information often has some redundancy. End users usually tend to think that redundancy is evil which costs extra money, as more disk space or network bandwidth is needed. Well, they are partially right, but optimal compression hardly ever exists. Moreover common compression ratio is mostly a question of efficiency. Now we know, there are almost always few bytes, one can play with, without destroying carried information. When information is hidden inside video the program or person hiding the information will usually use the DCT (Discrete Cosine Transform) method.

DCT works by slightly changing the each of the images in the video, only so much though so it's isn't noticeable by the human eye. To be more precise about how DCT works, DCT alters values of certain parts of the images, it usually rounds them up. For example if part of an image has a value of 6.667 it will round it up to 7.

V CONCLUSION

In this project, we aim to create a system which can enhance the existing data security practices by allowing the transmission of data in an extremely secure manner. We first compress the text file using Shannon Fano algorithm. Then we embed the file into a video. For doing this we used Enhanced Tiny algorithm (ETEA). Although the key size for ETEA is less but it has more number of cycles which makes it better providing the highest security. The major feature of ETEA is that it is both secure and more resistant to attacks. In this project, we also introduced two additional

features of background recording of activity and splitting-joining of audio file. Therefore, this altogether makes it a complete and wholesome system with high security and space management features. While data can never be completely secure because as the sensitivity of the data increases, the bounty on its head does too and a larger number of people will try their best to access it. We can however, slow down this process and make it much more difficult for them to perform this task, we can provide users with the satisfaction that their data is safe and in this day and age with privacy and data security being such an elusive quality, this is imperative.

REFERENCES

[1] Toubiana, Vincent, and Serge Papillon. "Secure data transmission." U.S. Patent Application 14/364,605.

[2] Security: Secure Internet Data Transmission http://www.windowsecurity.com/whitepapers/misc/Security_Secure_Internet_Data_Transmission.html

[3] Cohen, Ram, and Meir Zorea. "Secure Data Transmission." U.S. Patent Application 11/991,527.

[4] Herbert Schildt, Java: The Complete Reference, 7th edition, TMH.

[5] Fabien A.P., and Petitcolas, "Information Hiding: Techniques for Steganography and Digital Watermarking.", IEEE conference of digital processing, China,2012,pp 112-121.

[6] Behrouz A. Forouzan, (2006)―Cryptography and Network Security‖, Firstedition, McGraw- Hill.

[7] Ziv. J and Lempel A., "A Universal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory 23 (3), pp. 337–342, May 1977.

[8] Dr. Deepali Virmani, Nidhi Beniwal, Gargi Mandal, Saloni Talwar,"Enhanced Tiny Encryption Algorithm".

[9] Pasco.R., "Source coding algorithms for fast data compression", Ph.D thesis, Department of Electrical Engineering, Stanford University, 1976.

[10] Vikram reddy andem , "A cryptanalysis of the tiny encryption algorithm," Department of Computer Science, The University of Alabama, 2003.

[11] L. Peter Deutsch , "DEFLATE Compressed Data Format Specification," version 1.3, 1996.