A Novel Steganographical Approach to Text Message Hiding using Break Security Method

Mr.Baddepaka Prasad[1], Ms.Thodeti Spandana[2] , Mr.P.Pavankumar [3]
Department of CSE
[1, 2] KMIT, Narayanguda
[3]Research Scholar, Madhav University

Abstract*-Hiding data that is transferring in unsafe channel is an important issue in communication. Steganaography methods present a solution for increasing the data security in such conditions. In this paper, we are going to introduce an algorithm for embed-ding a message in a RGB 24-bit color image. Due to this, we scheme the image and place the message using a linked list like approach. Also LSB (Least Significant Bits) technique is got to work. Proposed method has many benefits that will be talked, Such as: making the hidden message detection harder (because of spreading message blocks irregularly in the image), creating a stego key for extracting message and flexibility in implementation.*

Keywords-Steganography, secure communication, data covering, carrier image, linked list, LSB

I. INTRODUCTION

On line facilities are closely tied with the issues concerning availability, integrity, confidentiality and authentication of information exchanged over communication media, which has lead to the evolution of information hiding techniques for securing communication

To do this, one of the common strategies is using steganography algorithms. The word steganography is derived from the Greek words “stegos” meaning “cover” and “graphy” meaning “writing” defining it as “covered writing” .Steganography is one such pro-security innovation in which secret data is embedded in a cover In other words, steganography is the process of hiding a secret message within a larger one in such a way that someone cannot know the presence or contents of the hidden message. Although related, Steganography is not to be confused with Encryption, which is the process of making a message unintelligible - Steganography attempts to hide the existence of communication .The notion of data hiding or steganography was first introduced with the example of prisoners' secret message by Simmons in 1983.

In steganography we are faced with two types of components: message and carrier. Message is the secret data which should be hidden; and carrier is the context that hides the message in it. Carrier can be of any types of data such as text, image, audio, etc. Message with embedded hidden information is called “stego-text” .

In this paper, we are going to hide a binary message in an image as the carrier material which we call it “cover image”. Message will be embedded sporadically with a structure like linked list, and random locations of its data blocks. By this, we are going to achieve two important goals:

a) Make the detection of message harder to gain to stricter security.
b) Create a security key for extracting message. Since the head of the message has a random location in the cover image, so the initial address of it can be used as a key.

For embedding the message in a 24-bit RGB image, we use the LSB (Least Significant Bit) technique. So, in the second section which is titled “Background” we will talk about basic concepts about RGB color space and the LSB technique fundamentals. In the third section that is named “Linked list structured message embedding”, we will get into the structure of message and the number of pixels needed to store it. Main part of this section is devoted to embedding algorithm. In the fourth section, titled as “Discussion on features” we are going to talk about the characteristics, advantages and uses of this technique. Finally, “Conclusion” is placed in the end of paper.

1.1 STEGANOGRAPHY

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word steganography is of Greek origin and means "concealed writing" from the Greek words steganos (στεγανός) meaning "covered or protected", and graphei (γραφή) meaning "writing". The first recorded use of the term was in 1499 by Johannes Trithemius in his Steganographia, a treatise on cryptography and steganography disguised as a book on magic. Generally, messages will appear to be something else: images, articles, shopping lists, or some other covertext and, classically, the hidden message may be in invisible ink between the visible lines of a private letter.

The advantage of steganography over cryptography alone is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

1.2 HISTORY The first recorded uses of steganography can be traced back to 440 BC when Herodotus mentions two examples of steganography in his Histories.Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand.

In his work "Polygraphiae" Johannes Trithemius developed his so-called "Ave-Maria-Cipher" with which one can hide information in a Latin praise of God. "Auctor Sapientissimus Conseruans Angelica Deferat Nobis Charitas Potentissimi Creatoris" for example contains the concealed word VICIPEDIA

1.3 TECHNIQUES

1.3.1 PHYSICAL:

Steganography has been widely used, including in recent historical times and the present day. Possible permutations are endless and known examples include:

Hidden messages within wax tablets — in ancient Greece, people wrote messages on the wood, then covered it with wax upon which an innocent covering message was written.

Hidden messages on messenger's body — also used in ancient Greece. Herodotus tells the story of a message tattooed on the shaved head of a slave of Histiaeus, hidden by the hair that afterwards grew over it, and exposed by shaving the head again. The message allegedly carried a warning to Greece about Persian invasion plans. This method has obvious drawbacks, such as delayed transmission while waiting for the slave's hair to grow, and the restrictions on the number and size of messages that can be encoded on one person's scalp. In the early days of the printing press, it was common to mix different typefaces on a printed page due to the printer not having enough copies of some letters otherwise. So a message could be hidden using 2 (or more) different typefaces, such as normal or italic, on a page of type.During World War II, the French Resistance sent some messages written on the backs of couriers using invisible ink

1.3.2 DIGITAL:

Modern steganography entered the world in 1985 with the advent of the personal computer being applied to classical steganography problems. Development following that was slow, but has since taken off, going by the number of "stego" programs available: Concealing messages within the lowest bits of noisy images or sound files. Concealing data within encrypted data or within random data. The data to be concealed are first encrypted before being used to overwrite part of a much larger block of encrypted data or a block of random data (an unbreakable cipher like the one-time pad generates cipher texts that look perfectly random if one does not have the private key).Chaffing and winnowing. Mimic functions convert one file to have the statistical profile of another. This can thwart statistical methods that help brute-force attacks identify the right solution in a cipher text-only attack.

Image of a tree with a steganographically hidden image. The hidden image is revealed by removing all but the two least significant bits of each color component and subsequent Normalization. The hidden image is shown below.

Injecting imperceptible delays to packets sent over the network from the keyboard. Delays in key presses in some applications(Telnet or remote desktop software) can mean a delay in packets, and the delays in the packets can be used to encode data.Changing the order of elements in a set.Content-Aware Steganography hides information in the semantics a human user assigns to a datagram. These systems offer security against a non-human adversary/warden.Blog-Steganography. Messages are fractionalized and the (encrypted) pieces are added as comments of orphaned web-logs (or pin boards on social network platforms). In this case the selection of blogs is the symmetric key that sender and recipient are using; the carrier of the hidden message is the whole blogosphere. Modifying the echo of a sound file (Echo Steganography).Secure Steganography for Audio Signals. Image bit-plane complexity segmentation steganography

Image of a cat extracted from the tree image above

1.4 NETWORK

All information hiding techniques that may be used to exchange steganograms in telecommunication networks can be classified under the general term of network steganography. This nomenclature was originally introduced by Krzysztof Szczypiorski in 2003. Contrary to the typical steganographic methods which utilize digital media (images, audio and video files) as a cover for hidden data, network steganography utilizes communication protocols' control elements and their basic intrinsic functionality. As a result, such methods are harder to detect and eliminate. Typical network steganography methods involve modification of the properties of a single network protocol. Such modification can be applied to the PDU (Protocol Data Unit),to the time relations between the exchanged PDUs, or both (hybrid methods).Moreover, it is feasible to utilize the relation between two or more different network protocols to enable secret communication.

These applications fall under the term inter-protocol steganography.Network steganography covers a broad spectrum of techniques, which include, among others:Steganophony - the concealment of messages in Voice-over-IP conversations, e.g. the employment of delayed or corrupted packets that would normally be ignored by the receiver (this method is called LACK - Lost Audio Packets Steganography), or, alternatively, hiding information in unused header fields.WLAN Steganography – the utilization of methods that may be exercised to transmit steganograms in Wireless Local Area Networks. A practical example of WLAN Steganography is the HICCUPS system (Hidden Communication System for Corrupted Networks)

1.5 DIGITAL TEXT

Unicode steganography uses lookalike characters of the usual ASCII set to look normal, while really carrying extra bits of information. If the text is displayed correctly, there should be no visual difference from ordinary text. Some systems, however, may display the fonts differently, and the extra information would be easily spotted.

Alternately, hidden (e.g., control) characters, and redundant use of markup (e.g., empty bold, underline or italics) can add embedded within a body of text to hide information that wouldn't be visually apparent when displayed, but can be discovered by examining the document source. HTML pages can contain code for extra blank spaces and tabs at the end of lines, as well as different colors, fonts and sizes, which will not be visible when displayed. A more trivial example is white text on a white background, which can be revealed by "selecting".

1.6 USING SUDOKU PUZZELS

This is the art of concealing data in an image using Sudoku which is used like a key to hide the data within an image. Steganography using sudoku puzzles has as many keys as there possible solutions of a Sudoku puzzle, which i671 × 1021. This is equivalent to around 70bitmaking it much stronger than the DES method which uses a 56 bit key

1.7 ADDITIONAL TERMINOLOGY

In general, terminology analogous to (and consistent with) more conventional radio and communications technology is used; however, a brief description of some terms which show up in software specifically, and are easily confused, is appropriate. These are most relevant to digital steganographic systems.

The payload is the data to be covertly communicated. The carrier is the signal, stream, or data file into which the payload is hidden; which differs from the "channel" (typically used to refer to the type of input, such as "a JPEG image"). The resulting signal, stream, or data file which has the payload encoded into it is sometimes referred to as the package, stego file, or covert message. The percentage of bytes, samples, or other signal elements which are modified to encode the payload is referred to as the encoding density and is typically expressed as a number between 0 and 1.

In a set of files, those files considered likely to contain a payload are called suspects. If the suspect was identified through some type of statistical analysis, it might be referred to as a candidate.

1.8 COUNTERMEASURES AND DETECTION

Detection of physical steganography requires careful physical examination, including the use of magnification, developer chemicals and ultraviolet light. It is a time-consuming process with obvious resource implications, even in countries where large numbers of people are employed to spy on their fellow nationals. However, it is feasible to screen mail of certain suspected individuals or institutions, such as prisons or prisoner-of-war (POW) camps. During World War II, a technology used to ease monitoring of POW mail was specially treated paper that would reveal invisible ink. An article in the June 24, 1948 issue of Paper Trade Journal by the Technical Director of the United States Government Printing Office, Morris S. Kantrowitz, describes in general terms the development of this paper, three prototypes of which were named Sensicoat, Anilith, and Coatalith paper. These were for the manufacture of post cards and stationery to be given to German prisoners of war in the US and Canada. If POWs tried to write a hidden message the special paper would render it visible. At least two US patents were granted related to this technology, one to Mr. Kantrowitz, No. 2,515,232, "Water-Detecting paper and Water-Detecting Coating Composition There for", patented July 18, 1950, and an earlier one, "Moisture-Sensitive Paper and the Manufacture Thereof", No. 2,445,586, patented July 20, 1948. A similar strategy is to issue prisoners with writing paper ruled with a water-soluble ink that "runs" when in contact with a water-based invisible ink.

In computing, detection of steganographically encoded packages is called steganalysis. The simplest method to detect modified files, however, is to compare them to known originals. For example, to detect information being moved through the graphics on a website, an analyst can maintain known-clean copies of these materials and compare them against the current contents of the site. The differences, assuming the carrier is the same, will compose the payload. In general, using extremely high compression rate makes steganography difficult, but not impossible. While compression errors provide a hiding place for data, high compression reduces the amount of data available to hide the payload in, raising the encoding density and facilitating easier detection (in the extreme case, even by casual observation).

1.9 APPLICATIONS

Usage in modern printers:

Steganography is used by some modern printers, including HP andXerox brand color laser printers. Tiny yellow dots are added to each page. The dots are barely visible and contain encoded printer serial numbers, as well as date and time stamps.

Example from modern practice:

The larger the cover message is (in data content terms—number of bits) relative to the hidden message, the easier it is to hide the latter. For this reason, digital pictures (which contain large amounts of data) are used to hide messages on the Internet and on other communication media. It is not clear how commonly this is actually done. For example: a 24-bit bitmap will have 8 bits representing each of the three color values (red, green, and blue) at each pixel. If we consider just the blue there will be 28 different values of blue. The difference between 11111111 and 11111110 in the value for blue intensity is likely to be undetectable by the human eye. Therefore, the least significant bit can be used (more or less undetectably) for something else other than color information. If we do it with the green and the red as well we can get one letter of ASCII text for every three pixels. Stated somewhat more formally, the objective for making steganographic encoding difficult to detect is to ensure that the changes to the carrier (the original signal) due to the injection of the payload (the signal to covertly embed) are visually (and ideally, statistically) negligible; that is to say, the changes are indistinguishable from the noise floor of the carrier. Any medium can be a carrier, but media with a large amount of redundant or compressible information are better suited.

From an information theoretical point of view, this means that the channel must have more capacity than the "surface" signal requires; that is, there must be redundancy. For a digital image, this may be noise from the imaging element; for digital audio, it may be noise from recording techniques or amplification equipment. In general, electronics that digitize an analog signal suffer from several noise sources such as thermal noise,flicker noise, and shot noise. This noise provides enough variation in the captured digital information that it can be exploited as a noise cover for hidden data. In addition, lossy compression schemes (such as JPEG) always introduce some error into the decompressed data; it is possible to exploit this for steganographic use as well.

Steganography can be used for digital watermarking, where a message (being simply an identifier) is hidden in an image so that its source can be tracked or verified (for example, Coded Anti-Piracy), or even just to identify an image (as in the EURion constellation).

II. SECURE COMMUNICATION

When two entities are communicating and do not want a third party to listen in, they need to communicate in a way not susceptible to eavesdropping or interception. This is known as communicating in a secure manner or secureCommunication. Secure communication includes means by which people can share information with varyingDegrees of certainty that third parties cannot intercept what was said. Other than spoken face-to-face communication with no possible eavesdropper, it is probably safe to say that no communication is guaranteed secure in this sense, although practical obstacles such as legislation, resources, technical issues (interception and encryption), and the sheer volume of communication serve to limit surveillance.

With many communications taking place over long distance and mediated by technology, and increasing awareness of the importance of interception issues, technology and its compromise are at the heart of this debate. For this reason, this article focusses on communications mediated or intercepted by technology.

Also see Trusted Computing, an approach under present development that achieves security in general at the potential cost of compelling obligatory trust in corporate and government bodies.

2.1 TYPES OF SECURITY

Security can be broadly categorised under the following headings, with examples Hiding the content or nature of a communication Code – a rule to convert a piece of information (for example, a letter, word, phrase, or gesture) into another form or representation (one sign into another sign), not necessarily of the same type. In communications and information processing, encoding is the process by which information from a source is converted into symbols to be communicated. Decoding is the reverse process, converting these code symbols back into information understandable by a receiver. One reason for coding is to enable communication in places where ordinary spoken or written language is difficult or impossible. For example, semaphore, where the configuration of flags held by a signaler or the arms of a semaphore tower encodes parts of the message, typically individual letters and numbers. Another person standing a great distance away can interpret the flags and reproduce the words sent.

Encryption, Steganography, Identity Based, Hiding the parties to a communication – preventing identification, promoting anonymity "Crowds" and similar anonymous group structures – it is difficult to identify who said what when it comes from a "crowd" Anonymous communication devices – unregistered cell phones, Internet cafes, Anonymous proxies, Hard to trace routing methods – through unauthorized third-party systems, or relays, Hiding the fact that a communication takes place, "Security by obscurity" – similar to needle in a haystack, Random traffic – creating random data flow to make the presence of genuine communication harder to detect and traffic analysis less reliable

Each of the three is important, and depending on the circumstances any of these may be critical. For example, if a communication is not readily identifiable, then it is unlikely to attract attention for identification of parties, and the mere fact a communication has taken place (regardless of content) is often enough by itself to establish an evidential link in legal prosecutions. It is also important with computers, to be sure where the security is applied, and what is covered.

2.2 BORDERLINE CASES

A further category, which touches upon secure communication, is software intended to take advantage of security openings at the end-points. This software category includes trojan horses, keyloggers and other spyware. These types of activity are usually addressed with everyday mainstream security methods, such as antivirus software, firewalls, programs that identify or neutralize adware and spyware, and web filtering programs such as Proxomitron and Privoxy which check all web pages being read and identify and remove common nuisances contained. As a rule they fall under computer security rather than secure communications.

2.3 TOOLS USED TO OBTAIN SECURITY

Encryption:

Encryption is where data is rendered hard to read by an unauthorised party. Since encryption can be made extremely hard to break, many communication methods either use deliberately weaker encryption than possible, or have backdoors inserted to permit rapid decryption. In some cases government authorities have required backdoors be installed in secret. Many methods of encryption are also subject to "man in the middle" attack whereby a third party who can 'see' the establishment of the secure communication is made privy to the encryption method, this would apply for example to interception of computer use at an ISP. Provided it is correctly programmed, sufficiently powerful, and the keys not intercepted, encryption would usually be considered secure. The article on key size examines the key requirements for certain degrees of encryption security.

The encryption can be implemented in a way to require the use of encryption, i.e. if encrypted communication is impossible then no traffic is sent, or opportunistically. Opportunistic encryption is a lower security method to generally increase the percentage of generic traffic which is encrypted. This is analogous to beginning every conversation with "Do you speak Navajo?" If the response is affirmative, then the conversation proceeds in Navajo, otherwise it uses the common language of the two speakers. This method does not generally provide authentication or anonymity but it does protect the content of the conversation from eavesdropping.

2.4 METHODS USED TO "BREAK" SECURITY

Bugging

The placing covertly of monitoring and/or transmission devices either within the communication device, or in the premises concerned.

Computers (general)

Any security obtained from a computer is limited by the many ways it can be compromised – by hacking, keystroke logging, backdoors, or even in extreme cases by monitoring the tiny electrical signals given off by keyboard or monitors to reconstruct what is typed or seen (TEMPEST, which is quite complex).

Laser audio surveillance

Sounds, including speech, inside rooms can be sensed by bouncing a laser beam off a window of the room where a conversation is held, and detecting and decoding the vibrations in the glass caused by the sound waves.

2.5 SYSTEMS OFFERING PARTIAL SECURITY

Anonymous cell phones

Cell phones can easily be obtained, but are also easily traced and "tapped". There is no (or only limited) encryption, the phones are traceable – often even when switched off – since the phone and SIM card broadcast their International Mobile Subscriber Identity (IMSI). It is possible for a cell phone company to turn on some cell phones when the user is unaware and use the microphone to listen in on you, and according to James Atkinson, a counter-surveillance specialist cited in the same source, "Security-conscious corporate executives routinely remove the batteries from their cell phones" since many phones' software can be used "as-is", or modified, to enabletransmission without user awareness and the user can be located within a small distance using signal triangulation and now using built in GPS features for newer models. Some cell phones (Apple's iPhone, Google's Android) track and store users' position information, so that movements for months or years can be determined by examining the phone.

Landlines

Analogue landlines are not encrypted, and it is very easy to tap them. Such tapping requires physical access to the line, easily obtained from a number of places, e.g. the phone location, distribution points, cabinets and the exchange itself. Tapping a landline in this way can enable an attacker to make calls which appear to originate from the tapped line.

Anonymous Internet

Using a third party system of any kind (payphone, Internet cafe) is often quite secure, however if that system is used to access known locations (a known email account or 3rd party) then it may be tapped at the far end, or noted, and this will remove any security benefit obtained. Some countries also impose mandatory registration of Internet cafe users.

Anonymous proxies are another common type of protection, which allow one to access the net via a third party (often in a different country) and make tracing difficult. Note that there is seldom any guarantee that the plaintext is not tappable, nor that the proxy does not keep its own records of users or entire dialogs. As a result anonymous proxies are a generally useful tool but may not be as secure as other systems whose security can be better assured. Their most common use is to prevent a record of the originating IP, or address, being left on the target site's own records. Typical anonymous proxies are found at both regular websites such as Anonymizer.com and spynot.com, and on proxy sites which maintain up to date lists of large numbers of temporary proxies in operation.

A recent development on this theme arises when wireless Internet connections ("Wi-Fi") are left in their unsecured state. The effect of this is that any person in range of the base unit can piggyback the connection – that is, use it without the owner being aware. Since many connections are left open in this manner, situations where piggybacking might arise (willful or unaware) have successfully led to a defense in some cases, since it makes it difficult to prove the owner of the connection was the downloader, or had knowledge of the use to which unknown others might be putting their connection. An example of this was the Tammie Marson case, where neighbors and anyone else might have been the culprit in the sharing of copyright files. Conversely, in other cases, people deliberately seek out businesses and households with unsecured connections, for illicit and anonymous Internet usage, or simply to obtain free bandwidth

III. BACKGROUNDS

An image can be represented by a collection of color pixels. The individual pixels are represented by their optical characteristics like “brightness”, “chroma” etc. Each of these characteristics can be digitally expressed in terms of 1s and 0s [7]. There are different color spaces that present different forms for storing images. A color space is a method by which it is possible to specify, create and visualize color [8]. The most common color space among all is RGB (Red, Green, Blue). Each pixel in a 24-bit bitmap image in this space is described by 3 sets of 8 bits (3 bytes), that each set contains the intensity value of individual red, green and blue. Combination of these values forms the characteristics of the pixel. Fig.1 illustrates this matter.

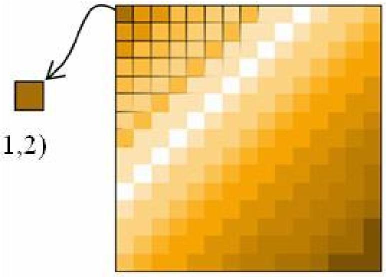


Fig.1 A pixel in RGB color space

Least Significant Bits (LSB) insertion is a simple approach for embedding information in image file. The simplest steganography techniques embed the bits of the message directly in to least significant bit plane of the cover image in a deterministics equence. Modulating the least significant bit does not result in human-perceptible difference because the amplitude of the change is small[9]. To hide a secret message inside an image, a proper cover image is needed. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. When using a 24-bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel. For example, the following grid can be considered as 3 pixels of a 24-bit color image, using 9 bytes of memory:

(0010011111110100111001000)

(0010011111100100011101001)

(1100100000010011111101001)

When the character A, which binary value equals 10000001,is inserted, the following grid results

(0010011**1**1110100**0**1100100**0**)

(0010011**0**1100100**0**1110100**0**)

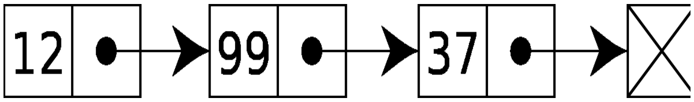
(1100100**0**0010011**1**11101001)

In this case, only three bits needed to be changed to insert the character successfully. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximal cover size. The result changes that are made to the least significant bits are too small to be recognized by the human visual system (HVS), so the message is effectively hidden [10].

As you see, the least significant bit of third color is remained without any changes. It can be used for checking the correctness of 8 bits which are embedded in these 3 pixels. In other words, it could be used as “parity bit”

IV. LINKED LIST STRUCTURED MESSAGE EMBEDDING

In computer science, a linked list is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of a datum and a reference (in other words, a *link*) to the next node in the sequence; more complex variants add additional links. This structure allows for efficient insertion or removal of elements from any position in the sequence.



A linked list whose nodes contain two fields: an integer value and a link to the next node. The last node is linked to a terminator used to signify the end of the list.

Linked lists are among the simplest and most common data structures. They can be used to implement several others Common abstract data types, including stacks, queues, associative arrays, and S-expressions, though it is not uncommon to implement the other data structures directly without using a list as the basis of implementation.

The principal benefit of a linked list over a conventional array is that the list elements can easily be inserted or removed without reallocation or reorganization of the entire structure because the data items need not be stored contiguously in memory or on disk. Linked lists allow insertion and removal of nodes at any point in the list, and can do so with a constant number of operations if the link previous to the link being added or removed is maintained during list traversal.

On the other hand, simple linked lists by themselves do not allow random access to the data, or any form of efficient indexing. Thus, many basic operations — such as obtaining the last node of the list (assuming that the last node is not maintained as separate node reference in the list structure), or finding a node that contains a given datum, or locating the place where a new node should be inserted — may require scanning most or all of the list elements.

In this section we are going to embed the message in cover image with a structure like what linked lists place in the memory. As you know, linked list is a data structure like an array, but the most important difference between them is in their placement in RAM. Arrays sit in sequential order of memory places, but linked lists get sporadic addresses, and each item (which is called “node”) stores proposed data and also the address of the next item in the memory [11]. Using this concept, we will embed the separate bytes of message sporadically in cover image, so that, address of the next byte far apart in the image be placed just after embedding each byte. By this, two advantages will be achieved: First, non sequence of message structure makes the detection harder, and it increases the security level. Second, the address of first byte of message could be used as stego-key. As you know, while working with linked lists, always the address of first node is stored in a pointer for accessing the linked list data. Losing this address means losing the data stored in linked list. So, we can take this concept to work in steganography for creating a key for message. Fig.2 shows the structure of message in the cover image.

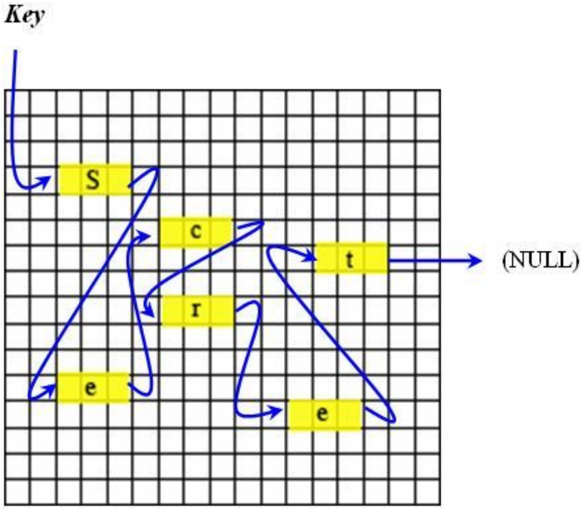


Fig.2Embedding a linked list structured message in cover image

Another important point we have to consider is about the random location of message bytes. Each image provides a 2 dimension (X,Y) space for putting the message in it. Now, what we need is an algorithm to generate the address of no repeated locations. There are some algorithms for it. For example, you can suppose the image as a simple1dimension array, and then do the block scheming. Each block can be a set of pixels for storing a byte of message and also some extra pixels to store the next byte address. The number of pixels in a block depends on the size of image. For an image with x*y pixels, following equation can be used to determine the number of pixels needed for storing the address.

(I)

$$p = \left\lceil \frac{k}{3} \right\rceil : x * y \leq 2^k$$

That *p* is the number of pixels for address. There are *(x*y)* items that should be addressed. So, we need *k* bits so that $x*y \leq 2^k$. It can be concluded that the number of pixels will be equal to *p*in ***(I)***. For example, suppose that there is a 20*30 pixels image. For addressing:

$$20 * 30 \leq 2^{10}$$

$$p = \left\lceil \frac{10}{3} \right\rceil = 4(pixels)$$

So each block in this image should contain 3 pixels for a byte of message, plus 4 pixels for address of then ext byte. After block scheming, now you can use the random numbers generation algorithms to choose the blocks forstoring data in them. We aren’t going to describe these algorithms. Just suppose that there is a function which is named "*RandomBlocks()*" and it does the block scheming and random block selection. For more information about generating random numbers see [12]. Here, we present an algorithm for embedding the message in a cover image with linked list structure. It is done by a function which we call it "write()".

```
Classblock:
{
block();// Constructor
void SetData(byte); //Sets the byte for data part of current block
void SetLink(block); //Sets the block for link part of current block
byte GetData(); // returns the Data part of current block
blockGetLink();// returns the Link part of current block
blockGetAddress();//returns the address of current block
}
functionWrite(message)
{
new=RandomBlock(); new.SetData(Firstbyteof message};
key=GetAddress();
previous=new;
for each byteinmessage //Fromsecondbyte
}
{
new=RandomBlock(); new.SetData(byte);
previous.SetLink(new.GetAddress); previous=new;
}
previous.SetLink(NULL);
}
```

For reading the message a recursive algorithm is presented. It is done by *read()* function. While calling this function for the first time, *key* should be sent as a parameter of this function.

```
byte Read(block)
{
if(block.GetLink==NULL)returnblock.GetData;
else returnread(block.GetLink);
}
```

DISCUSION ON FEATURES

In this section, we are going to get into the characteristics of this algorithm. So, the following lines can be pointed out:

Presented algorithm can be used for any kind of message embedding such as text, images and even the files; because all of them can be reached in bytes form.considering equation (I), maximum bytes of the message that can be embedded in a image with x*y pixels (nb) is calculated as follow:

$$\textbf{(II)} \quad n_b = \frac{\sum(pixels)}{p+3} = \frac{x*y}{p+3}$$

It is easily possible to add new blocks of data in the chain of message. Also, removing them could be done easily. More than one message can be embedded in a cover image. It means we can have more than one chain of message with different keys. This feature could be very useful when the receiver of message is more than one, and each of them should receive their own message. This algorithm can be used as a layer of programming in the process of securing data.

V. CONCLUSION

In this paper, we talked about basic notions of steganography and also took a look at LSB embedding technique in RGB 24-bit color images. After that, a way for image block scheming was introduced, in order to create a structure like linked lists. Also, some rules were defined to set the size of each block. It was mentioned that the goal of block scheming was creating stego-key and making the detection of message harder. Finally, the algorithm for embedding the message was presented, and its characteristics were talked.

REFERENCES

[1] The USC-SIPI Image Database, University of Southern Cal-ifornia, Signal and Image Processing Institute. Available at: http://sipi.usc.edu/database/, last accessed in November 2015.

[2] Kapil Juneja, Archana, Meenakshi, Covering Data in Video Files by using Tree Dimensions of Data Security, Interna-tional Conference on Science and Engineering, Rohtak, India, 2011.

[3] Sara Khosravi, Mashallah Abbasi Dezfoli, Mohammad Hos-sein Yektaie, A new steganography method based HIOP (High-er Intensity Of Pixel) algorithm and Strassen's matrix multipli-cation, Journal of Global Research in Computer Science, Vol. 2, No. 1, 2011.

[4] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Tech-niques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[5] Nick Nabavian, CPSC 350 Data Structures: Image Steganography, 2007, Available

at:http://www1.chapman.edu/~nabav100/ImgStegano/download/ImageSteganography.pdf

[6] G. J. Simmons, "The prisoners' problem and the subliminal channel" in Proc. Advances in Cryptology (CRYPTO '83), pp. 51-67.Berglund, J.F. and K.H. Hofmann, 1967. Compact semi-topological semigroups and weakly almost periodic functions. Lecture Notes in Mathematics, No. 42, Springer-Verlag, Berlin-New York.

[7] Christian Cachin, Digital Steganography, Encyclope-dia of Cryptography and Security, 2005.

[8] Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay and Sugata Sanyal, "Steganography and Steganalysis: Different Approaches", International Journal of Computers, Information Technology and Engineering (IJCITAE), Vol. 2, No 1, June, 2008, Serial Publications.

[9] Adrian Ford, Alan Roberts, Color Space Conversions, 1998,Available at: http://www.poynton.com/PDFs/coloureq.pdf

[10] Mohamed Amin, Muhalim and Ibrahim, Subariah and Salleh, Mazleena and Katmin, Mohd Rozi (2003)Information hiding using steganography. Project Report. Available at: http://eprints.utm.my/4339/1/71847.pdf

[11] Robert Krenn, Steganography and steganalysis, Internet Publication, March 2004. Available at:http://www.krenn.nl/univ/cry/steg/article.pdf

[12] Elis Horowitz, Sartag Sahni, Dinish Mehta: Fundamentals of Data Structures in C++, 2nd ed. Silicon Press, 2006.

[13] Pierre L'Ecuyer: Comparison of Point Sets and Sequences for Quasi-Monte Carlo and for Random Number Generation. SETA 2008: 1-17