Information hiding

Soldatov Nikolay

Preliminaries

This text was made as the summary of my presentation speech on JASS' 05 seminar, so it is structured according to the slides order and has references to the slides in corresponding moments. It may seem too fragmented, but in advantage it makes the text more structured and with help of presentation itself it would be easier to understand the basic concept of the subject.

The introduction

Recently in some of US newspapers appeared the information of using spam messages as the way of secret communications. This technique is supposed to be used by terrorist organizations and various military agencies, with help of steganographic algorithms, which are the part of a young information technology field called *the information hiding.* It is mainly divided into two parts: *steganography* and *watermarking*. Steganography is used not to encipher the information, but to hide the occurrence of communication in some sort of communication medium called *stego-object.* The secret information is called an *embedded* data. As an example we may consider images on the Internet, spam messages etc. The watermarking is commonly used to protect the information from changing or to provide the possibility of author's rights confirmation etc. The main difference is that cover-object plays a small role in steganography case and has a great meaning in the case of watermarking.

The interest of scientific society to this field rose in 1996 when the first academic conference on this subject was organized (see presentation, p. 2).

Use cases

To explain the interest to the subject of information hiding some area of application may be cited. For example steganography is known to be used by military and law enforcement agencies, by intelligence and counter-intelligence agencies, and to provide the anonymous communication on the Internet. Watermarking techniques are used in copyright protection. It is also may be useful in civilian purposes like digital elections and digital cash. Some more use cases will be mentioned later.

Road-map

Further the text will be structured the following way: first there would be given the basic theory of steganography, then the conception of robustness would be explained; the classification of information hiding techniques will help to realize the structure of the subject and after that some known weaknesses and attacks would be mentioned.

The basic theory of steganography

Now it will be important to understand the basic concepts of methods that are used in information hiding. There is a strong practical reason to understand whether it is able to provide the steganographic technique which secrecy is mathematically provable. Investigations in this way where made in [1].

Early results

It is better to start from the very beginning of the theory to clarify the main definitions. Speaking generally, the main purpose of steganography is to set up a channel, which will provide a secret communication between two persons and will guarantee that the

third person would not be able even to realize that the secret communication takes place, by simply looking at the cover medium.
One of the first researches where made by Simmons in 1983 are known as the "prisoners' problem".
Alice and Bob are in jail and try to work out an escape plan; they do not exchange secrets before going to prison, but have some kind of public keys known to each other. They're allowed to communicate with each other, but there is also a warden Willie, who will interrupt the communication if he will find something suspicious in messages (see presentation, p.6). The warden can be active or a passive one. *Active warden* has an additional ability to change messages, and in that case we have a general steganography type. The randomness of the cover-messages provide required channel, which seems to be secret especially in the passive warden case. As there is no way for message to be detected, Simons called the channel *"subliminal"*.
However, this scheme may not work in the active warden case. To avoid this *"supraliminal"* channel may be created. It means that the whole information stream there has a channel with relatively small bandwidth, and as it may contain the most perceptually significant components it makes unable to warden to modify it. So the active warden case was turned to a passive.
As we can see the capacity bounds of embedded data are strictly dependent on the stego-object properties, and it makes the problem over-specified. So the security evaluation problem could be divided into smaller parts for further research.

Robust marking systems

In addition to the cover object properties some more conditions must be added, which are connected with *robustness* of marking systems, and lead us from pure steganography to its application in watermarking.
Robustness is the one of the most important criteria for marking systems and may be considered as the simplest problem in one way and the hardest in other. The simplicity lies in that everyone might know that the object is marked, so there is often no reason to provide secrecy. But in the other way it becomes very important to protect the embedded data from changing, as the target setting turns every warden into on active one. As a working definition robust marking systems are ones that satisfy the following requirements [1]:

- Not to degrade the quality of a cover object, as it is very important for example for media objects.
- Detecting the presence and the value of the mark must require knowledge of a secret key (*stego-key*) (excepting the case of *visible watermarking*, which would be defined later).
- Marks must not interfere. If various marks are set they must not damage each other, because the other way it obviously will provide a wide gap for attacking the system.
- Marks must survive all attacks that change image quality.

As we can see this new conditions make the problem of security proving much more difficult in general abstract case.

Types of robust marking systems

Theoretically the following types of robust marking systems exist:

- Private marking systems, which are divided into two more types. *Type I* systems extract the mark from a possibly slightly changed image, using the original unmarked image as a prompting. *Type II* systems require the copy of the

watermark to compare it with embedded ones and give out “yes” or “no” answer whether the desired mark takes place.
- Public marking systems remain the most challenging problem, since it requires neither the original image nor the example of a watermark.
- Asymmetric marking is the one that allow everyone to see the watermark, but not to change it.

Steganographic systems

Here are some examples of steganographic systems that are available on Internet. *Jsteg*, *JPhide* and *OutGuess* are working with JPEG and GIF formats as stego-mediums. There is also *SecureEngine* that hides information in BMP, GIF, HTM, and TXT files. In [2] there are presented such systems as *Stegdetect* (allows automatically detect steganographic content in JPEG images) and *Stegbreak* (launches a dictionary attacks against steganographic systems to test whether steganographic content is indeed hidden in an image).

Classification of information hiding techniques

We now came to the point when more detailed classification and definition of information hiding systems need to be made. It is important to be done now because then we’ll have a deal with more concrete examples. This classification was made in [1] and it totally reflects the structure of the subject (see presentation p. 10).
As mentioned earlier *steganography* is the way of establishing a secret information transfer channel, which would be hard to detect for possible attacker.
Copyright marking may use the same techniques, but additionally must survive robustness attacks.
Marking can be divided into robust and fragile one. *Fragile watermarking* meant to be changeable. Its purpose is just to show whether a marked image was changed or not. The robust watermarking was described earlier.
Robust watermarks can be visible (*visible watermarking*) and invisible (*imperceptible watermarking*).
There is also a special branch of robust copyright marking called *fingerprinting*. So called *fingerprints* are some kind of serial numbers that show if license agreement was broken and the product was copied and used illegally.

Steganographic techniques

In this paragraph we’ll have a brief look at some of the steganographic techniques, which were used in ancient times and which are used now. They’re divided into several parts by the types of algorithms used in their implementation (see presentation p. 14). Some techniques may be referred to different types, but for simplicity they would be mentioned only in one paragraph.

Security through obscurity

This is the most trivial (in global meaning) type of steganographic algorithms. It is called trivial, because the main idea is simple: embedding the data in the most unexpected places to make warden impossible to understand whether some secret communication exists. However, implementations may sometimes be complicated.
In the book *Schola steganographica* it is described how messages could be send using musical scores. The idea is to connect each note with a letter, so the message will seem to be a musical composition.
As an example of more complicated algorithm we may consider expanded version of code proposed by Johannes *Trithemius*. The code uses 40 tables with 24 entries (one

for each letter of four languages). Each letter of the plain text is replaced by a word or a phrase from the table. So encoded text may look like a prayer or a spell. As we can see from these examples, the main definition of steganography is satisfied: the secret channel established and the occurrence of communication is hidden.
Additional examples may be found in [3] and [4]. In these books it is shown how we can use geometric drawings as a stego-medium, and of usage of *semagrams* and *acrostic* algorithms. Digital version of this type of steganographic techniques is to embed data in the least significant bits of a cover-object.

Camouflage

The next type of algorithms is based on camouflaging the messages, making them "invisible" for warden. The main disadvantage of this principle is that once the algorithm is known, it becomes dangerous to use it again, as it would be easy to break, because the *Kerckhoffs'* principle is obviously failed.
To prove it we may consider the classical camouflaging algorithm, which was invented by a Roman general *Histiaeus* around 440 B.C. (and was still used in the 20th century!). Histiaeus ordered to tattoo a massage on shaved slave's head, and wait till the hair grew back. So the slave became a stego-medium. But there is no need to know any kind of a key to break this "algorithm", so another attempt to deliver messages this way could be revealed.
More interesting example of camouflaging was represented by *Sho*, an artist, in his masterpiece, called "*Vexierbild*". He drew a painting on which you could see a strange landscape, if you looked straight. But if you looked at the painting from a proper corner you could see portraits of famous kings. Of course this method is too unpractical, and obviously breakable. It is just a matter of luck that these methods still survive.
The modern digital variant of these algorithms are masking algorithms (see presentation p. 17). They're based on human perceptual characteristics and been used for example for burying the data channels in audio CDs: the louder tone will mask the quieter tone with the same frequency.

Hiding the location of hidden information

This paragraph describes the methods that create the information channel with a small bandwidth in a wide data stream, to hide the location and even the presence of hidden information. To understand the method let us consider the following example: suppose that we have a stream of information and it is been transmitted with errors, which are caused by the nature of transmission channel (noise in radio-waves etc.). If we understand the cause of these errors we can create them by ourselves, and deliver information using Morse code (short delays between errors would mean "dots" and long – "dashes").
The implementation of this idea may vary, of course. In ancient China it used to be realized by creating sheets of paper with a set of symbols on it, which seemed to be random, and special paper masks. If the receiver would put the mask on the sheet of paper in a proper way he would get a message. The set of holes in the paper mask serves as some kind of a secret key for the algorithm. This algorithm was still used by a British bank in (1992).
In electronic publishing format features of a document may be used as an information carrier. Spaces between strings may slightly be increased or decreased to encode zeros and ones.

Spread the hidden information

The main difference between the hiding the location of hidden information and spreading the hidden information is that the goal of the last mentioned type is to survive the compression.

Patchwork [5] randomly chooses n bits of an image, using pseudo-random generator, and increases or decreases its luminosity, without any changing of the average luminosity of the image. It is said to survive even JPEG compression, but unfortunately it embeds only one bit of information. Providing robustness to compression to JPEG is done using the features of the format, like DCT (*Discrete Cosine Transform*) [2] (see presentation p. 20). The watermark in the modified message is said to present if the ratio $W \cdot W' / \sqrt{(W' \cdot W')}$ is greater than the given threshold. In this formula W represents the original watermark and W' represents possibly changed one.

Spread spectrum systems encode data in a binary sequence, which is been transferred into transmitting waves (like radio waves) and seems to be a noise to a not well-informed person.

Some tools may work with almost compressed objects. *MP3stego*, for example, works with *MPEG Audio Layer III bit streams*.

One of the new techniques is the *echo hiding*. It embeds echoes with different delays to a given audio stream, to encode zeros and ones, using cepstral transform (Fourier cosine transform of logarithm of power spectrum).

Techniques specific to environment

These techniques exploit the features of the environment. One of the most known military developments is *meteor burst communication*. Meteor traces in the atmosphere leave some kind of traces in radio-space. These traces may be used as stego-mediums for embedding data. The presence of communication would be hidden by the fact, that the traces are nature fact, and no artificial objects are added.

As the lamp, using in photocopiers have a high UV content, they may be arranged that papers would come out with some specific message embedded (like “VOID” for example).

Much more different techniques, like original (not digital) watermarks, holograms, microprints and luminescent fibers are used in document protecting.

Also *covert channels* should be mentioned. Cover channels are the channels that weren’t made for information transferring and not artificial, but natural casual emanation. Analyze the following example: suppose that we have to distribute some kind of software and do not want it to be copied illegal. We add a special program to our software that operates with electro-magnetic emanation of computer monitor, to transmit the unique signal. If we receive the same sequence of signals from different working stations, it aware us that our software is copied without license.

Known attacks

In this paragraph there would be described some disadvantages of steganographic techniques discussed above, and possible breaking algorithms.

As it was said before, too many conditions should be imposed on the system to be sure that it is provable safe. As we know that this problem was not solved, we may suppose that this fact leaves a lot of flaws for possible attacks. The classification of the attacks was described in [1], and basic moments will be described in this paragraph. The definitions of attacks would be described as follows (see presentation p. 22).

Jitter attacks

It is known that spread spectrum signals do not survive timing errors. So as an example of a basic attack could be considered a simple resynchronization.

More sophisticated attacks are presented. In [6] authors describe a special tool that is able to increase or decrease the length of a musical composition without changing a pitch (it is been used in broadcasting). Applying this tool would cause timing errors, and will make audio watermark (provided by echo hiding for example) unable to extract.

Robustness attack

Robustness attacks are the most important and destructive ones, because they exploit the weakest side of steganographic systems. In spite of that steganographic systems are bear most of the basic manipulations like rotation, resampling, resizing and compression, they're unsteady against combinations of them with applying of various kinds of geometric distortions. This is one of the basic ways for robustness attacks.
StirMark [7] is a tool that was invented for testing the robustness of different watermarking systems. It applies a large set of different geometric transformations, frequency displacement and deviation, and finally embeds a small error in each sample value. Quality loss is unnoticeable (see presentation pp. 25, 26) and watermarking algorithms may not survive this kind of cover-image changing. In general we may say that given an image as a target, one may invent a proper set of different distortions that would destroy the watermark without loosing any perceptual quality.

Attack on echo hiding

The most obvious attack on echo hiding method is to simply remove an artificial echo, which was added as a watermark or any secret data. The problem is to detect echo without any knowledge of previous parameters of original object, and parameters of added echo. This problem is known to be hard in general and is called "blind echo cancellation".
However, experiments on different audio signals shown, that there is a way to get quite accurate evaluations of the delay, using cepstrum analysis, when any artificial echo was added to the original signal. This leads us to more sophisticated way of attack with combination of the techniques obtained from these experiments and a simple cancellation as a "brute force" [1]. This methods works well if use it iterated.

The mosaic attack

Besides the robustness attacks there are other techniques that can demolish watermarking techniques as well.
The mosaic attack, for example is directed on modifying picture files in the way that prevents extracting of embedded mark. It is based on chopping the in a subset of smaller images. Subimages are placed close to each other so it is unable to distinguish parted image and original one. The watermark would be unable to extract if the partition would be quite strong. Anyway there is no algorithm, than can put a watermark into a single bit. If the illegal copied image will be placed on a pirate site, real time chopping may be used, to prevent extraction of a watermark by a special watermark-seeking crawler.

Protocol level attack

Attack may also use not the defects of the algorithms, but the basic principles of watermarking. For example, if two watermarks are embedded there is no way to figure out which one is the original. To prevent it timestamping or notarization may be used.

Implementation considerations

There is another way of attacking, exploiting weaknesses in the implementation, rather than in marking algorithms. Watermark distribution companies work the following way:

each user has a unique ID number, corresponding to a secret key, which is used in watermarking algorithm.
As the ID is public password search or disassembling may be used to create an impersonated user. Using a changed ID impersonated user may embed own watermark, with help of a debugger to bypass the checking of a previous watermark.

Statistical analysis

Before attacking, adversary should check whether the watermark persists. *Statistical analysis* may be used as a perfect tool. Some of the known analysis algorithms are presented in[2](see presentation p. 31).

References

[1] Fabien A. P. Petitcolas, Ross J. Anderson, Markus G. Kuhn; Information Hiding – A Survey, IEEE papers., 1998
[2] Niels Provos, Peter Honeyman; Detecting steganographic content on the Internet [3] J. Wilkins; Mercury: or the secret and swift messenger: showing how a man may with privacy and speed communicate his thoughts to a friend at any distance. 1694
[4] D. Kahn; The Codebreakers. 1996
[5] W. Bender, D. Gruhl, N. Morimoto, and A. Lu; Techniques for data hiding. 1996
[6] K. N. Hamdy, A. H. Tewk, T. Chen, and S. Takagi; Timescale modification of audio signals with combined harmonic and wavelet representations. 1997
[7] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn; Attacks on copyright marking systems.