**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 5, May 2014, pg.1051 – 1054***

**RESEARCH ARTICLE**

# Steganography on Android Based Smart Phones

**[1]S.Sivakumar, [2]B.Rajesh**

**[1]Department of Computer Science, Periyar University, Tamilnadu, India**

[1] ssk@gmail.com

**[2]Department of Computer Science, Periyar University, Tamilnadu, India**

[2] brajmca@gmail.com

**ABSTRACT**

***Steganography is an old technique to hide data into resources. In this paper we have explained how this technique can be used in android based smart phone. The paper includes basics of image representation, limitations of smart phones like android, other alternatives and implementation on android. There are some limitations on implementing steganography.***

## What is steganography?

Steganography is a process of hiding text or information in existing content. Mostly images or multimedia (audio/video) is used to hide the content. In simple terms one can say that using steganography one can hide text or other content into text or multimedia file. Say for example I have some confidential Information then I will hide that information into an image/audio/video file. This allows me to protect my content from the unauthorized access. Since the image/audio/video file in which I have stored information remains unchanged, no one can easily identify whether the image contains any other data inside it or not.

Steganography is used very commonly at many places, for example terrorists and criminals use steganography to hide their personal details and message while communicating with their team members, companies use steganography to hide some confidential information to restrict piracy or authenticate customers, black hat hackers use steganography to hide Trojans or virus and implant them on the victim"s machine.

As it is a traditional way to use this type of steganographic techniques on traditional computers like desktop, laptop, servers, etc… it can also be used on the mobile and smart phones.

Since the latest mobile/smart phones come with rich multimedia support there are chances that this effective steganography technique will be used or misused by many people around the world.

As the smart phones are still in their infant stage there are few issues in using steganography technique in them.

In following sections first I will discuss how technically steganography can be implemented and then I will discuss some common issues that can create challenges for IT professionals in implementing steganography on mobile/smart phones.

## Image Data Representation

*Black and White Images:* Binary images are encoded as a 2-D array, using one bit per pixel, where a 0 usually means „black‟ and a 1 means „white‟ (even though there is no universal agreement on that). The main advantage of this representation is its small size.

*Monochrome Images:* Gray-level images are also encoded as a 2-D array of pixels, using eight bits per pixel, where a pixel value of 0 usually means „black‟ and a pixel value of 255 means „white‟, with intermediate values corresponding to varying shades of gray. The total number of gray-levels is larger than the human visual system requirements, making this format a good compromise between subjective visual qualities and relatively compact representation and storage. An 8- bit monochrome image can also be thought of as a collection of bit-planes, where each plane contains a 1- bit representation of the image at different levels of detail.

*Color Images:* Representation of color images is significantly more complex and varied. The two most common ways of storing color image contents are: RGB representation – in which each pixel is usually represented by a 24-bit number containing the amount of its Red, Green, and Blue components – and *indexed* representation – where a 2-D array contains indices to a color palette (or look-up table – LUT).

*24-bit Color Images:* Color images can be represented using three 2-D arrays of same size, one for each color channel: Red, Green, and Blue. Each array element contains an 8-bit value indicating the amount of red, green, or blue at that point, in a 0 to 255 scale. The combination of the three 8-bit values into a 24-bit number allows for 2 24 (16,777,216, usually referred to as 16 million or 16 M) color combinations. An alternative representation uses 32 bits and includes a fourth channel, called the *alpha channel,* which provides a measure of transparency for each pixel and is widely used in image editing effects.

*Indexed Color Images:* A problem with 24-bit color representations is backward compatibility with older hardware which may not be able to display the 16 million colors simultaneously. A solution devised before 24-bit color displays and video cards were widely accessible was to adopt an indexed representation, in which a 2-D array of the same size as the image contains indices (pointers) to a color palette (or look-up table – LUT) of fixed maximum size (usually 256 colors).

## Limitations in implementing steganography on smart phones

- There cannot be a common algorithm - since there are many smart phone manufacturers. It is almost difficult to develop a common algorithm for steganography. Another issue is there are lots of variations in smart phones and their architectures; you can find different devices with different features and facilities from same vendor. Further on regular interval new versions or models of same product are launched. So, due to unstable market it seems very difficult to have a common steganographic algorithm for smart phones.
- Smart phones are small computing devices - even though smart phones are smarter than mobile phones they are not sufficient enough to provide efficiency of a traditional computer like desktop or laptop. Smart phones are still small computing devices and have limited resources compare to desktop or laptop. The file size and operations on files at low level are not very easy in smart phones. Another factor in smart phone is availability of APIs and libraries, the APIs or libraries available on desktop cannot be available on smart phones. So, algorithms developed for computers cannot be used for smart phones.
- Low level operations – Most of the smart phone vendors allow file editing at low level but still there are few smart phone vendors who are not giving access to low level operations on files. If you look at blackberry and apple then for security reasons they have kept some restricted APIs which cannot be used by third party developers. So you cannot use such restricted APIs in your application. Basically all the steganographic techniques need to alter the image at byte level. So if the smart phone on which you are working does not allow you to alter the image at byte level then you may not be able to implement steganography.
- *Testing environment* – on desktop/laptop we can develop, compile, run and test applications easily. But in smart phones we do not have such programming or testing environment. So it may happen that for smart phones we need to go through tedious job of testing and deployment. Even though emulators and simulators can be used for testing purpose, sometimes they cannot give exact results (especially for the runtime issues). This becomes very complex when you are developing an algorithm to cover most of the phones. For example if you are working on blackberry platform then there are around 100 simulators available for different blackberry devices. So for each simulator you need to test the app and that too can give you inaccurate results.

## Why Android?

As we discussed in last section it is really challenging to have a common algorithm for all existing smart phones, so it is wise to select a specific platform and develop an algorithm for it. Another issue we discussed in last section is about availability of APIs and libraries that can allow us to perform low level operations easily.

After considering above two major points I feel that at current stage Android is the best option for implementing steganography. There are many advantages of using android platform but the best part is it is an open source product. Since we have access to the source of the operating system, it can be altered easily as per our requirement. Even though it is very difficult and time consuming task to modify operating system for implementing the source code but as a last option if we wish to go for it then we have facility to do so.

Another advantage of Android is, it is java based and has rich set of APIs – especially related to image and multimedia.

Apart from Android there are few other open source operating systems in market like: MeeGo, OpenMoko and few other such. But Android is more powerful and it is widely accepted by many handset manufacturers around the world.

So considering all above factors, Android seems to be the best option at this moment.

**Steganographic Algorithms**

Since steganography is an old concept there are many algorithms for implementing it on traditional computers. These algorithms have pros and cons and based on requirements they can be used to achieve steganography. The most common algorithm for steganography is based on LSB (Least Significant Bits) and is proved to be more efficient compare to other algorithms.

*LSB:* The LSB algorithm, involves the modification of the LSB layer of images. In this technique, the message is stored in the LSB of the pixels which could be considered as random noise. Thus, altering them does not have any obvious effect to the image

**Implementation**

The implementation depends on selection of algorithm. In general for android devices an android application is developed with required classes to implement algorithm. The application can be developed with help of IDE (mostly eclipse). Initial testing can be done on emulator and final APK file is deployed to the device for live testing.

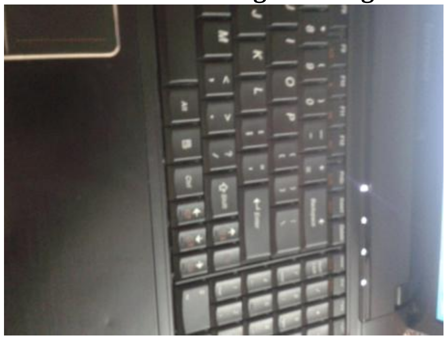
**Testing Environment:**

- *Android OS:* 2.2
- *Device:* Samsung Galaxy Ace
- *Processor:* 800 MHz ARM 11 processor
- *Camera:* 5 MP

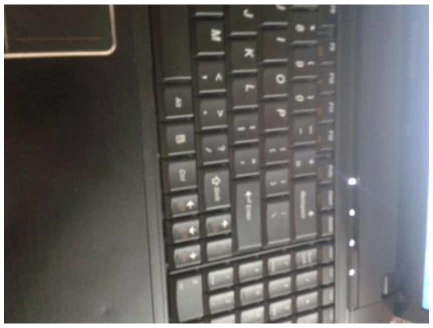
**Steps Performed:**

- Captured image from camera
- Image is compressed and saved to SD Card after steganography

**Original Image**



**RESULT Image with hidden text (After steganography)**



**Other Limitations of Steganography**

As we discussed steganography simply writes data to image by replacing bits or bytes of the image, so when an image having updated text is modified it may happen that the information stored in image is lost.

For example if I crop or resize the image and the bits that contain our information are affected due to this operation then that particular information does not remain accessible.

Similarly other operations like changing image settings or applying new effects may also cause loss of data.

But if this technique is used with internal communication or use then there are rare chances of modifying the image, because users know about the image and the hidden text and will not make any changes.
Especially in smart phone the image with hidden text is mostly communicated in the form of MMS. And the information remains there even same image is sent to many recipients either at a time or from one to another. So there are very rare chances of editing the image in smart phone itself.

**REFERENCES**

[1] Alternatives for Multimedia Messaging System Steganography-Konstantinos Papapanagiotou1, Emmanouel Kellinis2, Giannis F. Marias1, and Panagiotis Georgiadis1 [Dept. of Informatics and Telecommunications, University of Athens, Panepistimiopolis, Ilissia, Greece, GR15784]
[2] “Image Steganography: Concepts and Practice”, -M. Kharrazi, H. T. Sencar, N. Memon:[National University of Singapore (2004)]