DOI. 10.1109/ICCCYB.2010.5491253

Steganography on new generation of mobile phones with image and video processing abilities

Daniela Stanescu, Valentin Stangaciu, Mircea Stratulat,
Department of Computer and Software Engineering (DCSE), “Politehnica” University of Timisoara, 2, Vasile Parvan Blvd., 300223 – Timisoara, Romania
daniela.stanescu@cs.upt.ro, valys@dsplabs.cs.upt.ro, smircea@cs.upt.ro,

***Abstract*—Steganography is a science that focuses on hiding specific messages using specialized techniques in such a way as only the sender and the intended receiver are able to decipher it, as well as knowing of its existence. Important domains, besides classic computing, where steganography can be applied are domains using mobile and embedded devices especially mobile phones. This paper tries to state the fact that steganography can be successfully implemented and used into a next generation of mobile phones with image and video processing abilities. Important results presented in this article suggest the fact that steganography can be used inside mobile phones and tablets.**

***Keywords* — steganography, concealing, mobile phones.**

I. Introduction

Mobility is one of today’s working and researching direction and with mobility *secure communication* also comes into interest. One of the many ways of implementing a secure communication on mobile devices is using a method to hide information and moreover to hide the fact that there is, somewhere, hidden information. The domain that represents this concept is steganography. The main aspect is that steganography hides, embeds a secret message into another message [1].

Steganography has a great advantage over other methods of concealing information such as the fact that hidden messages do not attract attention.

Steganography operates over messages represented by text, images, videos and other forms of presenting information. In this field important research has brought into attention many important steganographic algorithms that may be used for concealing information.

Before continuing, an introduction into the most important terms of steganography has to come into interest. The *payload* is usually represented by the data that has to be concealed. The payload is hidden or embedded into a *carrier*. The resulting object after a steganographic operation occurs is called *a package, stego file or a covert message*. This object consists of a secret message – the payloads – embed into a public message – carrier. Another term frequently used in steganography is the *channel* which specifies the type of input.

In this paper the authors present the results obtained after using steganographic algorithms on microcontrollers or digital signal processors that are usually used in mobile phones or other embedded platforms. The channel or type of input used is a bitmap image. The test were mainly made upon the decoding part of the steganographic process.

II. Related Work

After important studies different techniques and algorithms were written to embed large amount of data into a picture as well as the requirements needed.

Information was hidden using digital watermarking with improved techniques based on the decorrelation property of the Karhunen-Loeve Transform [2] as well as a method of hiding messages in digital images based on YUV format and its derivatives [3]. These techniques were used especially against a current common issue like illicit copying and distribution of copyright material.

Eliminating detection of steganography and counteracting attacks meant to extract the hidden information was also a major concern. New and more complex algorithm have been developed to avoid the detection of hidden data as well as embedding hidden information in preprocessed images and in images where compression was applied. [4].

Other researchers present a steganographic model, which consists in choosing an object as a carrier and implementing a secret key in a micro architecture. As a mean of validating, Field Programmable Gate Arrays (FPGA) where used to sustain the idea. The contents of the secret message is distributed inside the carrier using a method based on a secret key known only by the parts normally involved in the communication. Only using the secret key the original message may be reconstructed. [5]

One of the first ideas in implementing a watermarking technique for any model of a digital camera is using a VLSI architecture as a processing unit. The prototype contained about 28500 gates using integration technology of 0.35 μm operating at 300 MHz. The goal was to assure intellectual property protection using a steganographic algorithm [6]. The study is further continued in implementing steganographic algorithms for hiding a secret image inside a bitmap image. The processing unit is represented by an integrated circuit similar to its predecessor but with significant improvements. [7]

Steganography can also be used in generating secure communication over public telephone network. In such cases, the algorithms are implemented using microcontrollers and digital signal processors (DSP) resulting in methods to transmit a secret message behind a simple telephone conversation. [8]

Other important steps have been made in taking steganography on embedded and mobile devices. Using a microprocessor based on ARM architecture and executing steganographic algorithms, result express the fact that secret message represented by a bitmap image can be easily incorporated in a carrier bitmap image thus hiding it and then successfully recovering it. [9]

Hiding data inside bitmap images using steganographic algorithms was also possible in the field of mobile phones. The method was applied during secure date transfers between a mobile phone and a personal computer both terminals needing to execute a program coded using the JAVA programming language. [10]

An important example of using steganography in mobile phones applications is the idea of hiding a black and white image inside an SMS message. The disadvantages in this case are the limitations of an SMS message and the limitation that the image has to be black and white only [11].

III. Steganographic algorithms on mobile devices. Tests and results

Nowadays mobile devices such as mobile phones or tablets offer high performance and computation power making possible to use sophisticated operations that in the past years could only be executed on classic personal computers. Mobile devices currently use high computational digital signal processors and microcontrollers with multi-core architectures as well as architectures based only on one processor core.

An important aspect in mobility is securing information transfer. The information that is transfered between mobiles phones is represented by text, audio, static images and video. Steganography can be used with all these types of data to secure the transfer between mobile devices.

This paper tries to state the fact that steganography can be successfully used to secure a data transfer between mobile devices also offering great performance. The tests and results that are going to be present were made using images as types of information.

Tests and comparisons were made using 3 hosts: ARM7 based microcontroller, a multi-core architecture digital signal processor and a personal computer. The ARM7 microcontroller operates at 60 MHz and is designed as a powerful RISC architecture. The multi-core processor is practically a graphic processing unit from Movidius, named ISSAC, and has specialized units for image and video processing. ISSAC consists of a number of cores each of them specialized in video and image processing and coordinated by another dedicated RISC core based on SPARC architecture, particularly a LEON core. ISSAC can operate at a frequency of 200 MHz and the processor clock is controlled by e Phase-Locked Loop. Prior to testing the algorithms the ISSAC's clock was adjusted at 60 MHz making it equal to the ARM's clock, a necessary setting for a proper comparison.

The challenge in implementing and executing steganographic algorithms on these 3 platforms was that all these platforms are different and have specific characteristics. For instance de ARM7 based platform has pipeline units but doesn't have specialized image processing instructions. ISSAC is a multi-core platform and this fact brings high performance but complicates the implementation because the algorithms have to be modified for parallel execution.

The first steps in this direction were made by the authors when implementing simple steganographic algorithms on ARM 7 platform [3].

There were used three types of steganographic algorithms: algorithms based on the LSB method, algorithms based on the YUV method and algorithms based on the Karhunen-Loeve Transform (KLT).

Algorithms based on these 3 methods were specifically modified and adapted to fit the 3 architectures used to obtain the results being presented later in this paper. Also another goal was to find the best algorithm that not only offers the highest performance but offers the best quality for the recovered image.

The algorithms based on the LSB method usually operate by hiding the most significant bits of the secret message image pixels within the least significant bits of the carrier image pixels. The number of bits that are involved in this operation may vary and the images are in RGB format. The following figure presents an example of the algorithm based on the LSB method. In this case there were used the 4 most significant bits from the secret message to be hidden within the carrier image.

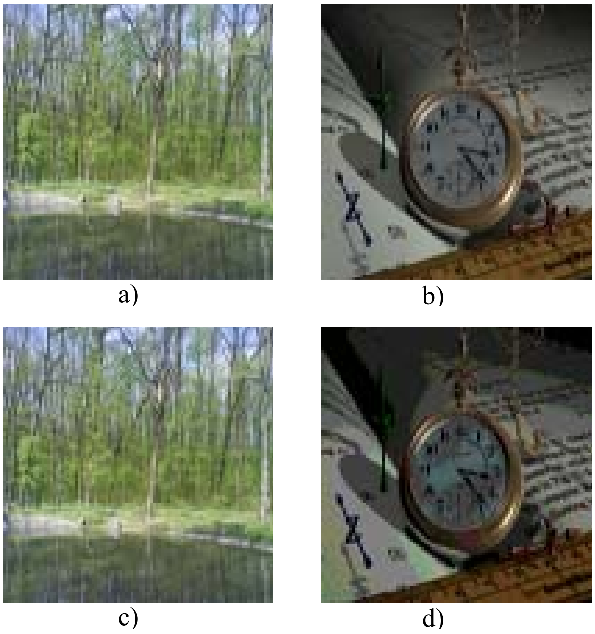
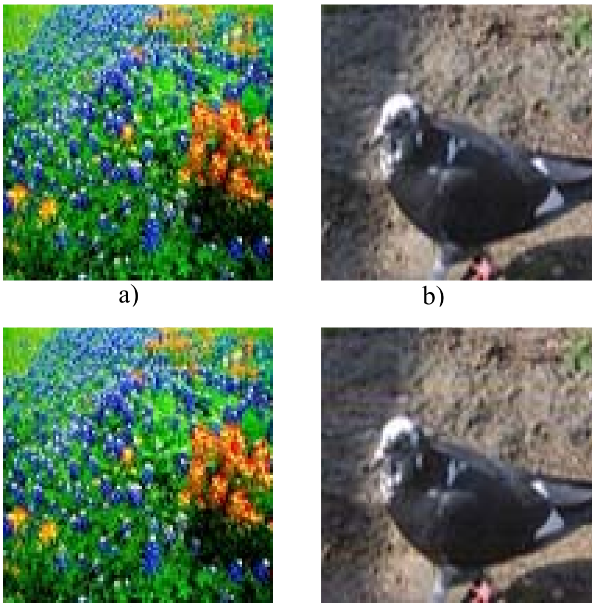


Fig.1 a) carrier b) payload c) stego image
d) recovered image

The YUV method is mainly based on the LSB method. Prior to applying a LSB based algorithm the images are converted from the RGB format to the YUV format. Following the application of the LSB algorithm the images are converted back to RGB format [3].

A Karhunen-Loeve Transform (KLT) based algorithm also uses LSB based algorithms. Differently from the a YUV algorithm, a KLT based algorithm does not hide the value of the color components of the pixels from the secret message into the color components of the pixels from the carrier image but hides parameters resulting after applying a Karhunen-Loeve Transform. [2]

The following figure present an example of applying steganography on a set of images using an algorithm based on the Karhunen-Loeve Transform and the LSB technique presented above.



a) b)

c) d)

Fig. 2 a) carrier b) payload c) stego image
d) recovered image

The following table presents the execution time of a LSB method based algorithm on the three platforms mentioned above using different sets of images of different sizes. In this case there were used the 2 most significant bits from the payload image to be hidden within the 2 least significant bits of the carrier image.

TABLE I

EXECUTION TIME MEASUREMENTS FOR DIFFERENT IMAGE SIZES USING A 2 BIT LSB BASED ALGORITHM

Carrier image size (bytes)	**Payload image size (bytes)**	**Decode time (ms) PC**	**Decode time (ms) ARM**	**Decode time (ms) ISSAC**
81.000	33.930	64	25	5,38
81.000	45.450	78	25	5,38
81.000	81.000	68	25	5,38
202.500	33.930	161	63	12,8
202.500	81.000	160	63	12,8
202.500	182.700	180	63	12,8
360.000	33.930	286	112	22,5
360.000	81.000	281	112	22,5
360.000	182.700	291	112	22,5
360.000	360.000	278	112	22,5
810.000	33.930	611	253	33,6
810.000	81.000	627	253	33,6
810.000	182.700	600	253	33,6
810.000	360.000	605	253	33,6
810.000	562.500	610	253	33,6
810.000	810.000	609	253	33,6

Using sets of images applying the same LSB based algorithm but using 4 bits from the payload image produced little execution time difference on the 3 testing platforms.

TABLE II

EXECUTION TIME MEASUREMENTS FOR DIFFERENT IMAGE SIZES USING A 4 BIT LSB BASED ALGORITHM

Carrier image size (bytes)	**Payload image size (bytes)**	**Decode time (ms) PC**	**Decode time (ms) ARM**	**Decode time (ms) ISSAC**
81.000	33.930	70	25	3,66
81.000	45.450	72	25	3,66
81.000	81.000	73	25	3,66
202.500	33.930	161	63	8,55
202.500	81.000	160	63	8,55
202.500	182.700	162	63	8,55
360.000	33.930	280	112	14,9
360.000	81.000	276	112	14,9
360.000	182.700	269	112	14,9
360.000	360.000	289	112	14,9
810.000	33.930	613	253	33,6
810.000	81.000	616	253	33,6
810.000	182.700	615	253	33,6
810.000	360.000	605	253	33,6
810.000	562.500	618	253	33,6
810.000	810.000	630	253	33,6

As mentioned above a YUV based algorithm was used to hide the entire contents of a payload image inside a carrier image. The following table present the execution times of the algorithm.

TABLE III

EXECUTION TIME MEASUREMENTS FOR DIFFERENT IMAGE SIZES USING A YUV BASED ALGORITHM

Carrier image size (bytes)	Payload image size (bytes)	Decode time (ms) PC	Decode time (ms) ARM	Decode time (ms) ISSAC
360.000	14.700	9976	1369	182,04
562.500	14.700	10707	1371	182,32
810.000	33.930	22403	3162	420,47
810.000	81.000	40936	7550	1006,66

Currently the architecture the ISSAC processor is based on is under development making also the development tools to be an on working project. Certain limitations have occurred in implementing the steganographic algorithms making the implementation of an algorithm based on Karhunen-Loeve Transform a work in progress. Currently a Karhunen-Loeve Transform algorithm was successfully implemented on the ARM architecture. The following table presents the execution time of this algorithm on sets of images.

TABLE IV

EXECUTION TIME MEASUREMENTS FOR DIFFERENT IMAGE SIZES USING A KLT AND LSB BASED ALGORITHM

Carrier image size (bytes)	Payload image size (bytes)	Decode time (ms) PC	Decode time (ms) ARM
81.000	33.930	5532	1243
81.000	45.450	5805	1340
81.000	81.000	6708	1370
120.000	81.000	8041	2040
120.000	120.000	8772	2097
202.500	33.930	10922	1283

Centralizing and charting the above information may help drawing conclusions and stating what algorithm suits better for a mobile device not only regarding the execution time but also regarding the quality of the recovered image.

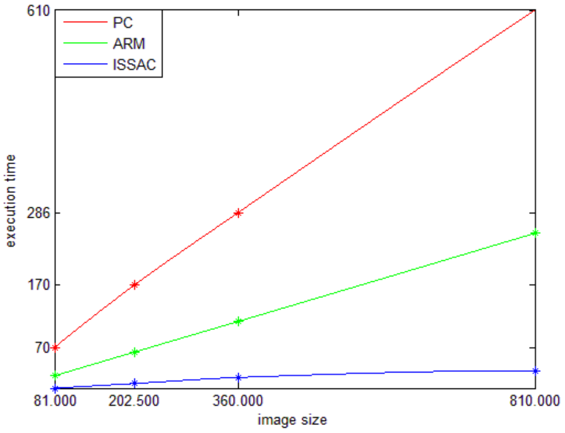


Fig 3. LSB with 2 bit based algorithm execution time time – execution time [ms], image size [bytes]

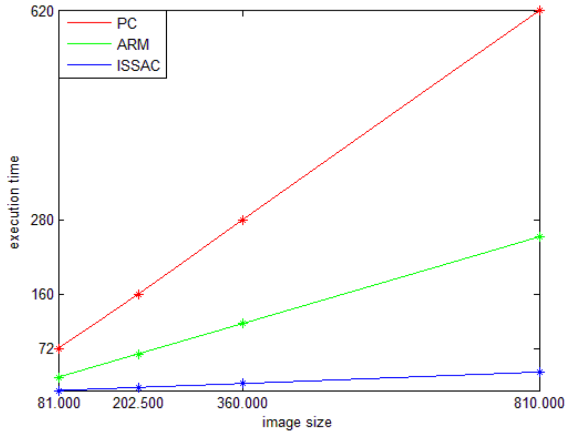


Fig 4. LSB with 4 bit based algorithm execution time – execution time [ms], image size [bytes]

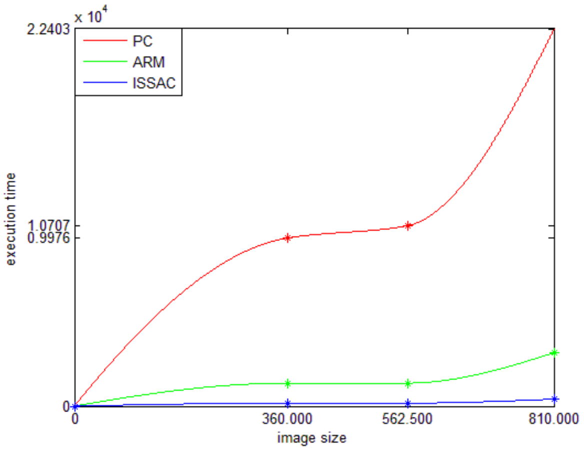


Fig 5. YUV based algorithm execution time time – execution time [ms], image size [bytes]

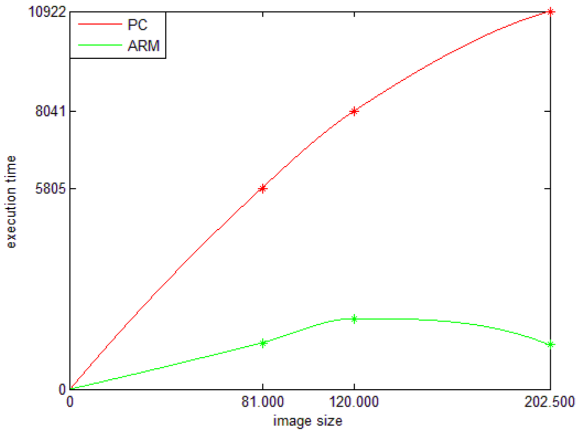


Fig 6. Karhunen-Loeve Transform based algorithm execution time time – execution time [ms], image size [bytes]

IV. CONCLUSIONS

After charting the gathered data we can conclude that the execution time of a steganographic algorithm is highly influenced by the size of the carrier image. Also the best execution time was obtained from the ISSAC processor for all algorithms. Besides the execution time the authors can conclude that the most suitable algorithm for steganography may be the one based on the Karhunen-Loeve Transform combined with the LSB method.

Microprocessors like ISSAC or based on the ARM architecture are frequently used in mobile devices and with the current implementation of steganographic on them we can admit that these algorithms may perfectly fit in a mobile phone with video and image processing abilities. In this way a user of a mobile phone may send secret information without even attracting attention that secret information is involved.

REFERENCES

[1] József Lenti, "Steganographic methods", *Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, H-1521, Budapest, Hungary, June 2000, pp. 249-258*

[2] D. Stanescu, M. Stratulat, B. Ciubotaru, D. Chiciudean, R. Cioarga, and D. Borca, "Digital Watermarking using Karhunen-Loeve transform," *Applied Computational Intelligence and Informatics, 2007. SACI '07. 4th International Symposium on, 2007, pp. 187-190*

[3] Stanescu, D.; Stratulat, M.; Groza, V.; Ghergulescu, I.; Borca, D. "Steganography in YUV color space", *Robotic and Sensors Environments, 2007. ROSE 2007. International Workshop on Volume , Issue , 12-13 Oct. 2007*

[4] Taras Holotyak, Jessica Fridrich, Sviatoslav Voloshynovskiy, "Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics", *Communications and Multimedia Security, 2005, Volume 3677/2005, ISBN* 978-3-540-28791-9, *pp. 273-274*

[5] Hala Farouk, Magdi Saeb, "*Design and implementation of a secret key steganographic micro architecture employing FPGA*", Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings, 16-20 Feb. 2004, Vol.3, pp. 212-217, ISSN:1530-1591 , ISBN: 0-7695-2085-5

[6] Saraju P. Mohanty, Nagarajan Ranganathan, Ravi K. Namballa, "*VLSI Architecture for watermarking in a secure still digital camera (S^2DC) design* ", IEEE Transactions on very large scale integration (VLSI) systems, Vol 13, Issue 7, July 2005, ISSN: 1063-8210

[7] Mohanty S.P., Kougianos E., Ranganathan N. ,"*VLSI architecture and chip for combined invisible robust and fragile watermarking*", Computers & Digital Techniques, IET, Sept. 2007, Vol 1, Issue: 5, pp 600-611, ISSN: 1751-8601

[8] Wu Zhi-Jun Niu Xin-Xin Yang Yi-Xian , "Design of speech information hiding telephone" , TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 28-31 Oct. 2002, Volume: 1, pp. 113- 116 vol.1, ISBN: 0-7803-7490-8

[9] D. Stănescu, V Stângaciu, I. Gergulescu, M. Stratulat, *"Steganography on embedded devices",* SACI, 2009 "Politehnica" University of Timişoara, România, pp, ISBN

[10] Mohammad Shirali Shahreza , "*An improved method for steganography on mobile phone*", Proceedings of the 9th WSEAS International Conference on Systems table of contents, Athens, Greece, 2005, Article No. 28 , ISBN:960-8457-29-7.

[11] Shahreza, S., "*Stealth steganography in SMS*", Wireless and Optical Communications Networks, 2006 IFIP International Conference on, 2006, ISBN: 1-4244-0340-57