International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 8, August - 2013

FPGA Hardware LSB Stegnography Technique Based on the Lifting Scheme

Gorre Josh Kumar[#1], U.N.Subhadra Devi[#2]

[1&2]Dept. of ECE, MVGR college of Engineering ,Vizianagaram ,AP, India

***Abstract*— The least-significant-bit (LSB)-based approach is a popular type of steganographic algorithms in the spatial domain. However, we find that in most existing approaches, the choice of embedding positions within a cover image mainly depends on a pseudorandom number generator without considering the relationship between the image content itself and the size of the secret message. Thus the smooth/flat regions in the cover images will inevitably be contaminated after data hiding even at a low embedding rate, and this will lead to poor visual quality and low security based on our analysis and extensive experiments, especially for those images with many smooth regions in this paper a novel discrete wavelet transformation is proposed to authenticated a multimedia image and at the same tie some ubiquitous secret message or image can be transmitted over the mobile network instead of direct embedding a message of image within the source image, choose a widow of size 3*3 of the source image and then convert it from spatial domain to frequency domain by using discrete wavelet transform is done as a final step of encoding by using LSB concept . And the Decoding is done by using reverse procedure.**

***Keywords*- Content-based steganography, least-signifi-cant-bit (LSB)-based steganography, pixel-value differencing (PVD), security, steganalysis.**

I. INTRODUCTION

Cryptography is the science of securing data; cryptanalysis is the science of analyzing and breaking secure communication. The earliest forms of information hiding can actually be considered to be highly crude forms of private-key cryptography; the "key" in this case being the knowledge of the method being employed (security through obscurity). Steganography books are filled with examples of such methods used throughout history. Greek messengers had messages tattooed into their shave head, concealing the message when their hair finally grew back. Wax tables were scraped down to bare wood were a message was scratched. Once the tablets were re-waxed, the hidden message was secure [3]. Over time these primitive cryptographic techniques improved, increasing both speed, capacity and security of the transmitted message.

Nowadays, crypto-graphical techniques have reached a level of sophistication such that properly encrypted communications can be assumed secure well beyond the useful life of the information transmitted. In fact, it's projected that the most powerful algorithms using multi kilobit key lengths could not be comprised through brute force, even if all the computing power worldwide for the next 20 years was focused on the attack. The possibility exists that vulnerabilities could be found, or computing power breakthroughs could occur, but for most users in most applications, current cryptographic techniques are generally sufficient.

The field of information hiding offers several reasons, the first being that "security through obscurity" provided that it the only security mechanism employed. Steganography for instance allows us to hide encrypted messages in mediums less likely to attract attention. A garble of random characters being transmitted between two users may tip off a watchful 3rd party that sensitive information is being transmitted; whereas baby pictures with some additional noise present may not. The underlying information in the pictures is still encrypted, but attracts far less attention being distributed in the picture then it would otherwise.

This becomes particularly important as the technological disparity between individuals and organizations grows. Governments and businesses typically have access to more powerful systems and better encryption algorithms then individuals. Hence, the chance of individual's messages being broken increases which each passing year. Reducing the number of messages intercepted by the organizations as suspect will certainly help to improve privacy. Another advantage is that information hiding can fundamentally change the way that we think about information security.

Cryptographic techniques generally rely on the metaphor of a piece of information being placed in a secure "box" and locked with a "key". The information itself is not disturbed and anyone with the proper key can gain access. Once the box is open, all of the information security is lost. Compare this to information hiding techniques where the key is embedded into information itself. Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. Cryptography is the science of securing data; cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern, patience, determination, and luck. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

Cryptography can be defined as the processing of information into an unintelligible (encrypted) form for the purposes of secure transmission. Through the use of a "key" the receiver can

decode the encrypted message (decrypting) to retrieve the original message.

Stenography improves on this by hiding the fact that a communication even occurred. The message *m* is imbedded into a harmless message *c* which is defined as the cover-object. The message *m* is then embedded into *c*, generally with use of a key *k* that is defined as the stego-key. The resulting message is then embedded into the cover-object *c*, which results in stego-objects.

II. STEGAOGRAPHY

Steganography means to hide secret information into innocent data. Digital images are ideal for hiding secret information. An image containing a secret message is called a cover image [4]. First, the difference of the cover image and the stego image should be visually unnoticeable. The embedding itself should draw no extra attention to the stego image so that no hackers would try to extract the hidden message illegally. Second, the message hiding method should be reliable. It is impossible for someone to extract the hidden message if she/he does not have a special extracting method and a proper secret key. Third, the maximum length of the secret message that can be hidden should be as long as possible.

"Steganography is the art of hiding information in ways that prevent the detection of hidden messages,"

III. HISTORY OF STEGANOGRAPHY

In Steganography comes from Greek and means "covered writing." The ancient Greeks wrote text on wax-covered tablets. To pass a hidden message, a person would scrape off the wax and write the message on the underlying wood. He/she would then once again cover the wood with wax so it appeared unused.

Many developments in steganography occurred during World War II. This included the development of invisible inks, microdots, and encoded messages. One known message sent by a German spy was: Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetables oils.

Extracting second letter in each word reveals:

Pershing sails from NY June 1.

IV. STEGANOGRAPHY IN THE DIGITAL AGE

Steganography is the art of secret communication. Its purpose is to hide the very presence of communication as opposed to cryptography whose goal is to make communication unintelligible to those who do not posses the right keys. Digital images, videos, sound files, and other computer files that contain perceptually irrelevant or redundant information can be used as "covers" or carriers to hide secret messages. After embedding a secret message into the cover-image, a so-called stego-image [6] is obtained. It is important that the stego-image does not contain any easily detectable artifacts due to message embedding. A third party could use such artifacts as an indication that a secret message is present. Once this message detection can be reliably achieved, the steganographic tool [7] becomes useless.

Obviously, the less information is embedded into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process. Another important factor is the choice of the cover-image. The selection is at the discretion of the person who sends the message. The sender should avoid using cover-images [20] that would be easy to analyze for presence of secret messages. For example, one should not use computer art, charts, images with large areas of uniform color, images with only a few colors, and images with a unique semantic content, such as fonts. Although computer-generated fractal images may seem as good covers[6] because of their complexity and irregularity, they are generated by strict deterministic rules that may be easily violated by message embedding

V. CRYPTOGRAPHY VS STEGANOGRAPHY

Cryptography is the science of encrypting data in such a way that no body can understand the encrypted message, whereas in steganography the existence of data is conceived means its presence cannot be noticed. The information to be hidden is embedded into the cover object which can be text, image, audio or video so that the appearance of cover object doesn't vary even after the information is hidden.

Information to be hidden + cover object = stego object.

To add more security the data to be hidden is encrypted with a key before embedding. To extract the hidden information one should have the key. A stego object is one, which looks exactly same as cover object with an hidden information.

VI. STEGANOGRAPHY VS WATERMARKING

Watermarking is another branch of steganography it is mainly used to restrict the piracy in digital media in steganography the data to be hidden is not at all related to the cover object, here our main intention is secret communication.

In watermarking the data to be hidden is related to the cover object it is extended data or attribute of the cover object, here our main intention is to stop piracy of digital data. Steganography is a very powerful tool because, as the stated above, it can be very difficult to detect

VII. PROPOSED SYSTEM

This paper proposed LSB Information Hiding algorithm which can Lifting wavelet transform image. The idea behind the LSB algorithm [6] is to insert the bits of the hidden message into the least significant bits of the pixels. Achieving the purpose of information hiding with the secret bits of information to replace the random noise, using the lowest plane embedding secret information to avoid noise and attacks, making use of redundancy to enhance the sound embedded in the way nature to be addressed. The results showed that

the proposed algorithm has a very good hidden invisibility, good security and robustness for a lot of hidden attacks.However, the limitation of capacity has led us to think about an improved approach which can be achieved through hardware implementation systems with the help of a programmable gate array (FPGA) board.

It is the process of embedding data within the domain of another data, this data can be text, image, audio, or video contents. The embedded watermark can be visible or invisible (hidden in such a way that it cannot be retrieved without knowing the extraction algorithm) to the human eye, specified secret keys [8] are taken into consideration in order to enhance the security of the hidden data

When compared with the DCT, DWT is found to be more robust against various attacks. THE discrete wavelet transform (DWT) is being increasingly used for image coding. This is due to the fact that DWT [7] supports features like progressive image transmission (by quality, by resolution), ease of compressed image manipulation, region of interest coding, etc. DWT has traditionally been implemented by convolution. Such an implementation demands both a large number of computations and a large storage—features that are not desirable for either high-speed or low-power applications.

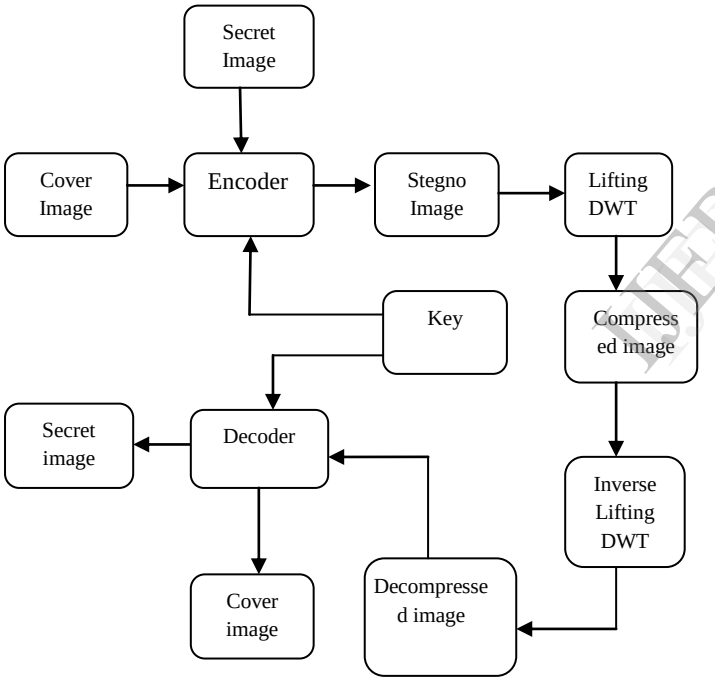

Fig 1: Overall Architecture Block Diagram

Least significant bit (LSB) substitution

- **Easy to understand and implement**
- **Used in many available stego tools**

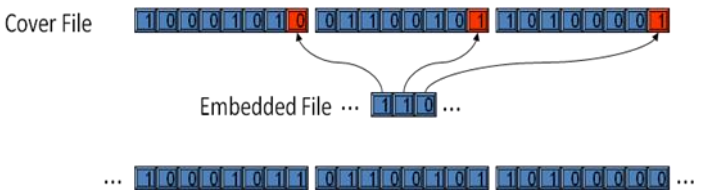

Fig 2: LSB Architecture

Messages are encoded in the least significant bit of every byte in an image. By doing so, the value of each pixel is changed slightly, but not enough to make significant visual changes to the image, even when compared to the original.

Example: Inserting the word "bomb" using LSB techniques:

- **b = 01100010**
- **o = 01101111**
- **m = 01101101**
- **b = 01100010**

Lifting Scheme

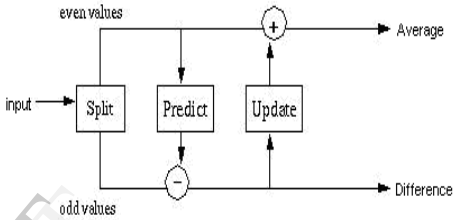

Fig 4: Lifting Scheme Process Flow

It is composed of three basic operation stages:

- **Splitting:** where the signal is split into even and odd
- **Predicting:** Even samples are multiplied by a predict factor
- **Updation:** The detailed coefficients computed by the predict step are multiplied by the update factors and then the results are added to the even samples to get the coarse coefficients.

Need for FPGA

- Reconfigurable
- Low power Dissipation
- Small in size
- Accuracy
- Floating point

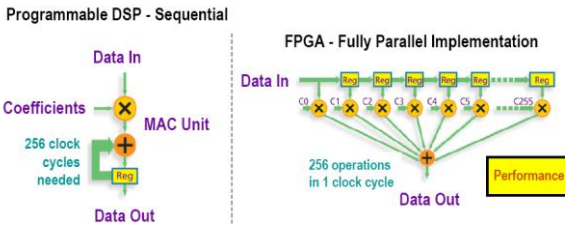

Fig 5: FPGA Vs DSP Processors Comparison

VIII. EXPERIMENTAL SETUP

A. Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system [9]. Embedded Development Kit Xilinx Embedded Development Kit (EDK) is an integrated software tool suite [11] for developing embedded systems with Xilinx Micro Blaze and PowerPC CPUs. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional.

A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification [11]. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software platform is defined by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

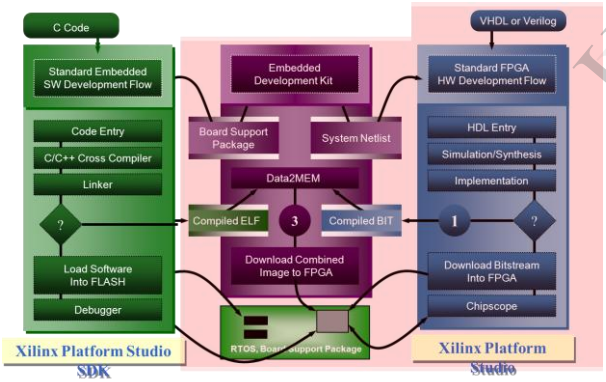

Fig 6: Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK [14] are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects. Platform Generator customizes and generates the processor system in the form of hardware netlists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bitstream Initializer tool initializes the instruction memory of processors on the FPGA shown in figure2. GNU Compiler tools are used for compiling and linking application executables for each processor in the system. There are two options available for debugging the application created using EDK [15] namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. SDK is built on the Eclipse open source framework. Soft Development Kit (SDK) [18] is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx Embedded Development Kit (EDK) [19].The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

IX. RESULTS AND SNAPSHOT

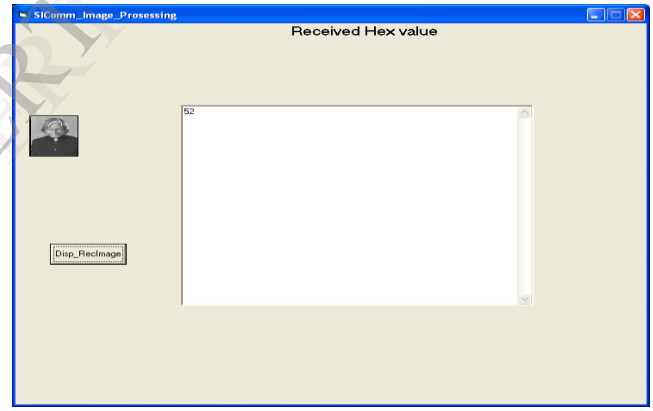

Fig 7: Input Image

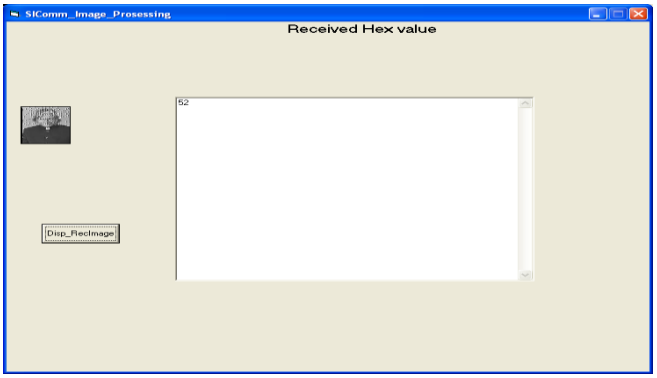

Fig 8: Stego Image after LSB Encrption

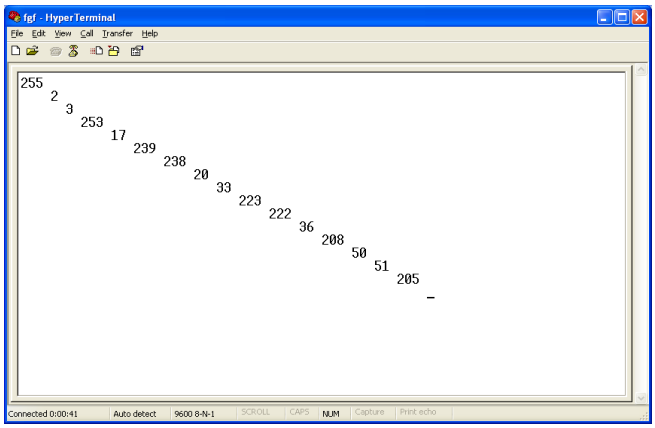

Fig 9: Original Data after LSB Decryption

X. TABULATION RESULTS

The Algorithm is implemented in Micro blaze Processor and the results are furnished in the tabulation below

```
Design Summary:
Number of errors:      0
Number of warnings:    3
Logic Utilization:
  Number of Slice Flip Flops:         976 out of   3,840   25%
  Number of 4 input LUTs:           1,560 out of   3,840   40%
Logic Distribution:
  Number of occupied Slices:                      1,174 out of   1,920   61%
    Number of Slices containing only related logic:   1,174 out of   1,174  100%
    Number of Slices containing unrelated logic:          0 out of   1,174    0%
      *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:          1,953 out of   3,840   50%
  Number used as logic:              1,560
  Number used as a route-thru:           7
  Number used for Dual Port RAMs:      256
    (Two LUTs used per Dual Port RAM)
  Number used as Shift registers:      130
  Number of bonded IOBs:                62 out of      97   63%
    IOB Flip Flops:                    122
  Number of Block RAMs:                  4 out of      12   33%
  Number of MULT18X18s:                  3 out of      12   25%
  Number of GCLKs:                       2 out of       8   25%
  Number of DCMs:                        1 out of       4   25%
  Number of BSCANs:                      1 out of       1  100%

   Number of RPM macros:           3
Total equivalent gate count for design:  325,662
```

XI. CONCLUSION

In this paper report we have presented a new method of adaptive steganography with higher embedding capacity. The embedding capacity of the approach is controlled through the filter cut-off frequency. The approach was analyzed and shown to have a very high confidentiality due to the sharpness of information recovery with the cut-off frequency. The New Approach is using lesser hardware architecture with an reasonable speed of 83MHz Approximately in a Spartan 3 FPGA with 50MHz Clock Crystal. So we can use it to the steganography technique very easily than other techniques without any problem

XII. REFERENCES

[1]. B. Weaver, Now You See It, Scientific Computing 24.6 (May 2007): 18-39.

[2]. B. Glass, Hide in Plain Sight, PC Magazine 21.18 (15 Oct. 2002): 75.

[3]. Tucker, Patrick. "Hiding Secrets in Computer Files." Futurist 40.5 (Sep.2006): 12-12.

[4]. R. Gonzales, and R. Woods, Digital Image Processing, Addison Wesley Publishing Co., 1993.

[5]. C. Birslawn, Fingerprint Go Digital, Notices of American Mathematical Society, Vol. 42, No.11, P. 1278-1283, and Nov. 1995.

[6]. W. Sweldens, Building Your Own Wavelets at Home, Wavelets in Computer Graphics, ACM SIGRAPH Course Notes, 1996.

[7]. A. Calderbank, I. Daubechies, W. Sweldens, and B. Yeo, Wavelet Transforms that Map Integers to Integers, Mathematics Subject Classification, 42C15, 94A29, 1996.

[8]. Xilinx, http://www.xilinx.comlproducts/design.

[9]. Xilinx PicoBlaze 8-bit Embedded Microcontroller userGuide. http://www.xilinx.comlsupport/documentation/user Jluides/ugl29.

[10]. Digilent Inc., Digilent Nexys2 Board Reference Manual

[11]. Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i

[12]. Xilinx Inc. Micro Blaze Reference Manual, version 10.1.

[13]. Xilinx Inc. Xilinx ISE and Xilinx EDK tools.

[14]. Spartan-3 Starter Kit Board User Guide, Xilinx, Inc.

[15]. Embedded System Tools Reference Manual, Xilinx, Inc

[16]. Sparta n-3 FPGA Family: Complete Data Sheet

[17]. Platform Studio User Guide, Xilinx, Inc.

[18]. Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i

[19]. Xilinx, Embedded System Example, XAPP433, version 2.2, 2006.