Resisting Blind Steganalysis in Real Time Covert Communication

Milav Dabgar
PG Student,
Dept. of ECE,
L.D. College of Engineering,
Ahmedabad, Gujarat, India.

Priti Muliya
Asst. Professor,
Dept. of ECE,
L.D. College of Engineering,
Ahmedabad, Gujarat, India.

Ashish Purani
Technical Associate (VLSI),
eInfochips Training and Research Academy (eiTRA),
Ahmedabad, Gujarat, India.

ABSTRACT
In this work, we review recent developments in the field of steganography keeping real time covert communication applications in mind. We then propose a microarchitecture to transform the existing one of most successful Steganographic technique, Yet Another Steganographic Scheme (YASS) to the hardware platform for satisfying the current Steganographic application needs. This hardware will be able to resist the Self-calibration based blind Steganalysis attacks, which are the most successful attacks in breaking any Steganographic technique till now.

General Terms
Information security, steganography, data hiding, watermarking.

Keywords
Covert communication, YASS, digital design, FPGA, JPEG, blind steganalysis.

1. INTRODUCTION
Today digital multimedia has become the most preferred form of information and in this Digital Information age information security is a major area of interest around the globe among researchers. Data hiding and data encryption are the two ways to achieve security. Steganography, a technique used for data hiding has been practiced since the early ages of human life, but the recent applications requires faster processing, portability, and real time processing support.

In this work we have first reviewed some steganographic techniques which provides significant undetectability in blind steganalyst's active warden framework, then we have reviewed some recent work focused on hardware realizations of various steganographic techniques. Finally we have filtered out the technique which are quite successful in resisting blind Steganalysis and are feasible in terms of hardware realization as well. The successful completion of the work will enable us to build special purpose dedicated IP core that can be reused to build up real covert communication system. We have arranged the document sections as like in Section 2, recent developments along with existing limitations are outlined. In Section 3, we propose approach to solve the problem with algorithm used is discussed. In Section 4, architectural overview is given along with description of the basic building blocks. We finally provide conclusion in Section 5.

2. RECENT DEVELOPEMNETS AND LIMITATIONS
Looking into the historical literatures work of steganography there are thousands of algorithms which have been proposed and within short enough time they have been broken by steganalyst. That is why the main concern of the steganographer has become to develop the algorithm which are quite tough to be broke at the same time real time processing needs have motivated researchers to work on transforming techniques to hardware platform.

2.1 Resisting Blind Steganalysis
The first breakthrough in the field of steganography was when in Dec. 2004, K. Solanki et al have published their work on Image Adaptive Algorithm with ECC in [1]. They have adopted information-theoretic analyses of the steganography with a view to hiding large volumes of data with low perceptual degradation under AWGN (Additive White Gaussian Noise) attacks. On the very next year, the same team of researcher at VRL have developed statistical restoration framework. In their following work in [2] they have tried statistical restoration of DCT coefficients and these resulted in giving exact replica of cover image pmf. This was verified by noting KL (Kullback—Liebler) divergence between two pmfs. As part of revision to [2] they have illustrated their Statistical Restoration framework for Quantization index modulation (QIM) based hiding in [3], this reduces detect ability while preserving the same robustness and severely affects the steganalysis performance of both DCT-histogram and blockiness methods.

The next big breakthrough was when A. Sarkar, the new research fellow joined VRL team. A. Sarkar et al have published work that resists blind steganalysis in [4], YASS (Yet Another Steganographic Scheme) works on the DCT coefficients just the same way as described in the above three papers by [1, 2 and 3] but with little tricky modification to solve the problem faced by them. Here they have devised a scheme that works by hiding data in random Selection of blocks within macro blocks and as consequence of it desynchronizes steganalyst. This algorithm can be used for active steganography as it defends against blind steganalysis and distortion attacks as well. In future work they have considered to increase hiding capacity, and to provide mechanisms for randomizing the embedding process. This upgrade resists medium level of attacks by keeping the same rates of hiding and finally results in lower detection rates too.

After five long years Xiang Yang et al have developed a technique for Effective Steganalysis of YASS in [7]. The existing Steganalysis methods for YASS detection were mostly specific techniques. Whereas blind steganalysis requires high dimensional feature vectors to detect YASS. Here [7] have proposed lower dimensional feature sets for blind steganalysis. As a countermeasure to work by [7], two new revisions to YASS is published [8, 9]. Specific Steganalysis probability on the proposed method is less than 59%, while it was about 95% and above on YASS. The most recent work published in August 2014 by Vivek Amruth et al in [10] highlights a method for providing security in Smartphones that makes it impossible to get the message without knowing the knowledge of the method used. They have proposed use of YASS steganography for compressing secret images.

2.2 Providing Hardware Realizations

In [13] B.J. Mohd et al have presented a FPGA implementation LSB steganography on Altera cyclone II FPGA. It has in build processor along with the programmed microarchitecture for steganography giving a balance between quality, hiding capacity and undetectability. . In [14], H.Y. Leung et al have designed and implemented a real time steganographic device that gives proper Selection of suitable technique for given cover media.

Ammar Odeh have presented a faster (nearly 11.27 Gbits/seconds) implementation of steganography for text in their publication [15]. In [16], E. Gómez-Hernández have given a steganographic technique called 'ConText' technique. They hide more data in edges of objects in the image, which is much more difficult to detect. It provides significant data hiding rate of 61.5 Mbits/second. H. Farouk and M. Saeb have proposed a hardware realization for video, audio or image steganographic model which works in real-time.

2.3 Summary

From brief literature survey we come to conclusions that, spatial domain techniques are simple and comes handy when the need is of less security & this algorithms are easy to implement on hardware too. Whereas transform domain techniques requires comparatively more computations and more complex to implement but it provides good enough security and robustness with reduced computational speed. No work has been reported till date which tries to achieve both the functionality together, that is resisting blind steganalysis in real time is still beyond the limits.

3. THE ALGORITHM STEGO

After doing a deep literature study of all the YASS related publication, we conclude that YASS algorithm with its revisions applied is still undetectable to both Specific and Blind Steganalysis techniques. In addition we found one very useful practical application proposal by [10] which again motivates us to stick with hardware realization of YASS on FPGA. Before going to digital design to provide the functionality, as a first step to solve the problem we state the algorithm which is inspired from the YASS streamline.

Begin:

1. Load Cover image pixel wise and message bit stream from UART through FIFO into Block RAM of FPGA.
2. Provide all the other parameters governing the embedding process i.e. pseed, dseed, B, per, Thresh, noused, design & output JPEG quality etc. either hardcoded or soft coded.
3. Perform color space conversion.
4. Partition the original image into Macro Blocks of size B*B, where B>=8, Then choose any single block randomly within the big block to load into cache memory. The key for this is shared between both the ends of communication link.
5. Now for each chosen block take 2D DCT, Divide by respective quantization matrix for QFh, Hide the data in selected band of low frequencies AC coefficients using QIM, Multiply it by respective quantization matrix for QFa and finally take 2D IDCT to get back the image in spacial form. Figure 1 illustrates the statistical restoration in QIM.
6. Store the block of image back into image Block RAM.
7. Repeat 5 and 6 till the image ends.

End

This procedure helps us in avoiding blind steganalysis to some significant extent.

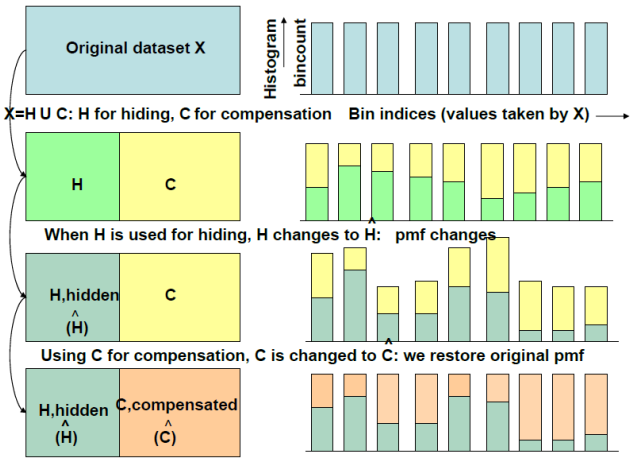

Figure 1: Statistical restoration in QIM [3]

4. THE MICROARCHITECTURE

In this section we give details on how the above explained steganographic algorithm can be transformed to a microarchitecture.

4.1 Architectural Overview

The system architecture of the proposed core is given in the Figure 2 below. This core interacts with the host CPU through the pins provided here. This block performs the function of embedding message data into image using the above described YASS algorithm. We now discuss pin function of it.

Clk, Rst and Start: Clk Provides necessary timing to the core, the operation are synchronized to this clock. The signal to this pin may be a derived clock form the host CPU or external addition crystal. Rst pin is for resetting the internal operations of the chip and return it to its ideal state. Start is 1 bit input signal which informs the core to start the steganography process. This signal usually receives a positive pulse from the host to start the embedding process.

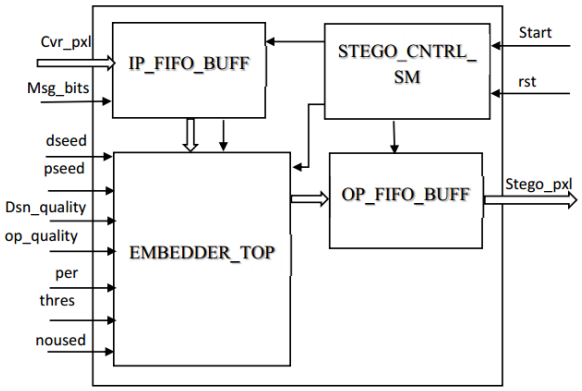

Figure 2: The Microarchitecture of the Algorithm Stego

Msg and Cvr_Pxl: This is the input message bit stream, we can give message bit stream equivalent of any content as input to this pin/s. The image is read in to the chip pixel by pixel. So for this process we need to store the image pixel values from unsigned integer into 8 bit format. We can read pixel by pixel whole image and store it into RAM within the chip to perform the block level operations.

Dseed and pseed: these are the keys or seeds to the random number generator and should be given by the host. Dseed is used to generate dither sequence and pseed is used to generate Sel vector which decides by doing permutations to which data hiding occurs and which pixel to be kept reserved for compensation.

Design_Quality and Op_Quality: these parameter refer to the resultant images quality after JPEG compression. We generally choose Design_Quality = 50 and Op_Quality= 75.

Per, Thres, and noused: These are the parameter specific to the YASS embedding. Per indicates what amount of coefficients to use for overall embedding process. Thres denotes the predefined Threshold value of JPEG coefficient for which we will go on hiding and noused indicated the number of coefficients used per 8*8 block.

Stego_Pxl: the final output JPEG Stego image which contains the hidden message bit stream is flushed out through this pins pixel by pixel.

The architecture is made up of some modules which we highlight here.

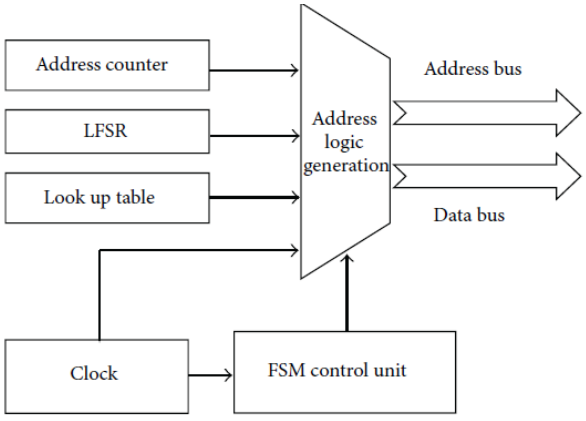

Figure 3: The Address Generator Module

Stego_Ctrl_SM & Address Generator: This is the main state machine which controls all macro level functions. It receives Clk, Rst and start signal from the host CPU. It gives SoC signal command start of conversion and in response receives EoC (end of conversion) signal when it's done with current frame. It gives FIFO_Rd and FIFO_Wr to FIFO buffer to control FIFO read and FIFO write operation. The address generator block provide necessary referencing to the memory modules and is controlled by the Controller.

Embedder_Top: This block is the implementation of the YASS embedding algorithm. It receives control signals from Stego_Ctrl_SM and data from host in the form of Cvr_Pxl and outputs a Stego_Pxl.

FIFO Buffers: This block are used to make embedding speed independent of I/O latency. It keeps loading the Cvr_Pxl values in advance and stores it in FIFO RAM and delivers when embedder asks for it. The functions of Op_FIFO_Buff is analogous to Ip_FIFO_Buff.

4.2 The Embedder Module

This Module can be broken down into below sub modules shown in the figure 4.

IRAMIF: A Block of RAM is required to store the image and the relevant computational parameters. The size of it will be decided after designing the all internal blocks. It will be comparatively spanning a larger area of our chip area.

Cache: A comparatively small but faster block of memory is initialized here which serves to store a current block of image on which the hiding operations are right now being performed. We will load and store the data in cache after operations from each block

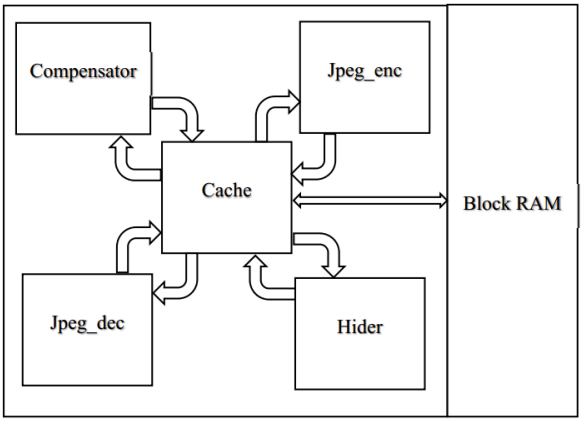

Figure 4: The Embedder Module

Jpeg_Enc: This block will load a block of pixel values and after performing standard JPEG compression of the taken Design_Quality, stores the JPEG coefficients block into cache.

Hider: This is the core block which actually takes jpeg block from cache message bits from cache and performs data hiding into JPEG coefficients.

Jpeg_Dec: This block will load a block of coefficient values and after performing standard JPEG decoding of the given Op_Quality, stores the uncompensated Stego pixel values of a block into cache.

Compensator: This performs the statistical restoration as was discussed in the earlier discussions.

This whole process repeats for all the blocks and at the output of Stego_Top we finally receive compensated Stego image pixels.

5. RESULTS

In this section we show the results obtained after implementation of algorithm in MATLAB software.

We have taken standard lena.jpg 512*512 gray scale color image as a cover image and have inserted random bit stream as a message into this image using our Matlab function for YASS embedding. The message can be anything text, image, audio, video, documents or anything that you can represent into bit stream. The code works for color as well as grayscale images of any size of any format (.jpg, .tiif, .bmp etc.).

Figure 5 shows the original JPEG Image 'lena.jpg', Figure 6 is the image obtained after data hiding, you can see that the image is perceptually identical to the original cover image.

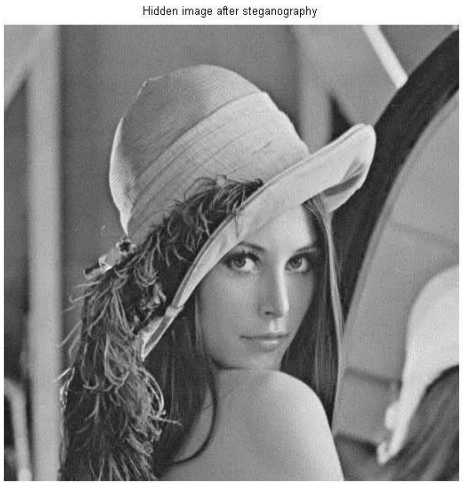

Figure 5: Original image lena.jpg

Figure 6: Stego image obtained after embedding random data bit stream

Figure 7 and Figure 8 shows original and final image histograms showing pmfs of cover and stego images respectively. It is clear from this results that we have successfully matched the stego image pmf to cover image pmf. Figure 9 is the histogram of the image before applying the statistical restoration.

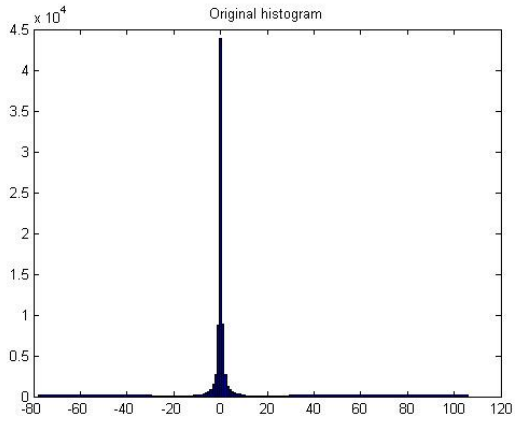

Figure 7: Original Image Histogram

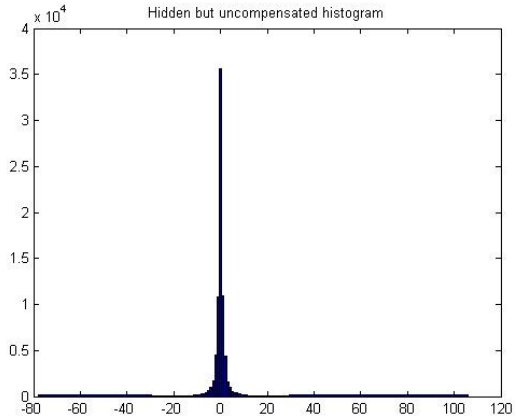

Figure 8: Final Image Histogram

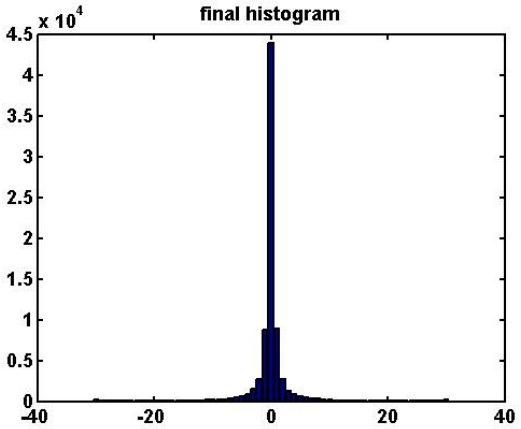

Figure 9: Histogram of Hidden but Uncompensated image

We have also shown the divergence between them in Figure 10 which shows a blank line, i.e. identical histograms.
The results are quite encouraging as we can exactly match the pmfs of cover and stego images exactly. Zero K-L divergence has been achieved, with very little perceptual distortion. This concludes our discussion on results of implementation.

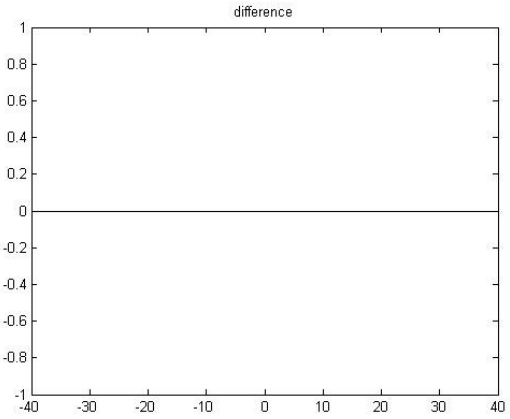

Figure 10: Histogram showing divergence between original & final image histogram

6. CONCLUSIONS AND FUTURE WORK

In this we have developed a microarchitecture which can resist blind steganalysis in real time, which has never been attempted till date. This techniques requires comparatively more computations and more complex to implement but it provides good enough security.

However we have just achieved functional requirements till now in the design part. We will optimize physical parameters like logic gates number in RTL level and pipeline to reduce time delay and then implement it in FPGA. The Implementation should be tested further by emulating more different blind and specific attacks.

7. REFERENCES

[1] K. Solanki, U. Madhow, B.S. Manjunath, S. Chandrasekaran and N. Jacobsen, "Robust image adaptive data hiding based on erasure and error correction", IEEE Transactions on Image Processing, vol. 13, no. 12, pp. 1627-1639, Dec. 2004.

[2] K. Solanki, U. Madhow, B.S. Manjunath, S. Chandrasekaran and K. Sullivan, "Statistical restoration for robust and secure steganography", in Proc. IEEE International Conf. on Image, Processing, Genova, Italy, vol. 2, pp. 1118-1121, Sep. 2005.

[3] K. Solanki, U. Madhow, B.S. Manjunath, S. Chandrasekaran and K. Sullivan, "Provably secure steganography: Achieving zero K-L divergence using statistical restoration", in Proceedings IEEE Int. Conf. on Image Processing, Atlanta, GA, USA, pp. 125-128, Oct. 2006.

[4] A. Sarkar, K. Solanki and B.S. Manjunath, "YASS: Yet Another Steganographic Scheme that Resists Blind Steganalysis", in Proc. 9th Int. Workshop on Information Hiding, Saint Malo, Brittany, France, pp. 16-31, June 2007.

[5] A. Sarkar, K. Solanki, and B.S. Manjunath, "Further Study on YASS: Steganography Based on Randomized Embedding to Resist Blind Steganalysis", in Proc. SPIE - Security, Steganography, and Watermarking of Multimedia Contents X, San Jose, California, vol. 6819, pp. 681917-681917-11, Jan. 2008.

[6] A. Sarkar, U .Madhow , B.S. Manjunath "Matrix Embedding With Pseudorandom Coefficient Selection and Error Correction for Robust and Secure Steganography"; IEEE Trans. Info. Forensics and Security, vol.5, no.2, pp. 225-239, March 2010.

[7] Xiang Yang, Zhang Wen-hua "Effective Steganalysis of YASS Based on Statistical Moments of Wavelet Characteristic Function and Markov Process"; International Conference on Computer Science and Electronics Engineering (ICCSEE), Volume:1, pp. 606 – 610, March 2012.

[8] Xianming Lv, Lihong Ma, Jing Tian "An Improved YASS Approach Using Irregular Host-Blocks and Modified Quantization Index Modulation "; in Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 309 –312, July 2012.

[9] Li-Hong Ma, Xian-Ming Lv, Jing Tian "Modified YASS algorithm with virtual host block Selection and model based embedding "; in 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 73 – 78, July 2012.

[10] Vivek Amruth, Amrita P "Multi-Level Steganography for Smart phones"; in First International Conference on Networks & Soft Computing (ICNSC), pp. 81 –84, Aug. 2014. 51

[11] Anindya Sarkar, PhD Thesis, "Novel Image Data-Hiding Methodologies for Robust and Secure Steganography with Extensions to Image Forensics" university of California - Santa Barbara, July 2012.

[12] Piyush Goel, M.Tech. Thesis, "Data Hiding in Digital Images: A Steganographic Paradigm" Indian Institute of Technology–Kharagpur, May, 2008.

[13] B. J. Mohd, S. Abed, T. Al-Hayajneh, and S. Alouneh, "FPGA hardware of the LSB steganography method" in International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1-4, May 2012.

[14] H.Y. Leung, L.M. Cheng, L.L. Cheng, Chi-Kwong Chan, "Hardware Realization of Steganographic Techniques" in Third international conference on Intelligent information hiding and multimedia signal processing, pp. 279-282, November 2007.

[15] Ammar Odeh, Khaled Elleithy, and Miad Faezipour, "Fast Real-Time Hardware Engine for Multipoint Text Steganography" in Systems, Applications and Technology Conference (LISAT), pp. 1-5, May 2014.

[16] E. Gómez-Hernández, C. Feregrino-Uribe, and R. Cumplido, "FPGA hardware architecture of the steganographic context technique," in 18th International Conference on Electronics, Communications and Computers, pp. 123-128, 2008.

[17] H. Farouk and M. Saeb, "Design and implementation of a secret key steganographic microarchitecture employing FPGA," in Proceedings of Design, Automation and Test in Europe, pp. 212-217, 2004.

[18]