# Low cost data authentication scheme and hardware design

*Santi P. Maity*[1], *PrasantaK.Nandi*[2], *MalayK.Kundu*[3]*andAyanBanerjee*[4]
Bengal Engg. & Sc. University, Shibpur, P.O.-Botanic Garden, Howrah, India, 711 103[1,2,4]
Indian Statistical Institute, 203 B. T. Road, Kolkata, India, 700 108[3]
*spmaity@telecom.becs.ac.in*[1], *pkn@cs.becs.ac.in*[2], *malay@isical.ac.in*[3], *ayan@telecom.becs.ac.in*[4]

## Abstract

Emergence and rapid expansion in the world wide web (WWW) has created a challenging problem related to the security in digital multimedia signal transmission. Over the years digital data hiding schemes are being used widely for copyright protection, authentication and integrity verification of the digitized media. The paper proposes an algorithm that employs channel coding scheme for hiding a gray scale image like information within a gray scale cover image. Implementation of the proposed low cost algorithm may be speeded up significantly by hardware realization and circuit can be integrated into the existing digital still camera framework.
*Index Terms:* Channel coding, data hiding, hardware design, negative modulation

## 1 Introduction

Digital data hiding techniques with its various names and forms are being used widely for a variety of applications ranging from copyright protection, data authentication, broadcast monitoring, security in communication to data indexing etc. by allowing an auxiliary message embedding within a digital multimedia signal [1][2]. Data embedding in digital images can be implemented either by modifying the pixel values (spatial domain) of the cover image or the coefficients (transform domain) obtained by using various transformations such DFT [3], DCT [4], Walsh-Hadamard [5], and wavelets [6] etc. The latter approaches offer higher resiliency over the former at the cost of high computation that very often makes their hardware realization complicated and thus the schemes become unsuitable for real time applications.

Hardware implementation of data hiding scheme offer advantages over software realization in terms of less area, low execution time, and less power. If a chip is fitted in the digital devices, the output video or images can be marked right at the origin although the same can be done using software after those videos or images downloaded to the computer. But, in this case embedding software will take more time compared to hardware. The example of TV broadcast will highlight the significance where digital media is to be marked in real time and hardware is the only solution.

In this paper a spatial data embedding scheme in digital images and hardware implementation are proposed. The algorithm can be used for real time transmission of a message through digital media. Reliability in data transmission is further improved by using channel coding scheme in the form of variable redundancy that is incorporated among the different bit planes of the message signal based on their relative significance.

The paper is organized as follows: Sections 2 and 3 describe the algorithm and the hardware realization respectively. Sections 4 and 5 present some experimental results and conclusions respectively.

## 2 The proposed algorithm

The cover image $I$ is a gray-scale image of size $(N \times N)$ where $N = 2^p$ and the auxiliary message $W$ is also a gray image of size $(M \times M)$ where $M = 2^n$. Results are reported here considering 4 bits/pixel gray image of size $(64 \times 64)$ as message and $(256 \times 256)$, 8 bits/pixel gray image as cover data.

### 2.1 Transmitter operation

The gray-scale image like message signal is spatially dispersed by using a pseudo random (PN) sequence of length $l$=$(M \times M)$ and is converted into binary string. An extended binary string is formed by incorporating variable redundancy. Higher redundancy is assigned to the higher bit plane since they contain visually significant data and less or no redundancy for lower order bit planes that contribute more subtle details in the image [7]. In the present case, 4-th bit plane of the auxiliary message is repeated nine (9) times, 3-rd bit five (5) times, and no redundancy for the remaining two LSBs (least significant bit). The extended binary message string is then encrypted using another pseudo random (PN2) sequence. Each bit of the encrypted binary string

replaces suitable LSB (least significant bit, say 3-rd or 4-th) plane of the cover image pixel values. To accommodate the effect of possible attack like low pass filtering and for possible survival of the embedded information, further bit manipulation is done according to some predefined norms by exploiting spatial masking effect of suitable size. The scheme is implemented by estimating the tendency of possible change in gray value after the attack like mean filtering. The process is called here as negative modulation.

### 2.2 Receiver operation

The stego image, with or without external attack, is transformed to one-dimensional sequence of the pixel values. The embedded data is picked up from the appropriate bit plane and the string is then decrypted using the pseudo noise code (PN2). Each decrypted substring of length 16 is then partitioned into sub substrings (smaller substrings) based on the degree of redundancies incorporated on the different message bits. Binary detection is then applied for each sub substring based on the majority decision rule i.e. if more than 50% symbols of a sub substring are '1', decision for decoding is '1' otherwise '0'. The similar decoding process is applied for all the sub strings and the bits of a sub string are converted to a decimal equivalent value. The values represent the pixel value of the decoded message signal. The number of positional mismatch in the symbols between the strings 1 (embedded bit string) and 2 (decoded bit string) are counted and are divided by the total number of symbols in the string in order to calculate the probability of error denoted by $p(e)$. If a sub string consists of $l$ number (an odd number) of symbols and the string consists of total $k$ number of such sub strings, $P(e)$ that denotes the probability of making wrong decision for all the sub strings can be expressed as follows:

$$P(e) = (\sum_{n}^{l} \binom{l}{n} {p_e}^n (1-p_e)^{l-n})^k \qquad (1)$$

where $n = (l+1)/2$. All $k$ number sub strings are assumed to be independent among each other. Lower value of $P(e)$ indicates better robustness of the algorithm and the value is zero when there is no message loss due to external attacks.

## 3 Hardware design

The hardware design of spatial data embedding technique consists of two main parts, the transmitter and the receiver.

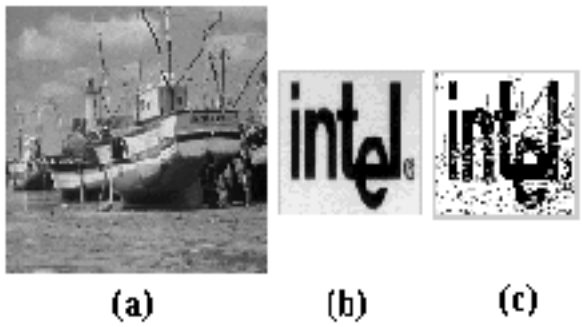


Figure 1: (a) Cover image; (b) message signal; (c) decoded message after noise addition in stego

### 3.1 Transmitter design

The major sub blocks in the transmitter , as shown in Fig. 2, are bit select unit, message extension unit, control unit, data embedding unit.

*Bit select Unit* selects bit plane for data embedding (using the ST [0:3] bus), and allows message bit to flow into the embedding unit at the desired time (acting as the MUX control input). The required components for the unit are three (3) T flip-flops (FF), one (1) 3 to 8 line decoder, one (1) 8 to 1 line MUX, three (3) inverters.

*Message extension unit* transforms 4-bits of one pixel value of the message signal to 16 bits by adding redundancy. At the beginning of embedding cycle (when 1 pixel value of the message is completely embedded), 4 bits are loaded into a 4-bit parallel-in-serial out (PISO) shift register. The output is taken from FF in which 4-th bit of the message pixel is initially loaded. A binary counter that counts from 0-15 and a logic circuit together control the operation when the bits will be shifted in the register. First CLOCK ENABLE (CE) signal comes from the control unit when the counter has counted up to 8 since 4-th bit of the message pixel is repeated 9 times. Control circuit provides the next high CE pulse when the counter counts 13 since 3-rd bit of the message pixel is repeated 5 times. Now 3-rd bit is shifted out to let the 2 bits to come to the output. The next two shifts occur on the counts of 14 and 15 giving the last two bits which were to be repeated once as output. The required hardware components for this unit are four (4) D FF, one (1) 4-bit binary counter, four (4) 2 to 1 line MUX, one (1) 2 input AND gate, one (1) 3 input AND gate, two (2) 4 input AND gate, one (1) 3 input OR gate.

*Control unit* controls the loading of bits for both the message signal and the cover image by enabling the mux within these units at the particular time

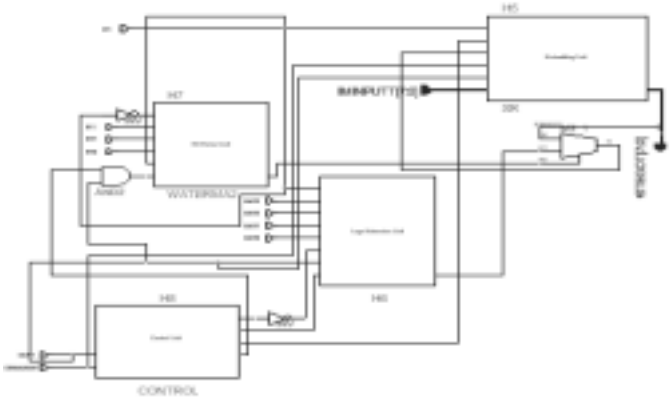

Figure 2: Block diagram of transmitter

instant and acts as the master clock of the system. This unit is provided with an 8-bit counter that can count one complete hiding cycle i.e. (16*9 cycles). The control unit provides various pulses to the other units to enable their operations during this time interval. The unit provides a LOW pulse to the embedding unit at every 9-th clock cycle so that the parallel data from the cover image is loaded at its input. At the beginning of each embedding cycle it sends a LOW pulse to the message extension unit to load the message pixel to be embedded in that cycle. It also sends one more LOW signal to the message extension unit just before sending the message load pulse to synchronize the internal counter of this unit with the load cycle. The total counting cycle of the counter is not required for the operation. So after a message pixel value is embedded, the counter is reset so that it can start afresh for a new embedding cycle. The required hardware components are one (1) D FF, one (1) 8-bit binary counter, eleven (11) inverter, two (2) 2 input AND gates, four (4) 2 input XOR gate, six (6) 2 input XNOR gates, one (1) 4 input AND gates, two (2) 5 input AND gates, one (1) 2 input OR gate, one (1) 8 input OR gate, two (2) 8 input NOR gate.

*Embedding unit* embeds the message bits after getting from the message extension unit in the bit plane of the cover image as specified by the bit select unit. It consists of an 8-bit loadable shift register (SR) with feedback of the 4-th bit to the serial unit. Image bits are loaded in parallel at every 9-th clock cycle as guided by the control unit. For the next 8 cycles, the bits are shifted across register with the message bit being loaded into the register at time indicated by the bit select unit. This bit replaces a particular image bit and continue shifting through the SR. One full rotation is complete after 8 clock cycles. The image bits return to their original positions after the message bit inserted in the specified position. The embedded bit is taken out in parallel-simultaneously the next pixel bits of the cover image are loaded into the unit. The required hardware components are eight (8) D FF, and eight (8) 2 to 1 line MUX.

### 3.2 Receiver design

The major sub blocks in the receiver are *Bit extraction unit*, *Decision making unit*, and *message formation unit*.

In *bit extraction unit*, message bit was extracted from the proper bit (where message bit was embedded) of the 8-bit data. The data is loaded into 8-bit parallel -in-parallel out register and the proper bit is taken from the output of a 8:1 multiplexer. The required hardware components are 8-bit loadable register, 8 to 1 multiplexer.

In *decision making unit*, the redundancy incorporated in the transmitting side is removed to form 4-bit message pixel from the 16 bit extracted data obtained from bit extracting unit. One 4 bit counter is used to count how many 1's have been received from the first 9 received bits, the other 4-bit counter will count the same from the next 5 bits. One 2 to 4 decoder is used to select which counter will be enabled at what time. Thus the decoder select input is taken from a logic circuit that will generate 00 for first 9 clock cycle, 01 for neat 5 clock cycle, 10 and 11 for the next two. The decoder input comes from the previous stage. The required hardware components are one (1) 2 to 4 decoder, three (3) 4-bit binary counters, and six (6) 2-input OR gates, two (2) 2-input AND gates, five (5) 3-input AND gates (having one or two inverted inputs), four (4) 4-input AND gates (having one or two inverted inputs), and XOR gate.

Each message pixel consists of 4-bits and bits are stored in serial in parallel out shift register. One 4 to 1 MUX is used for storing the proper bit at proper time. The respective bit is stored at proper time serially into the shift register by proper controlling the MUX input. A D flip-flop is used for resetting the shift register at proper time instant. CE (clock enable) terminal is controlled by some logic circuit for controlled loading of the register. The required hardware components are one (1) 4 to 1 MUX, one (1) serial in parallel out shift register, 1 D flip-flop, one (1) 4 input OR gate. Fig. 5 shows the hardware design of receiver circuit.

## 4 Result and discussion

The algorithm has been implemented over a large number of bench marked images [8] and various forms of common image processing operation as well as deliberates attacks have been simulated in stego

data. It has been seen that extracted messages are quite recognizable even after greater depth of degradations like linear and non linear spatial filtering, sharpening, histogram equalization, lossy JPEG and JPEG 2000 compression, noise addition, rescaling etc. occurred in the cover image.

The quality of the extracted message is quantified by probability of bit error $P(e)$ and the mutual information value $I(X;Y)$. The expression of $I(X;Y)$ is represented as follows where random variables $X$ and $Y$ represent the transmitted and decoded message respectively. If $p(x_i)$ represents the probability of occurrence of the i-th pixel value in the transmitted message and $p(y_j/x_i)$ represents the channel transition matrix, $I(X;Y)$ that represents the average amount of information received from the signal degradation, can be expressed as follows [9]:

$$I(X;Y) = \sum_i \sum_j p(x_i)p(y_j/x_i) \log \frac{p(y_j/x_i)}{\sum_i p(x_i)p(y_j/x_i)} \quad (2)$$

where $i, j$ represent the index of the symbols.

Fig. 1(a), 1(b) and 1(c) represent the cover image Fishing Boat [8], the message and the extracted message respectively where stego data is corrupted by additive gaussian noise with variance 0.05. The entropy of the message, shown in Fig. 1(b), is 0.867 while the $I(X;Y)$ value for the decoded message, shown in Fig. 1(c), is 0.54. The $I(X;Y)$ values for the decoded messages are 0.46 and 0.49 when the stego images undergone spatial mean and median filtering using window sizes $(3 \times 3)$. The values of the same objective measures for the decoded messages are 0.41 and 0.38 when the stego images undergone lossy compression operations JPEG and JPEG 2000 respectively at quality factors 75.

Figs. 3, 4, and 5 show the hardware design of the control unit of the transmitter, embedding unit, and the receiver circuit respectively. The other units of the transmitter and receiver circuits are also designed (not shown) and required hardware components of all the units are already mentioned. The design is implemented using XILINX SPARTAN series FPGA (Field Programmable Gate Array).

# 5 Conclusions

An algorithm for covert image-in-image communication and its hardware realization is proposed in the paper. Channel coding scheme based on the relative significance of the different bit planes of the gray scale message improves robustness efficiency against common signal processing operations as well as deliberate external attacks. The algorithm can be applied for integrity verification, authentication, security in data communication and copyright protection of the digitized media. Although only tested on images, there is no inherent reason why the approaches described here would not work for one-dimensional audio signals or video sequences. Algorithm requires few simple computations and digital circuit realization allows its application for real time multimedia data transmission.

Table 1: Probability of error in single bit, 4-th bit and 3-rd bit position of the decoded message due to mean filtering using window size of $(3 \times 3)$

| Length of sub string | Prob. of bit error $p(e)$ | Prob. of wrong decision in 4-th bit $P(e_1)$ | Prob. of wrong decision in 3-rd bit $P(e_2)$ |
|---|---|---|---|
| 4 | 0.107742 | 0.000663 | 0.009957 |
| 16 | 0.148758 | 0.002734 | 0.023853 |
| 64 | 0.219437 | 0.013599 | 0.064379 |
| 256 | 0.32301 | 0.053181 | 0.154462 |

**References**


[1] B. Macq, Special issue on identification and protection of multimedia information, *Proc. IEEE* 87: 1673-1687, 1997.

[2] C. D. Vleeschouwer, J. F. Delaigle and B. Macq, Invisibility and application functionalities in perceptual watermarking-an overview, *Proc. IEEE* , vol. 90, pp 64-77, 2002.

[3] J. O. Ruanaidh and T. Pun, Rotation, scale and translation invariant digital image watermarking, *Proc. ICIP*, Atlanta, GA, pp. 536-539, October, 1997.

[4] C. T. Hsu and J. L. Wu, Hidden digital watermarks in images, *IEEE Transactions on Image Processing*, vol. 8, pp. 58-68, 1999.

[5] S. P. Maity and M. K. Kundu, An image watermarking scheme using HVS characteristics and spread transform, *Proc. ICPR-2004*, vol. 4, pp. 869-872, August 23-26, 2004.

[6] S. P. Maity and M. K. Kundu, A blind CDMA image watermarking scheme in wavelet domain, *Proc. ICIP 2004*, Singapore, pp. 2633-2636, October 2004.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, New York, 1992.

[8] http:// www.cl.cam.ac.uk/ fapp2/watermarking.

[9] B. P. Lathi, *Modern Digital and Analog Communication Systems*, 3rd edition, Oxford University Press, New Delhi, India, 1999.

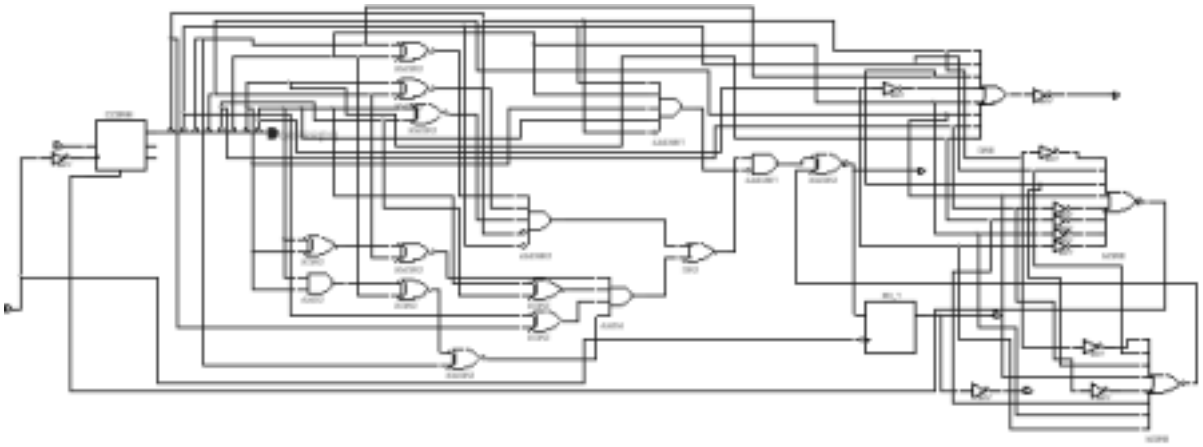

Figure 3: Hardware design of control unit at transmitter

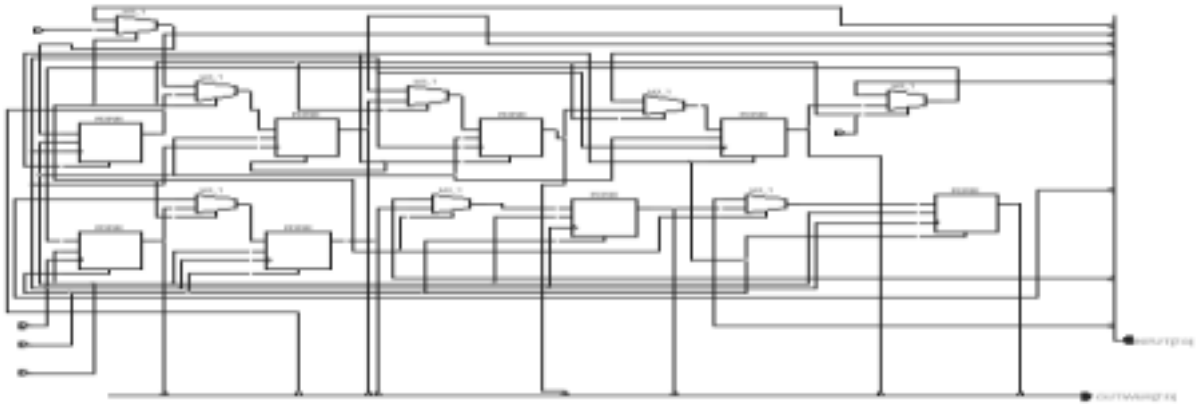

Figure 4: Hardware design of embedding unit

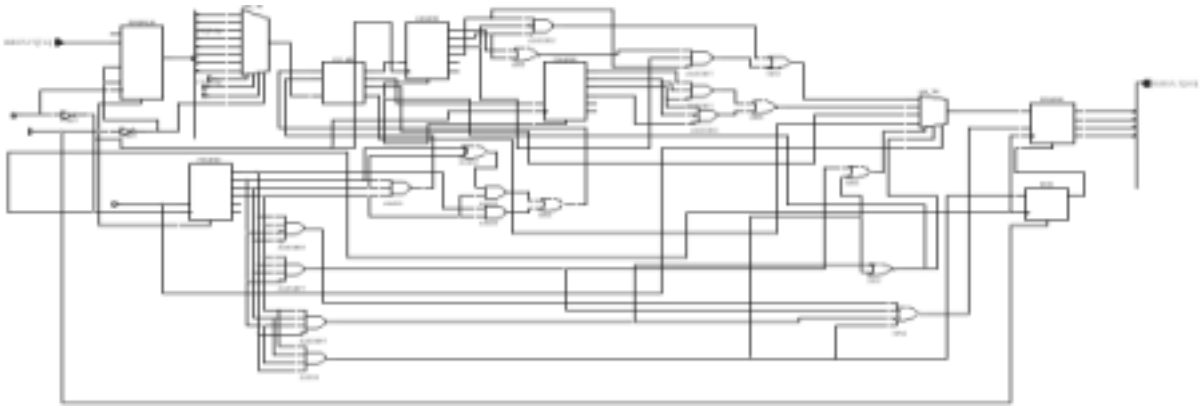

Figure 5: Hardware design of the receiver circuit