A Joint Source-Channel Steganography Decoding Paradigm and It’s Implications on Capacity

Ishtiaq Saaem and Hong Man
Visual Information Environment Lab (VIEL)
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, New Jersey 07030 USA

Received: November 11, 2005   Accepted: Nov. 25, 2005

ABSTRACT
Steganography has become extremely popular lately as a means of achieving security for many multimedia applications. However, research on stego-attack mechanisms has also kept pace with this development. Robustness to such stego-attacks usually requires a compromise in stego-capacity. In this paper we propose a joint source-channel coding paradigm for attack robustness with small overhead. We test our framework using a least significant bit based embedding algorithm and random additive attacks. Our results lead us to believe that the joint source-channel coding paradigm has significant potential in this area.

I.  INTRODUCTION
The rise in the popularity of multimedia content rich applications has increased the need to address the security issues involved in delivering multimedia over different transmission channels. Steganography is one such security paradigm that is currently receiving significant interest [1]. Steganography is the science of embedding one media type within another for various applications like copy protection, fingerprinting, authentication, covert communication etc. As with any other security paradigm, attacks have been designed for steganographic algorithms as well [2]. In this proposal we will study the problem of enhancing steganographic capacity via intelligent attack adaptive detection. Note, we use the terms “detector” and “decoder” interchangeably.

II. ATTACKS AND THEIR IMPACT ON CAPACITY
Voloshynovskiy et al. [2] classify stego-attacks into four categories, namely, removal, geometric, cryptographic and protocol attacks. Removal attacks aim at removing the embedded signal without cracking the security of the embedding algorithm while geometric attacks aim to distort the detector synchronization with the embedded signal—Unzign and Stirmark [3]. Public domain software used for watermark benchmarking, integrate a variety of geometric attacks. Simple examples of geometric attacks include pixel shifting, scaling, rotation and downsampling. Cryptographic attacks on the other hand actually aim at cracking the embedding scheme itself. Practically, such attacks are rare due to the inherent computational complexity. Protocol attacks aim at attacking the entire concept of the watermarking application. For example, the copy attack is aimed at estimating the hidden signal from a stego-signal and then copying it on to another target data [4]. One may also classify these attacks as intentional or unintentional depending on whether the attacker intended to defeat the purpose of watermarking or not. Examples of the
second type of attack could include compression and downsampling. Attacks often cause complete loss or severe distortion of the hidden message. Robustness for these attacks can be achieved either using smarter embedding techniques or better detection techniques. In some cases robustness can be achieved by designing intelligent detectors that anticipate some types of attacks. We hypothesize that given some minimal information about the embedded signal and attack statistics, we can design a robust detector that will cause a significant improvement in payload. This paper proposes one such detector.

Embedding algorithms usually trade-off capacity in order to achieve robustness [5]. In fact, attacks have often been modeled in the literature as a communication channel [6] and forward error correcting codes (FECs) have been used to make the watermark robust to these attacks [7]. This method has a drawback in that FECs offer robustness at the cost of additional bits. This means that the effective payload for watermarking is reduced. To alleviate this burden on the payload, one may compress the watermark before embedding; however, there is still an incentive to search for alternative measures for robustness that does not decrease the payload significantly. In fact, compressing the hidden message may, in some cases, lead to increased vulnerability to attacks.

Our algorithm uses the joint source-channel coding paradigm [8] to approach this problem. We model the attack as a binary symmetric channel with attack probability $\varepsilon$.

III. PROPOSED DETECTOR

We model the hidden message as a first order Markov process compressed using an entropy code (variable length code). Our problem is to find the best possible estimate of the hidden message given the statistics of the attack and the hidden message. Let $C^n = \{c_{j,n}\}_{j=1}^n, \forall j$ denote the set of all possible hidden messages of length $n$, and let $R^n = \{r_{j,n}\}_{j=1}^n, \forall j$ represent the hidden message after the attack. Then, from statistical theory of communication, we know that the maximum a posteriori (MAP) probability detector is also the minimum probability of error detector. Mathematically the most likely hidden message $\hat{C}^n = \{\hat{c}_{j,n}\}_{j=1}^n, \forall j$, is given by

$$\hat{C}^n = \arg \max_j (\Pr(R = r^n_j | C = c^n_j))$$ (1)

As mentioned earlier, the embedded message is often compressed in order to increase the payload. When VLCs are used to compress the hidden message, Equation (1) becomes

$$\hat{c}_{j}^{n(j)} = \arg \max_{c_{j}^{n(j)}} \Pr(R = r_{j}^{n(j)} | C = c_{j}^{n(j)}) \Pr(C = c_{j}^{n(j)})$$ (2)

where $B$ is the number of bits in the embedded message and $n(j)$ is the number of words in the message $c_{j}^{n(j)}$. The number of words in the message will differ with the partition of the bit sequence because of the use of VLCs before embedding.

If the attack probability is $\varepsilon$, and the source can be modeled as a first order Markov process, then the most likely hidden message $\hat{c}_{j}^{n(j)}$, is obtained by maximizing
Figure 1. Example State-Space for the Codebook: 0, 10, 110, 111.

where, $\Pr(c_{j,1})$ is the probability that the first word is $c_{j,1}$, $d_H[c_{j,k},r_{j,k}]$ is the Hamming distance between the $k^{th}$ transmitted and received word, $\Pr(c_{j,k}|c_{j,k-1})$ is the probability that codeword $c_{j,k-1}$ preceded $c_{j,k}$ and $l_k$ is the length of the $k^{th}$ codeword.

IV. THE STATE SPACE AND THE ALGORITHM

A brute force approach to the above maximization is computationally inefficient and hence we cast this as a dynamic programming problem. The variable length nature of the hidden data implies a need for a specially constructed state-space [8]. This state-space consists of two classes of states: the complete and incomplete states. The detector is in a complete state if the most recent bit received was in fact the last bit of a particular codeword and in an incomplete state otherwise. The degree of incompleteness is simply the number of bits of an incomplete codeword that has already been received by the detector. During the decoding process, the detector moves from state to state with the advent of each new bit and a trellis can be used to show the time-evolution of this state space.

At each stage of the trellis, there are as many nodes as there are states. The state-space and trellis diagram are depicted for an example Huffman code, in Figures 1 and 2.

Figure 2. Trellis Diagram for the Codebook 0, 10, 110, 111.
We now, briefly, review the important steps of the MAP algorithm for variable-length encoded first order Markov sources. Two operations take place at each stage of the decoding algorithm. One consists of examining the metrics involved for all paths, entering each node (at the stage-stage pair) in the trellis and the other consists of looking for a merger of the paths and the actual declaration of decoded words (if there is a merge). For complete states, the path update step involves finding the best path metric leading to the state and retaining it; whereas, for the incomplete states, we must retain the metric values of all the paths back to the last complete state. The complexity of a trellis algorithm is generally considered to be equal to the size of the state-space: \( N + l_{\text{max}} - 1 \). However, as the incomplete states involve only a copy operation, the complexity is actually somewhat less. Also, when the greatest common divisor of the codeword lengths \( g \) is greater than one, further reductions are possible, since we need only consider \( g \) samples at a time. A formal statement of the algorithm, in the context of error resilience, can be found in [8].

We now, briefly, review the important steps of the MAP algorithm for variable-length encoded first order Markov sources. Two operations take place at each stage of the decoding algorithm. One consists of examining the metrics involved for all paths, entering each node (at the stage-stage pair) in the trellis and the other consists of looking for a merger of the paths and the actual declaration of decoded words (if there is a merge). For complete states, the path update step involves finding the best path metric leading to the state and retaining it; whereas, for the incomplete states, we must retain the metric values of all the paths back to the last complete state. The complexity of a trellis algorithm is generally considered to be equal to the size of the state-space: \( N + l_{\text{max}} - 1 \). However, as the incomplete states involve only a copy operation, the complexity is actually somewhat less. Also, when the greatest common divisor of the codeword lengths \( g \) is greater than one, further reductions are possible, since we need only consider \( g \) samples at a time. A formal statement of the algorithm, in the context of error resilience, can be found in [8].
Figure 5. PSNR of extracted stegomessage with and without MAP.

Figure 6. Bit error probability with and without MAP.
V. EXPERIMENTAL SETUP AND RESULTS

For our experiments we used the 256x256 “Lena” image as the host and 128x128 “Barb” (quantized using a 41 level mid-tread quantizer with step size of 6.2) and Huffman encoded (at 4.9bpp) as the hidden image. The embedding mechanism is least significant bit (LSB) based, which we assume is known to the attacker. The encoder set-up is depicted in Figure 3.

The attack probability, $\varepsilon$, is estimated using pilot symbols embedded within the image. The LSBs of the first few pixels in the image are reserved for the pilot symbols (which can be either 1 or 0). The receiver then compares the relevant LSBs with the received version and estimates the attack probability from it. If added security is needed, the pilot symbols may also be embedded using some key known to the sender and the receiver. Then the LSBs relevant to the hidden message are extracted, and the MAP detector is applied to it. We assume that the source statistics are sent to the receiver separately. We also note that this is marginal overhead in comparison to using a forward error correcting codes (FECs) to achieve robustness. Figure 4 depicts the decoder module.

Figure 5 shows the performance of the proposed MAP detector and the Huffman decoder in terms of the PSNR of the recovered hidden image. As can be seen, the proposed detector does significantly better than the Huffman decoder.

Figure 6 shows the percentage of extracted bits that are in error, which can also be regarded as detection bit error.
probability. The results obtained here are an average of 8 realizations of attacks. We again emphasize that this improvement has been achieved without the use of FECs and with only a small overhead, implying that the capacity has been enhanced by this method.

Finally, Figure 7 shows a more perceptual comparison of the proposed method and the Huffman decoder.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a new paradigm for robustness against attacks in a stego-system. Our results show that the proposed framework based on joint source-channel coding seems to be promising in this area. This paradigm presents positive implications for capacity enhancement in robust steganography. Future work will involve modeling more complex embedding and attack processes and designing joint source-channel coding paradigm based on these models.

ACKNOWLEDGEMENTS

This work was partly funded by the Stevens Wireless Network Security Center.

REFERENCES

University of Northern Iowa
Students First

- more than 120 majors
- 850 faculty dedicated to your success
- welcoming, friendly campus
- undergraduate research opportunities
- new facilities for wellness, science, performing arts and student activities
- excellence in teaching with renowned programs in business, music, science, education and more

www.uni.edu

Admissions: 800-772-2037