2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Hiding Secret Messages In Huffman Trees

Philip C. Ritchey
Department of Computer Science
Purdue University
West Lafayette, IN USA
pritchey@cs.purdue.edu

Vernon J. Rego
Department of Computer Science
Purdue University
West Lafayette, IN USA
rego@cs.purdue.edu

***Abstract*—A novel scheme for hiding information in Huffman trees is presented. Huffman trees are graphical representations of Huffman codes, which are optimal prefix-free codes generated by a simple algorithm given the distribution of the symbols. Huffman codes are used by MP3 and JPEG file formats for compressing audio and image files, but are also used for text documents and other digital media. The capacity, robustness and security of the information hiding scheme are analyzed and measured empirically. The results show that the scheme achieves a low capacity relative to the size of the cover-object and that the channel is not robust against an active adversary which knows about the channel. The security results show that an informed passive adversary's detection accuracy is greater than 90% if she has access to the symbol distribution and equivalent to pure guessing otherwise.**

***Keywords*-information hiding; steganography; covert channels;**

I. Introduction

Steganography is a term that refers to techniques whose goal or purpose is to enable communication between two parties in such a way as to prevent any unintended recievers, such as third-party adversaries, from discovering the existence of the secret, or even of the communication channel itself. The art of steganography is more than 2000 years old, with two examples found in the Histories of Herodotus [1]. The first recorded usage of the term occurs in Trithemius' Steganographia, a book written circa 1499 CE which is ostensibly about magic but which is, in fact, a treatise on crytpography and steganography [2]. Before computers and the digital age, steganography was limited to physical techniques, e.g. Histiaeus tattooing a slave's shaved head and waiting until the hair grows back to send the slave to deliver the message or Mary Queen of Scots' letters smuggled in and out of her confinement in barrels of beer [3]. However, the complexity and enormous resources offered by computers and high speed communication networks proved to be very fertile ground for the science of steganography to grow into the vast and varied field that it is today.

Huffman codes are optimal prefix-free codes for a given distribution and can be constructed according to a simple algorithm developed by Huffman [4]. The codes are optimal in that they produce the shortest expected codeword length. Huffman coding is used as a back-end to multimedia file formats, such as MP3 for audio and JPEG for images, but can also be used on text and for general purpose compression [5]. A Huffman code can be represented graphically by a tree, referred to as a Huffman tree, and the codebook and tree are interchangeable.

Chen et al. present two methods for embedding hidden messages using modified Huffman trees, in which the message is hidden in a Huffman-encoded object by prefixing [6] or suffixing [7] the codewords with bits of the secret message. However, examination of the Huffman tree will immediately reveal that an information hiding scheme is being used. In both cases, testing whether or not the Huffman tree contains duplicate symbols will reveal whether or not the compressed object is hiding a secret message.

The information hiding scheme presented here hides information in the structure of the Huffman tree as opposed to the content. Thus, the tree does not contain duplicate symbols which would immediately reveal its true nature. It will be shown that, under the right circumstances, the stego-tree is indistinguishable from a clean tree for the same content. Since the first priority of a covert channel is to remain hidden, the scheme presented here does not hesitate to trade capacity for increased security.

The remainder of this paper is organized as follows. Section II defines the information hiding scheme and presents algorithms for implementing it. Section III presents analytical results on the capacity and robustness of the scheme as well as experimental results for the security of the scheme and Section IV concludes the paper.

II. Hiding in Huffman Trees

Hiding information in Huffman trees exploits the property of such trees that any other tree that assigns the same codeword lengths to the symbols will also be optimal. Thus, there is redundancy in the structure of a Huffman tree which can be exploited for information hiding. When building the Huffman tree, the standard algorithm recursively removes and merges the least probable symbols into supersymbols, the probability of which is the sum of the probabilities of the symbols in its composition, and then adds the new supersymbol to the list. This is repeated until just one supersymbol remains, which contains all the original symbols and has

978-0-7695-4712-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IIH-MSP.2012.23

probability 1. Since the choice of how to order the children of a node is arbitrary, the specific choice made can encode a number of bits. Therefore, after the tree is built, the secret information can be embedded by reordering the children of each internal node of the tree according to Algorithm 1. The extraction algorithm, for recovering the hidden bits from the tree, is shown in Algorithm 2.

Algorithm 1: Huffman Tree Information Hiding

Input: Huffman tree T, a secret message M of sufficient length
Output: Steganographic Huffman tree T
$Q = \emptyset$
Add T to Q
repeat
  $t = Q_1$
  $Q = Q \setminus t$
  if t *is a leaf* **then**
    Continue
  end
  if t *has a NULL child* **then**
    $n = \lfloor log_2(\frac{D!}{B!}) \rfloor$
  end
  else
    $n = \lfloor log_2(D!) \rfloor$
  end
  $m = M_{i=1}^{n}$
  $M = M_{i=n+1}^{|M|}$
  Convert the bitstring m to an integer w
  Sort $t.children$ lexicographically by symbol
  Permute $t.children$ according to w
  for *each* $c \in t.children$ **do**
    Add c to Q
  end
until $Q == \emptyset$;
return T

Algorithm 2: Huffman Tree Information Extraction

Input: A Steganographic Huffman tree T
Output: A secret message M
$D = |T.children|$ is the degree of T
n_S is the number of symbols at leaf nodes of T
$B = (D-2) - ((n_S - 2) mod (D-1))$
$M = \epsilon$, the empty string
$Q = \emptyset$
Add T to Q
repeat
  $t = Q_1$
  $Q = Q \setminus t$
  if t *is a leaf* **then**
    Continue
  end
  if t *has a NULL child* **then**
    $n = \lfloor log_2(\frac{D!}{B!}) \rfloor$
  end
  else
    $n = \lfloor log_2(D!) \rfloor$
  end
  Convert the ordering of $t.children$ to an integer w
  Convert w to a n-bit binary string m
  Append m to M
  for *each* $c \in t.children$ **do**
    Add c to Q
  end
until $Q == \emptyset$;
return M

III. Capacity, Robustness and Security

Three key properties of covert channels are capacity, robustness and security. The capacity of the channel is the number of bits that can be sent per transmission, usually measured in bits per object or bits per symbol. The robustness of the channel is the amount of noise, or tampering, that the channel can endure without preventing covert communication from occuring. Robustness may also refer to the types of noise and tampering that the channel can resist; for example, some image watermarking techniques are robust against resizing and compression [8] and some linguistic watermarking techniques are robust against summarization and rephrasing [9]. The steganographic security of the channel is a measure of the indistinguishability of the steganographic usages of the channel from the non-steganographic usages. Steganographic security differs from cryptographic security in that cryptography is concerned with preventing the *content* of the message from being read by anyone other than the intended recipient, while steganography is concerned with preventing the *existence* of the message from being discovered by anyone other than the intended recipient.

A. Capacity

Theorem III.1. (Hiding Capacity of a D-ary Huffman Tree) *The number of bits that a D-ary Huffman tree can encode in its structure is given by*

$$n = \lfloor \log_2(\frac{D!}{B!}) \rfloor + \lfloor \log_2(D!) \rfloor \frac{n_S - (D-B)}{(D-1)} \qquad (1)$$

where n_S is the number of symbols, $D \geq 2$ is the degree of the tree and $B = (D-2) - ((n_S - 2) mod (D-1))$ so that $(D-B)$ is the number of symbols merged at the deepest level of the tree.

Proof of Theorem III.1 is included in the expanded version of the paper.

Thus, the number of bits that a tree with degree $D = 2$ can embed is given by $n = n_S - 1$ bits, where n_S is the number of symbols in the alphabet.

B. Robustness

The channel is said to be *robust against* that which an adversary, let us call her Wendy, can do that does not disrupt the channel or cause information losses. Everything else, i.e. that which Wendy can do to disrupt the channel and cause information loss, is called a *countermeasure*.

Anything that Wendy does to the stego-object that does not change the tree structure or codebook will not affect the hidden message. Thus, the covert channel is robust against modifications to the content of the cover-object, such as addition and deletion. As an extreme example, suppose that Wendy intercepts a Huffman coded text document from Alice to Bob and, supposing that the secret is embedded in the text of the document, decodes the document and replaces the text with entirely new text before recoding the document and sending it on to Bob. Since Wendy did not modify the Huffman tree section of the object, the hidden message which resides in the structure of the tree remains intact.

On the other hand, there are a number of countermeasures that Wendy can employ to annihilate the hidden message from the stego-object without degrading the object. For example, she could corrupt the message by randomly permuting the children, or she can change the message by clearing it and embedding her own (perhaps a warning for Alice and Bob to behave). To protect against an active adversary, Alice and Bob should uses digital signatures.

C. Security

In abidance with Kerckhoffs' law [10], it assumed that Wendy knows that Alice and Bob are using Huffman trees in their covert communication. To detect the usage of the channel, Wendy must devise a steganalysis system which can determine if a given Huffman tree is hiding information. The most straightforward method of accomplishing this is to hypothesize that clean Huffman trees have a certain property that trees modified to hide information do not have and to decide whether or not a given tree contains hidden information by testing whether or not the property holds. Three such properties might be (1) that the children of every node are in lexicographical order, (2) that the children of every node are in order by likelihood of symbol and (3) that the tree is in the canonical form. With these three properties, Wendy can construct a decision rule which classifies an object as clean if any of the three properties hold and as a stego-object otherwise. Wendy can use the content of the message to estimate the distribution of the symbol alphabet to use in testing the ordering of nodes by probability.

This decision rule was tested against two sets, each consisting of 1000 stego-objects and 1000 clean objects using binary trees. The accuracy of the classification was recorded for different alphabet sizes and lengths of content. The first set of stego-objects was generated by picking a random distribution for the symbol alphabet, building the binary Huffman tree given that probability distribution, embedding a random secret message into the tree and then generating content according to the distribution of the alphabet. The results of the security test using the first test set are shown in Figure 1. The second set of stego-objects was generated by building the Huffman tree given the empirical distribution of the content, and then embedding the secret message. The results of the security test using the second test set are shown in Figure 2.

Figure 1 shows that Wendy's detection accuracy goes to 50%, equivalent to guessing, as the size of the alphabet increases. The peak in accuracy centered around alphabets of size 6 is due to the fact that clean trees are much more likely than dirty trees to be in probabalistic order, which increases Wendy's accuracy in distinguishing between the two as the content length increases so that Wendy has a better estimate of the true symbol distribution. As the number of symbols increases, Wendy's ability to distinguish between clean objects and stego-objects diminishes because clean trees turn out to be no more likely than stego-trees to be in any particular order.

In Figure 2 it is shown that Wendy's detection accuracy is significantly improved in the case where the Huffman tree is generated based on the empirical distribution of the content. In this case, Wendy's estimate of the symbol distribution is identical to the symbol distribution used to generate the Huffman tree. The increased accuracy of her estimate increases the power of the probabilitic-ordering test to distinguish between clean trees and stego-trees. When only one symbol of content is transmitted, Wendy's accuracy is equivalent to guesswork because she does not have a good estimate of the symbol distribution. The slow drop-off of accuracy as the size of the symbol alphabet increases is due to stego-trees which happen to be in order, causing false-negative errors. Wendy's detection accuracy increases as the length of the content increases because the increased content length gives a more fine-grained symbol distribution, which in turn reduces the likelihood of a stego-tree being mistaken for a clean tree.

IV. Conclusions

This paper has presented a novel information hiding scheme which uses the structure of Huffman trees to encode secret data. The capacity, robustness, and security of the resultant covert channel were analyzed and discussed. It was shown that the capacity of the channel is related to the degree of the tree and the size of the symbol alphabet through Equation 1.

The robustness analysis argued that the covert channel is robust against any and all modifications to the content of the cover-object. However, an active adversary with knowledge of the stego-system can employ countermeasures that reduce or eliminate the capacity of the channel, such as recoding the object using a different but equivalent tree. Since robustness

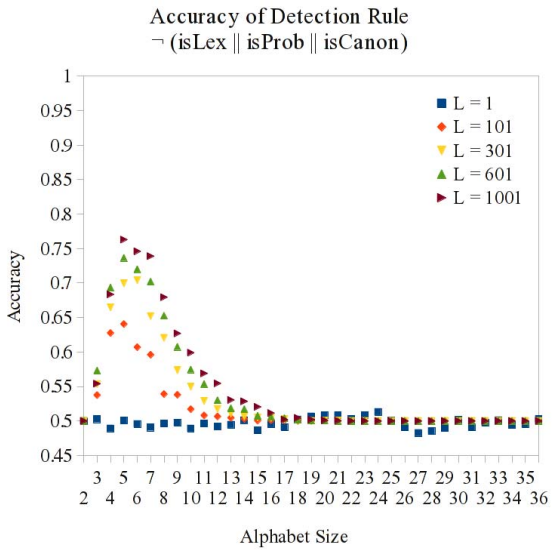

Figure 1. The accuracy of the detection rule that a tree which is not in some order is marked as stego. This data is from a simulation which assumes that the distribution that generated the Huffman tree is the same as that from which the content is constructed.

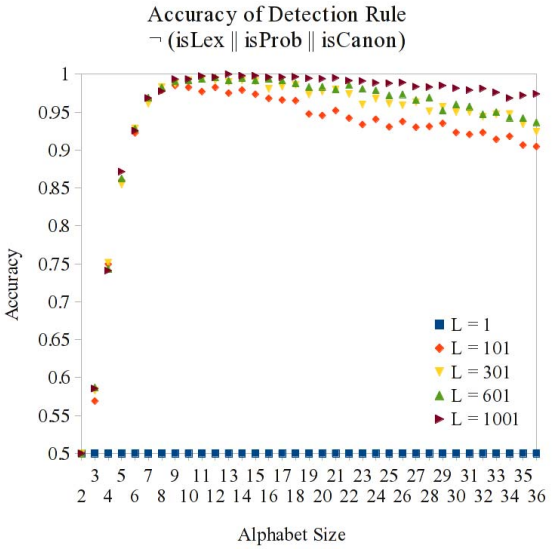

Figure 2. The accuracy of the detection rule that a tree which is not in some order is marked as stego. This data is from a simulation which assumes that the Huffman tree is generated using the empirical symbol distribution of the content.

is only a concern against an active adversary, the usage of a digital signature scheme would suffice to prevent Wendy from tampering with the objects in the channel.

Finally, the security analysis showed that a passive adversary with knowledge of the stego-system can acheive high accuracy of detection in the case where the Huffman tree is generated using the empirical symbol distribution of the content. On the other hand, if the tree is built using an *a priori* distribution that Wendy must estimate using the content, her accuracy is no better than pure guesswork. The results obtained suggest that, if the two are willing to put up with slightly sub-optimal compression, then Alice and Bob can trade a lower compression ratio for dramatically increased security, e.g. they can reduce Wendy's accuracy from greater than 90% as shown in Figure 2 to 50%, equivalent to pure guesswork, as shown in Figure 1.

REFERENCES

[1] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding: A survey," *Proceedings of the IEEE (special issue)*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] J. Reeds, "Solved: The ciphers in book (iii) of (t)rithemius's (s)teganographia," *Cryptologia*, 1998.

[3] S. Singh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor, 2000.

[4] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Institute of Radio Engineers*, vol. 40, pp. 1098–1101, 1952.

[5] D. Salomon, *Data Compression: The Complete Reference*, 4th ed. Springer, 2007.

[6] K.-N. Chen, C.-F. Lee, C.-C. Chang, and H.-C. Lin, "Embedding secret messages using modified Huffman coding," in *Intelligent Information Hiding and Multimedia Signal Processing*, 2009, pp. 278–281.

[7] K.-N. Chen, C.-F. Lee, and C.-C. Chang, "Embedding secret messages based on chaotic map and Huffman coding," in *International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 336–341.

[8] F. Sebe, J. Domingo-ferrer, and J. Herrera, "Spatial-domain image watermarking robust against compression, filtering, cropping, and scaling," in *In Information Security Workshop*. Springer-Verlag, 2000, pp. 44–53.

[9] M. J. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, "Natural language watermarking and tamperproofing," in *In Lecture Notes in Computer Science, Proc. 5th International Information Hiding Workshop 2002*. Springer Verlag, 2002, pp. 7–9.

[10] A. Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires*, vol. 9, pp. 5–83, 161–191, January , February 1883.