Practical Internet Steganography: Data Hiding in IP

Deepa Kundur
Texas A&M University, College Station
Department of Electrical Engineering
3128 TAMU, College Station Texas, USA 77843-3128
deepa@ee.tamu.edu

Kamran Ahsan
Government of Ontario
123 Edward Street
Toronto, Ontario Canada
kamran.ahsan@cbs.gov.on.ca

Abstract

This paper investigates practical techniques and uses of Internet steganography*. Internet steganography is the exploitation of Internet elements and protocols for the purpose of covertly communicating supplementary data. Each scenario facilitates the interaction of fundamental steganographic principles with the existing network security environment to more generally bridge the areas of data hiding, network protocols and security.*

1. Introduction

Steganography is the process of discreetly hiding data in a given *host carrier* for the purpose of enhancing value or subliminally communicating information. Steganography is possible through the existence of covert channels. The concept of a covert channel was first introduced by Lampson [1] as a channel that is used for information transmission, but that is not designed nor intended for communication. The host carrier may be any message or information source that contains redundancy or irrelevancy. For instance, digital images are often used to embed covert information; the limited capability of human vision allows data hiding or watermarking of this type of media.

The host carrier can also be an element or mechanism of an overt communication channel such as a network packet or protocol. An overt channel can be exploited to create the presence of a covert channel by judiciously selecting components of the overt channel that contain or exhibit redundancy or irrelevancy. These components by virtue of their ambiguous nature can be modified for the purpose of *embedding* and later *detecting* supplementary covert information without affecting overt information flow.

Many recent research efforts have been focused on steganography of media host carriers such as digital images. In this position paper, we assert that a more important problem that has received little attention to date is that of Internet steganography. The sheer volume of Internet traffic provides a higher bandwidth vehicle for covert communications which leads to a plethora of applications impractical for standard image steganography. Furthermore, it can be shown that through “ideal compression” steganography in images can be eliminated [2]. However, given the design mantra of the Internet involving communication, connectedness and collaboration which has led to an “open” intentionally redundant specification, it is highly unlikely that Internet steganography can be subverted.

The objectives of this paper are to:

1. demonstrate two existing covert channels at the network layer of the Internet,
2. relate the available bandwidth for covert communication, traditionally associated with a violation of system security policy, to potentially performance enhancing network applications.

In the next section we provide a practical framework for Internet steganography. Sections 2 and 3 provide simple examples of covert channels in the Internet Protocol (IP) followed by application scenarios that can make effective use of the unutilized bandwidth.

1.1 Framework

In this work we consider the situation in which the covert channel piggybacks on an already established overt channel such as the Internet. The embedding and detection processes for covert communications do not interfere with normal overt information flow. We assume that our communicating parties denoted as *Alice* and *Bob*, transfer information overtly over a computer network, and employ data hiding involving the TCP/IP protocol suite to communicate supplementary information covertly. Data hiding is employed through a *stego-algorithm* that takes as input the covert message C_k, a sequence of network packets $\{P_k\}$ known as the cover-network packet sequence, and possibly a secret key to generate a stego-network packet sequence $\{S_k\}$ that contains the overt payload of $\{P_k\}$ while piggybacking C_k. The stego-network packet sequence $\{S_k\}$ is sent to Bob over a computer network. The covert message C_k traverses a generally non-ideal channel characterized by the network behavior on the packet S_k. Given a model of the channel, Bob estimates the covert message to produce C_k^*.

The basic framework is shown in Figure 1. We summarize the relevant points below:

- For reasons of security, Alice and Bob may employ the use of a symmetric secret key, so that only Bob can detect the information accurately. In addition, for the packet-sorting approach, Alice and Bob are both assumed to implement IP Sec.
- The stego-network packet, S_k, may pass through one or more intermediate network nodes in order to reach Bob. The covert channel, by definition, must be non-detectable by these nodes, which can be ensured if and only if the intermediate node finds no critical difference between $\{P_k\}$ and $\{S_k\}$ while processing the sequence. The reader should note that we define “non-detectability” with respect to standard network node capability and not with respect to a human observer.
- At an intermediate node, a stego-network packet S_k may be dropped due to the non-availability of buffer capacity. However, this possibility is assumed to be non-existent in our analysis of the proposed algorithms. We are focusing on that network traffic that is most unlikely to be dropped due to buffer unavailability. Such a condition is possible, if we consider QoS mechanisms through which network traffic can be prioritized as a preferred class. In addition, we assume that there is remote possibility that the same stego-network packet may be corrupted during transmission and consequently dropped by the data link layer mechanisms.

1.2 Previous Work and Perspective

Our research can be considered in part, a logical extension to [3] and [4] in which Handel and Sandford and Wolf propose the reserved or unused fields in packet headers for data hiding; we adopt more specific approaches. The proposed data hiding scenarios are more practical and robust since they are based on redundancies and multiple interpretations of process strategies of the Internet protocol. This makes the scenarios non-detectable to various automated security mechanisms and intermediate nodes.

This research also expands on the work presented in [5], since concrete steganographic encoding and decoding techniques are suggested for interoperability with network security mechanisms such as firewalls and routers. The header fields we select for data hiding are flexible for sending covert data without interfering with standard network processes and for being accessible to routers and firewalls. In contrast, [5] suggests the use of the TCP sequence number and acknowledgment number fields, which reflect the number of bytes sent and acknowledged. These fields, we believe, are therefore inflexible for covert data transfer.

Overall, we attempt to provide improved data-hiding approaches on the basis of several interesting papers in the area. Moreover, we take into account, in part, the effect of the network on the stego-network packet S_k and provide novel application scenarios utilizing concepts proposed by [6]. Together, we hope to present a more comprehensive picture of the data hiding in computer networks.

The reader should note that our techniques for Internet steganography are more specific and ad hoc than those commonly found for images. The primary reason for this is because our host carrier is fundamentally more *structured* than bitmap media [7]. The notions of redundancy and irrelevancy exploited for covert communications are tied to the available model of the host carrier. This leads to more general formulations of the problem for digital image steganography in which there exists well-known mathematical representations of the host and human perception. In contrast, for carriers such as network packets and protocols in which ambiguity in representation is more specific and must be identified through brute-force study, the techniques are ad hoc.

2 Packet Header Manipulation

In designing our data-hiding approaches, we consider functional interfaces required in all IP implementations; a summary of these interfaces is found in [8]. In this way, we avoid identifying covert channels specific to particular IP software implementations, which are not available for general use.

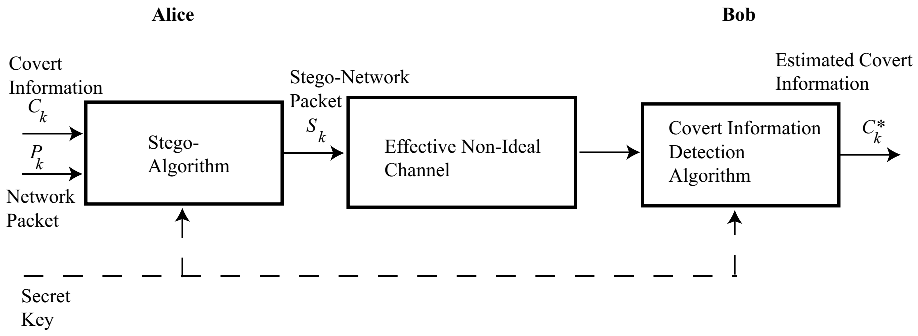
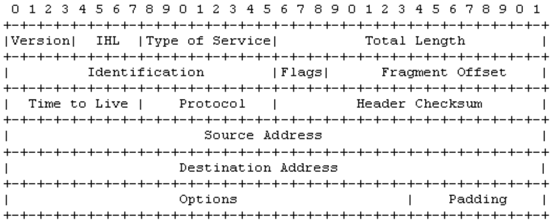


Figure 1. The general covert channel framework in TCP/IP



```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. The IP header

A close study of [8] reveals that redundancy exists in the Internet Protocol's fragmentation strategy. Figure 2 displays the IPv4 header. The Flags field contains fragmentation information. The first bit is *reserved*; the second is denoted DF (to represent Do not Fragment), and the third is denoted MF (to represent More Fragment). An un-fragmented datagram has all zero fragmentation information (i.e., MF = 0 and 13-bit Fragment Offset = 0), which gives rise to a redundancy condition; i.e., DF (Do not Fragment) can carry either "0" or "1", subject to the knowledge of the maximum size of the datagram. This aspect is exploited in our scenario.

Consider two workstations on the same network with users Alice and Bob, who have decided to have a covert communication employing the protocol suite of the network. They are aware that the network administrator is very security conscious and that the TCP/IP software is configured properly as per the security policy of the organization. Alice and Bob have knowledge of the MTU (maximum transmission unit) of their network and are aware of the fragmentation strategy, which follows the standard design considerations of IP [8]; any datagram of length below or equal to the MTU is not fragmented.

The embedding algorithm avails the fragmentation strategy of the Internet Protocol, and the DF bit is used to send a covert bit to Bob. Bob accordingly reads the DF bit and gets the covert message from Alice, who is sitting on the same network.

On the basis of the above explanation, it can be shown the datagrams in Tables 1 and 2 bear the same meaning to the overt network provided that Alice and Bob have the MTU information beforehand and know, for the purposes of this example, that the MTU is greater or equal to 472. In such a case, neither datagram will be fragmented although they differ in the DF bit. Thus, this redundancy leads to the possibility of covert information through judicious selection of each representation. Datagrams 1 and 2 sent by Alice can therefore communicate "1" and "0" respectively to Bob. The constraint, however, is that both parties require prior knowledge of the MTU.

ID field	Flags	Frag.Offset	Total Length
XXX..XX	010	000...0	472

Table 1. Datagram 1 to communicate "1".

ID field	Flags	Frag.Offset	Total Length
XXX..XX	000	000...0	472

Table 2. Datagram 2 to communicate "0".

To demonstrate how both datagrams are similar from the perspective of a network, we note that Datagram 1 is of moderate length, but fragmentation is not allowed, since the DF bit is set. Datagram 2 of the same length has the fragmentation bit unset, yet fragmentation is not possible since it is below the value of the MTU. Since Alice and Bob know the MTU of their network and have agreed to send a datagram of a size smaller than the MTU, there will be no fragmentation.

2.1 Potential Applications

Associating supplementary information sent via covert mechanisms employing packet header manipulation algorithms find the following application scenarios:

1. Enhanced filtering criteria in packet-filtering routers (firewalls). If the additional information pertains to a user or an application, a more reinforced filtering policy can be defined. Therefore, in addition to performing filtering on the basis of source and destination addresses, source and destination ports and protocol, the router can have enhanced criteria for filtering packets based on user and application information.

2. A client server architecture wherein several clients can make a request to the FTP server of, say, a library. A log file can be maintained, for audit purposes, based on the request sent by various users (based on their user information tied to their "request packets"). Moreover, serving the request, like transferring a digital image to the user, can have the same user information or library information tied to the content packets. This scenario of *security tied to the content* avoids the violation of digital copyrights by the specific user.

3. A logging process for the above application scenario based on user or application specific information will complete the picture (i.e., logging of valid user), maintain the record of user requests based on user information and, ultimately serve the user requests by having either the user information or the server/source (library) information tied to the content packets to avoid the copyright violation.

4. Value is added to content delivery networks. A content delivery network is an overlay network to the public Internet or private networks, built specifically for the high performance delivery of content. This concept adds intelligence to networking wherein the network makes path decisions based on more than simple labels such as the IP address. Having specific information tied to the content itself would ease up the process of content delivery networking. Decisions based on specific information within the content can incorporate smart routing mechanisms within the networks. The same information can also be used for the accounting and billing processes of the same networks.

3 Packet Sorting

In this section, we describe in general terms the use of packet ordering for covert communication. There are $n!$ ways in which to arrange a set of n objects. If the order of these objects is not of concern, then there is an opportunity through judicious selection of an arrangement of the n objects, to communicate covertly a maximum of $log_2\ n!$ bits. For a set of $n = 25$ objects, for instance, 83.7 bits can be communicated. The capacity for data hiding increases dramatically for large n.

Applying this principle to our framework, we consider data hiding through network packet sequence ordering. Modifying the sequence number of packets requires no change of the packet content, so it is expected that the network protocols or processing of packets will not be significantly different. Based on these advantages, the feasibility of data hiding using packet sorting within the IP layer is explored.

The packet sorting and resorting processes require a reference in order to relate sorted packet numbers to their actual natural order to decode the covert information. Such a reference is available within the IP Sec environment, which includes two protocol headers, ESP and AH that provide a 32-bit *sequence number field*. The primary aim of this field is to detect *replay attacks*; hence it is directly related to packet sequencing. The *anti-replay mechanism* is intended to determine whether a received packet is a duplicate or not. This replay protection therefore also provides a means to associate an order to packets which can be exploited for covert communication.

Unlike the packet header manipulation approach, in this scenario the cover object is a *sequence* of network packets $\{P_k\}$ rather than a single packet P_k to embed one covert message symbol. The stego algorithm "sorts" the original cover-network packet sequence, resulting in the generation of "sorted" packets with the sorted sequence numbers in their headers. The reader should note that by sorting we are only referring to modification of the sequence number fields of a packet stream. We are not physically swapping location values of the packets at the transmitter. The covert data detection process identifies the covert information sent by estimating it from the sorted packet stream.

The reader should note that the stego-network packet sequence, $\{S_k\}$, may pass through one or more intermediate network nodes in order to reach

Bob. Under ideal network conditions in which proper physical sequencing is guaranteed by the network, the estimation of the covert information is a trivial task of using a look up table to map the sorted sequence to a corresponding covert bit sequence. However, since in general the network cannot guarantee proper sequencing for packet delivery, the transmission process must be modeled as a non-ideal channel characterized by *position error(s)* imposed by the practical network behavior. The covert data estimation process at the receiver must attempt to recover the information in order to minimize bit error. By virtue of our practical formulation, the process is somewhat ad hoc. We do not detail such a method in this paper, but the reader is referred to [9] for one possible non-ideal channel model for the network and a corresponding estimation process.

3.1 Usage Scenarios - Packet Sorting

We believe that the packet sorting technique in the IP Sec environment finds following applications:

1. Preliminary (added) authentication in IP Sec environment.
2. A mechanism to facilitate enhanced anti-traffic analysis by having packets with sorted sequence numbers.
3. Enhanced security mechanisms for IP Sec protocols, especially for ESP operating in tunnel mode, which enable enriched security for virtual private networks (VPNs).

4 Final Remarks

In this paper, we have presented two simple covert channels within the Internet Protocol. They both exploit redundancies in the representation of information. Our first approach based on packet header manipulation makes use of the redundancy in the representation of the fragmentation strategy. Our second approach based on packet sorting employs the sequence number field in IP Sec and its somewhat redundant relation to the physical ordering of the packets (i.e., in ideal network conditions, there is a one-to-one correspondence between the two; in a general network the sequence number and physical ordering are correlated) in order to transmit covert information. Simple covert channels based on irrelevancy within the IP specification are also possible. For instance, the unused or reserved bits within specific fields can be exploited to carry covert data [3].

Our demonstration of the existence of the covert channels through these algorithms points to the possibility of associating additional information in the network packets. These steganographic scenarios provide flexibility to explore this possibility both in the IP Sec and non-IP Sec environments.

References

[1] B. W Lampson, "A note on the confinment problem," in *Proc. of the Communications of the ACM*, October 1973, number 16:10, pp. 613–615.

[2] R. J. Anderson and A. P. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, May 1998.

[3] T. Handel and M.Sandford, "Hiding data in the OSI network model," Cambridge, U.K., May-June 1996, First International Workshop on Information Hiding.

[4] M. Wolf, "Covert channels in LAN protocols," in *Proceedings of the Workshop on Local Area NetworK Security (LANSEC'89)*, T.A.Berson and T.Beth, Eds., 1989, pp. 91 – 102.

[5] C. H. Rowland, "Covert channels in the TCP/IP protocol suite," Tech. Rep. 5, First Monday, Peer Reviewed Journal on the Internet, July 1997.

[6] R. Ackermann, U. Roedig, M. Zink, C. Griwodz, and R. Steinmetz, "Associating network flows with user and application information," in *Eight ACM International Multimedia Conference*, Los Angeles, California, 2000.

[7] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in internet traffic with active wardens," in *Proc. Fifth Int. Workshop on Information Hiding*, F. A. P. Petitcolas, Ed., October 2002, number 2578 in Lecture Notes in Computer Science.

[8] University Southern California Information Sciences Institute, "Internet protocol, DARPA internet program , protocol specification," September 1981, Specification prepared for Defense Advanced Research Projects Agency.

[9] K. Ahsan and D. Kundur, "Practical data hiding in TCP/IP," in *Proc. ACM Workshop on Multimedia Security*, 2002.