

STEGANOGRAPHY IN NETWORK PROTOCOLS

INTRODUCTION

Steganography refers to the process of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. The message is deciphered using a hidden key known only to the sender and the intended receiver. Steganography is possible through the existence of covert channels. Covert channel is a channel that is used for information transmission but one that is not designed nor intended for communication (Lampson, 1973). Most common modes used for steganographic messages are images, text, video and audio modes.

With the advent of the internet, the vast volumes of internet traffic provides a high bandwidth vehicle or a carrier for subliminal communications. With the design mode of internet involving communication- connectedness and collaboration which has led to “open” system environment and internationally redundant specifications, Internet provides fertile grounds for proliferation of steganography (Kundur & Ashan, 2002).

1. PREVIOUS WORK :OVERT CHANNELS IN NETWORK PROTOCOLS

Some of the early work on covert channels in network protocols was carried out by Girling who focused on Local Area Networks (LAN). He identified there are obvious covert channels in the LAN environment and introduced the notion of wiretapper who monitors the activities of a specific transmitter on the LAN. The covert communicators are the transmitter and the wiretapper (Girling, 1987). As an extension to Girling's work, focus on LAN protocols can be noted in Wolf's work where he establishes that the encryption process, which is the basic

security mechanism for LAN protocols, is inadequate in blocking the covert channel activities (Wolf, 1989). Data hiding in OSI model (Open System Interconnection) focus on the system elements, which can be utilized for data hiding activities. This model takes in to account standard network environments and architecture. In this work of Handle and Sanford, the basic principles of hiding data in the layers of the OSI are established (Handle & Sanford, 1996). Other work on steganography in network protocols includes the work of Rowland, which focus specifically on the IP and TCP headers being used for hiding data. This work which establishes the existence of covert channels in the TCP /IP protocol suit (Rowland, 1997) His work which takes a more pragmatic approach to encoding and decoding activities is considered as a breakthrough in the field of steganography research (Ashan, 2002). Internet steganograpy is a concept developed by the Katzenbeisser and Petitcolas who extends the study of potential for data hiding in the TCP/IP protocol suit. The significance of the TCP/IP protocol suit lies in the vast volume of hidden data that can be communicated through the TCP/IP packets, which are used in communicating thousands of Internet packets (Katzenbeisser & Petitcolas, 2000). Existing studies in the field establish the existence of covert channels in the network protocols and explores the devising of techniques of embedding and extraction of data at transmission point and at the receivers end.

2. MEANS OF STEGANOGRAPHY IN NETWORK PROTOCOLS

Two main approaches to hiding data in network packets involves packet header manipulation and through packet sorting. In the case of TCP/IP protocol suit, four conceptual layers containing a stack of protocols can be noted and steganograpy activities can take place in one or many of these layers.

Table 1 – TCP/IP Protocol Suit with stack of general protocols in each layer

Application Layer	FTP, SMTP, DNS, TelNet
Transport Layer	TCP,UDP
Internet Layer	IP, ICMP, IGMP
Data Link Layer	Network Interface and Device Drivers

These methods can be used in a variety of areas such as the following: - Bypassing packet filters, network sniffers, and "dirty word" search engines. -Encapsulating encrypted or non-encrypted information within otherwise normal packets of information for secret transmission through networks that prohibit such activity ("TCP/IP Steganography"). - Concealing locations of transmitted data by "bouncing" forged packets with encapsulated information off innocuous Internet sites (Rawland, 1997)

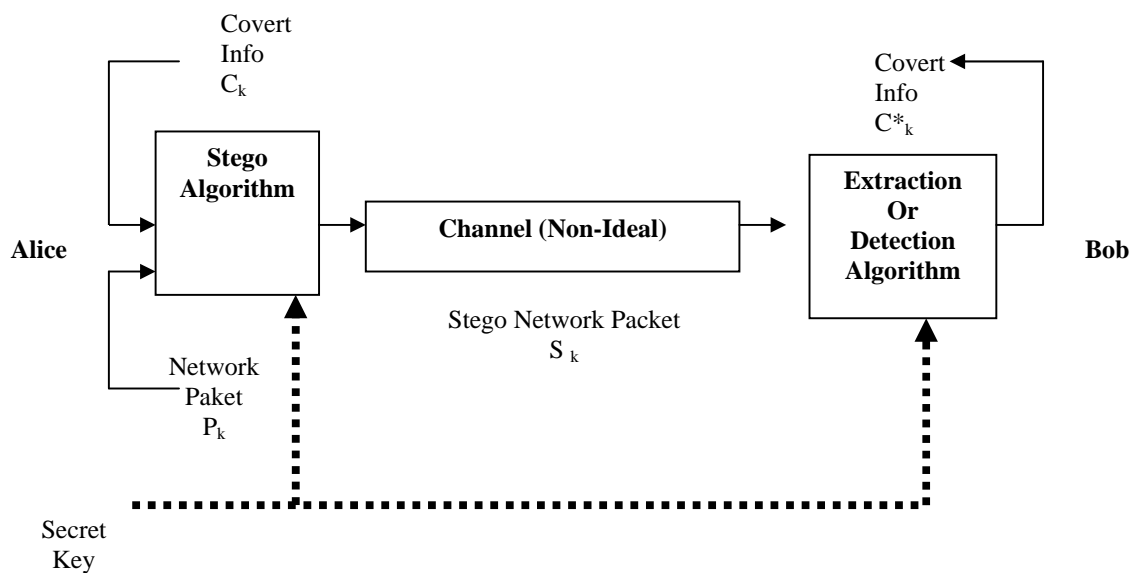


Figure 1– The General Covert Channel Framework in TCP/IP

3. Packet Header Manipulation Approach to Data hiding

The TCP/IP header contains a number of areas where information can be stored and sent to a remote host in a covert manner. The identification field of the IP protocol helps with re-assembly of packet data by remote routers and host systems. Its purpose is to give a unique value to packets so if fragmentation occurs along a route, they can be accurately reassembled (Ashan, 2002). The layered structure of network requires that the IP datagrams to encapsulate information received from the transport layer. For example the IP header encapsulate ICMP messages, the IGMP report and query message. IP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram) the techniques associated with packet header manipulation makes use of the redundancy in the representation of information in the Internet protocol for effective data hiding. The following scenarios provide example of data hiding through packet header manipulation in IPv4 Header.

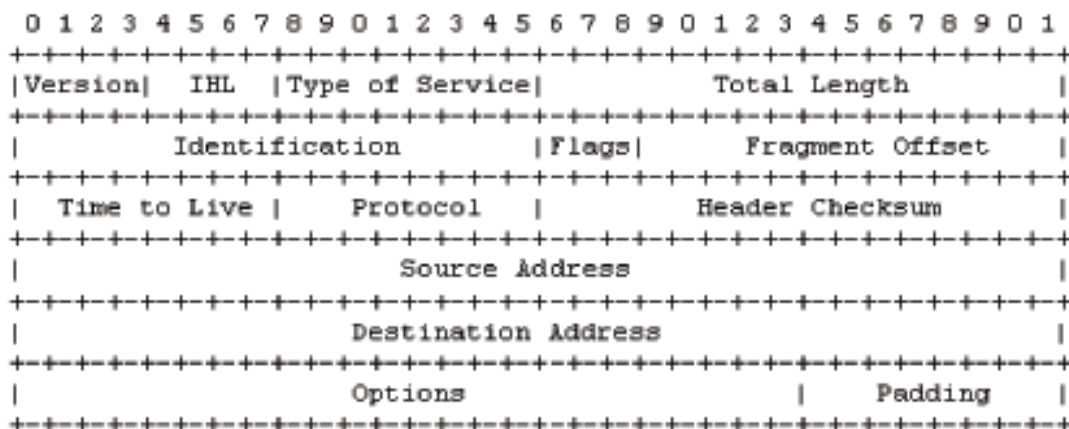


Figure 2 - IP Header

The existence of redundancy in the Internet protocol fragmentation strategy has been established by a close study of the Protocol specifications of “Internet protocol Darpa Internet

program)¹. It can be noted that the design aspects of the Internet Protocols makes identification field of the IP header independent of the fragmentation process (Ashan, 2002). Within each header there are multitude of areas that are not used for normal transmission or are "optional" fields to be set as needed by the sender of the datagrams. An analysis of the areas of a typical IP header that are either unused or optional reveals many possibilities where data can be stored and transmitted. The advantage of hiding data in the more mandatory fields than the optional fields is that it provides robustness resulting from these fields being unlikely or less likely to be altered in transit than say the IP or TCP options fields which are sometimes changed or stripped off by packet filtering mechanisms or through fragment re-assembly (Rowland, 1997).

3.1 Use of IP Header for Data Hiding - Scenario 1

Using the most commonly cited imaginary covert communicators of the network steganography, Alice and Bob who are sharing the same network aim at establishing a covert communication through the exploitation of the network protocol suit's covert channels. They are aware of the MTU of the system and aware that the fragmentation strategy of the network is in line with the standard design considerations of IP. The figure 2 illustrates the IP header and the flag field contains fragmentation information. The first bit is reserved, the second marked DF to indicate DO not Fragment and the third bit is marked MF meaning More Fragmentation. An unfragmented datagram contains all zero fragmentation information which allows for a redundancy. The DF bit can carry either a "0" or "1" subjected to the knowledge of the Maximum Transmission Unit (MTU) The data hiding scenario 1 takes the above redundancy as the means for achieving its covert transmission objectives as described below. In this scenario,

¹ U. S. C. Information Science Institute, "Internet protocol, darpa internet program, protocol specification."September 1981, Specifications prepared for Defense Advance Research Project.

the following two datagrams indicated in Table 1 & 2 appear the same for the overt network. However Alice and Bob who are aware of the network MTU beforehand are allowed the possibility of engaging in subliminal communication through the judicious selection of each representation. Figure 3 provides the process of covert communication taking place in data hiding scenario-1. Table 2 and 3 provides the two datagrams, which the covert communicators may use without being detected by the overt channel.

Table 2 - Datagram 1 – Communicating “1” with the DF bit set

Datagram	16-bit Id. field	3-bit flag field	13-bit frag. offset	16-bit Total len.
1	XX...XX	010	00...00	472

Table 3- Datagram 2 – Communicating “0” with the DF bit unset

Datagram	16-bit Id. field	3-bit flag field	13-bit frag. offset	16-bit Total len.
1	XX...XX	000	00...00	472

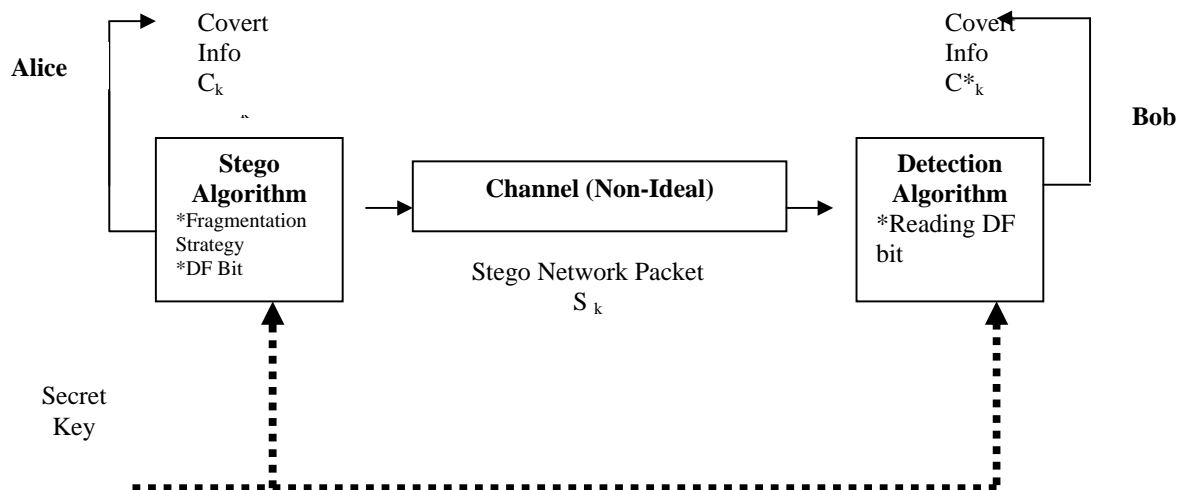


Figure 3 – Diagram of covert communication with fragmentation of DF bit

From the network perspective, these two data grams are similar. Datagram 1 which is of moderate length and does not allow fragmentation since the DF bit is set while the Datagram 2 which is of the same length has the fragmentation bit unset but not fragmented since its set below the value of the MTU. With the covert communicators being aware of the MTU value, this redundancy is being thus exploited to communicate “1” and “0” covertly.

3.2 Use of IP Header for Data Hiding - Scenario 2

While the other scenario of data hiding which was discussed was restricted to situations where the covert communicators were in the same network and possessed prior knowledge of the MTU of the system, this scenario explores the possibility of using the identification field since the datagrams generated by the communication parties should not contain Options in the IP header. This scenario utilizes the identification field and the version and Internet header length fields in combination of the IPv4 header. The advantage of the method is rare chance of being detected by the network monitoring mechanisms as most of them focus on checking the data in each respective field and does not take the possibility of combination fields. The recipient of the covert communication can extract the encoded message only by having prior knowledge of the encoding scheme. The datagram in table 4 indicates a covert message being sent using this scenario of data hiding where the letter “A” is embedded in the identification field. This datagram can be transmitted via the Internet with the potential for getting fragmented but it does not pose an issue as the datagram can be reconstructed using the same identification field. Because of the sheer amount of information one can represent in a 32 bit address space (4,294,967,296 numbers), the sequence number makes an ideal location for storing data. Aside from the obvious example given above, one can use a number of other techniques to store

information in either a byte fashion, or as bits of information represented through careful manipulation of the sequence number (Rawland, 1997).

Table 4 – Datagram encoded with letter “A” in the ID field

4-bit ver. 0100	4-bit IHL 0101	8-bit TOS XXXXXXUU	16-bit Tot Len. XXXXXXXXXXXXXXXXXX	
16-bit Ident. 0000 0100 RRRRRRRR			3-bit flags XXX	13-bit Frag. Off XXXXXXXXXXXXXXXXXX
8-bit TTL XXXXXXXX		8-bit Protocol XXXXXXXX	16-bit Checksum XXXXXXXXXXXXXXXXXX	
32-bit Source Address XX				
32-bit Destination Address XX				

3.3 Data Hiding Scenario 3

This scenario utilizes the same identification field of the IPv4 header but uses the concept of Chaotic mixing or toral automorphism to generate the identifier (Pitas & Voyatzis, 1996). The sorted sequences are generated from the original sequence based on a structure provided by the toral automophosis. The requirement is that the covert parties have means of exchanging the required keys beforehand. These keys include, the main key determining the number of elements in the generated sequence, the sub key, which affects the period of the iterated transformation the third key which is the number of times the toral automorphism is applied. The figure 4 indicates the communication process using the scenario. The robustness of a data hiding scheme would lie in it’s non delectability by he administrator or the network monitoring systems and this scenario of data hiding which uses chaotic mixing techniques has the advantage of structured scrambling to facilitate such robustness.

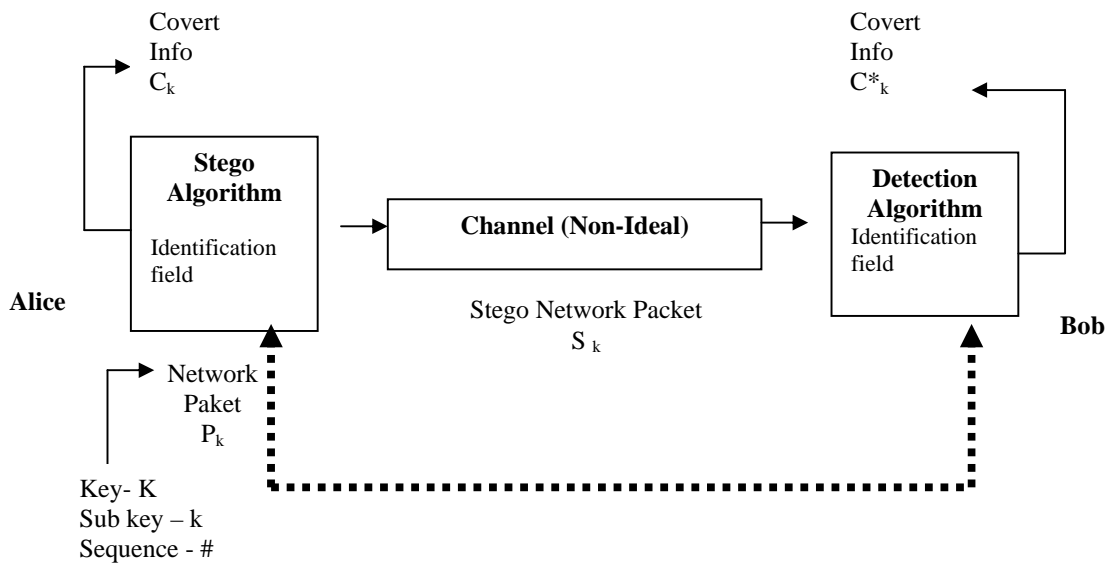


Figure 4 – Diagram of covert communication using scenario 3

3.4 ISN Method- Using the TCP Header

TCP is a transport protocol and the TCP header provides areas for exploitation for covert data transfer. This method exploits the three-way handshake where the sending party will send the payload in the SYN ISN. The receiving party will send data with the acknowledgment. There are many possible redundancy conditions in the TCP header, which is in figure 5

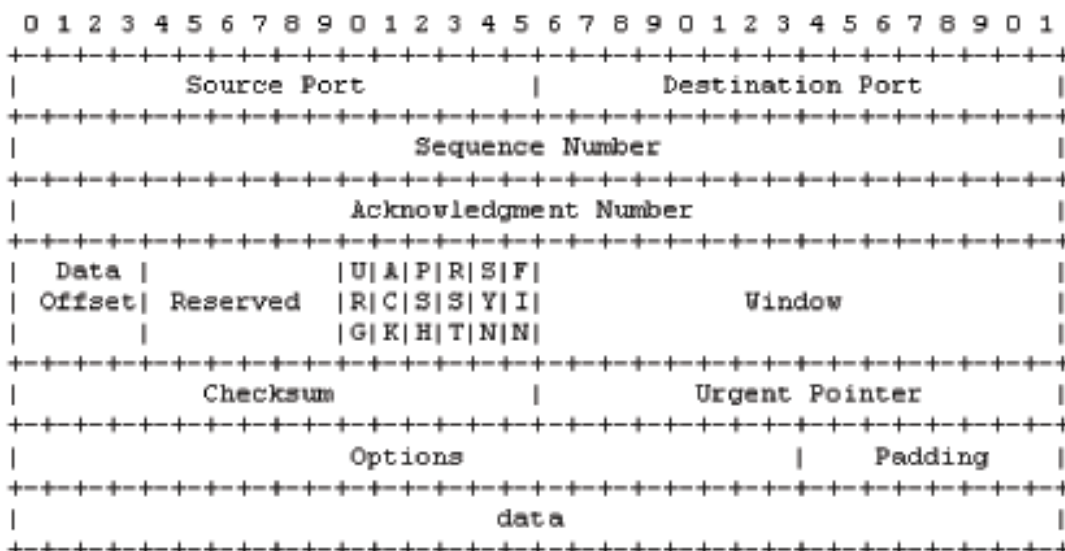


Figure 5 – TCP Header Diagram

The TCP header contains a 6-bit field labeled as code bits, URG, ACK, PSH, RST, SYN and FIN and these provide for 64 possible combinations. Covert channels exploits the possible redundancies within these possible combinations. The advantage of this method is that it is a robust bi-directional covert channel. Tools utilized for the covert communication are the ncovert and covert_TCP

3.5 Use of ICMP Channels for covert communications

ICMP is the mechanism, which is used by routers and host servers to communicate of IP datagram problems back to senders. Exploitation areas in the ICMP includes the Echo reply and Echo response, ICMP Address mask request and Router solicitations. The ICMP Echo reply and Echo request contains an optional data field allowing variable length data to be returned to the sender. In this scenario IP options as the router alert, the time stamp and the route record can be used in encapsulating. This is a popular covert carrier as it is universally available and offers high bandwidth for covert data. Tools used in covert encapsulation includes Loki, ICMP Tunnel, Ish and 007Shell.

3.6 ACK Sequence Number Field Bounce Method

In addition to above scenarios of manipulating the IP header of the packet, another method discusses the possibility of basic spoofing of IP addresses to enable a sending machine to "bounce" a packet of information off a remote site and have that site return the packet to the real destination address (Rawland, 1997). By making the packet to appear as it's coming from a source, which is not the actual source, the covertness of the transmission can be made anonymous. To originate the covert communication through this method, the sender constructs a

packet that contains the forged source ip address, the forged source port, the forged destination ip address, the forged destination port and the TCP SYN number with encoded data. This method can be used in sending messages to servers which has restrictions of access by making the packet appear as being sent from another source. If the bouncer server has a round-robin DNS, then the receiver has to listen for the data coming in from a specific port. The advantage of using this method is that as most firewalls are passive to incoming ACKs, and mainly check on the SYNs , thus allowing the encoded data to pass through. If a network site has a correctly configured router, it usually does not facilitate a forged packet with a network number that is not from it's network to pass through its outbound route. However many routers are not configured with this protection in mind and thus these covert communications exploit this situation.

3.7 Data hiding in Ipv6

Data hiding possibilities in the newest network protocol the IPv6 is being currently explored by experts, inquisitive users as well as prospective covert communicators looking to overcome the security aspects of the new protocol. The Ipv6 header has been streamlined for efficiency, and the new format introduces the concept of an extension header, allowing greater flexibility to support optional features. The IPv6 packet is composed of two main parts: the header and the payload. The header is in the first 40 bytes of the packet and contains both source and destination addresses (128 bits each), as well as the version (4-bit IP version), traffic class (8 bits, Packet Priority), flow label (20 bits), payload length (16 bits), next header (8 bits), and hop limit (8 bits). Next comes the payload, which can be up to 64k in size in standard mode, or larger with a "jumbo payload" option. Figure 6 provides an illustration of the IPv6 header.

Version	Traffic Class	Flow Label	
Payload length		Next Header	Hop Limit
Source Address			
Destination Address			

Figure 6. IPv6 header formats.

3.7.1 Messaging over IPv6 Destination Options

The options field of the IPv6 header can have variable length. The first 16-bits of each extension header are reserved for the Next Header type that follows, and 8-bits for the header length. These fields must be TLV encoded and aligned to a multiple of 8 octets. The first two high order bits of the options field specify what action must be taken if the option type is not recognized. 00 denotes - Skip this option and continue processing the header while 01 instructs to - Discard the packet. These aspects of the header can be exploited and a covert channel can be implemented using the destination options extension.

Example

The example below illustrates how the IPv6 header's destination option can be exploited to communicate covert messages for the covert communicators, Alice and Bob. It is assumed that they are connected to the same network. The multicast covert message from Alice is "hi bob, how are you doing?" and then Bob's response follows as "hi alice". The high bandwidth of the IPv6 header, such long messaging is possible with this mode. A closer look at the payload of the IPv6 packet will show the encoded message.

[RAW] Size: 90 bytes Data:

[RAW] 3a 04 17 23 3c 61 6c 69 63 65 3e 20 68 69 20 62 6f 62 :.# hi bob

[RAW] 2c 20 68 6f 77 20 61 72 65 20 79 6f 75 20 64 6f 69 6e , how are you doin

[RAW] 67 3f 0a 00 81 00 7d 80 00 00 00 00 00 00 00 00 00 00 g?....}.....

[RAW] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

[RAW] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

In the above example, the first octet specifies the Next Header Value, ICMPv6 (0x3a = 58) in this case. Second octet specifies the length of the extension header. Next octet is the destination option type followed by the option length. Octets 5 to 39 contain the payload including the message. 40th octet is padding for alignment reasons. Octets 41 to 90 contain the ICMPv6 Echo-Reply part (Graf, 2005).

3.8 Use of HTTP and other Protocols for covert communication

In addition to the methods discussed above, use of HTTP protocols, DNS method and the Use of IGMP protocol also is present. Wide use of HTTP protocol can be attributed to it being the most widely deployed protocol and the tolerance of its use in almost all organizations. HTTP's design errors also facilitate its use as a wide spread covert communication carrier. In contrast to the HTTP, other lower layer protocols have limited data carrying capacities and low bandwidth, as well as the possibility for alterations of protocol credentials in transit. Multiple techniques can be used in HTTP as the multiple proxies, encryption, encoding, multiple HTTP headers etc that makes the method virtually impossible to detect (Singh, 2005)

4. DATA HIDING BY PACKET SORTING

The use of packet ordering process too is utilized as a mode of covert communication and uses the TCP/IP protocol suit for hiding data. The basis of this method is that there are $n!$ ways of arranging a set of n objects and if there is no restriction for the way in its arranged, then there is a potential for covert communication through judicious selection of the n objects. The advantage is that as the capacity for data hiding increases dramatically with large n . As the modification of the order of the packets bears no changes to the packet content such as the payload or the headers, no major modifications in the protocol definition, or the design. Transmitting covert data through data sorting involves a packet sorting and resorting process and requires a reference in order to relate packet numbers to their actual order. The natural packet ordering sequence of the cover network is used in undoing (resort) the stego-network packet sequence ordering (sorting) to extract both the covert and overt information contained within the packet. The disadvantage of this method lies in that the stego-network packet sequence $\{S_k\}$ will have to pass through one or more intermediate networks in reaching the final destination and may face delays of certain packet deliveries which will affect the stego-network packet sequence. Due to this possibility of delays, the transmission process is modeled as a non-ideal channel characterized by position errors inherent in practical network behavior. However it has been established that very few connections suffer from out of order deliveries (Mogul,1992). Figure .. illustrates the embedding and detection process of the sorting and resorting algorithm.

5. PROTECTION, AND DETECTION

Current research trends in TCP/IP protocol suit focus on protection and defense mechanisms as much as on the performance issues. Protection from these techniques include the

use of an application proxy firewall system which is not allowing packets from logically separated networks to pass directly to each other. Use of Active Wardens, which acts as a security agent and make decisions on how to treat suspicious covert channels and flush them out of the systems with minimal modifications to the overt channels operation is now being implemented. Detection of steganography techniques is still at a stage of infancy and most of available work

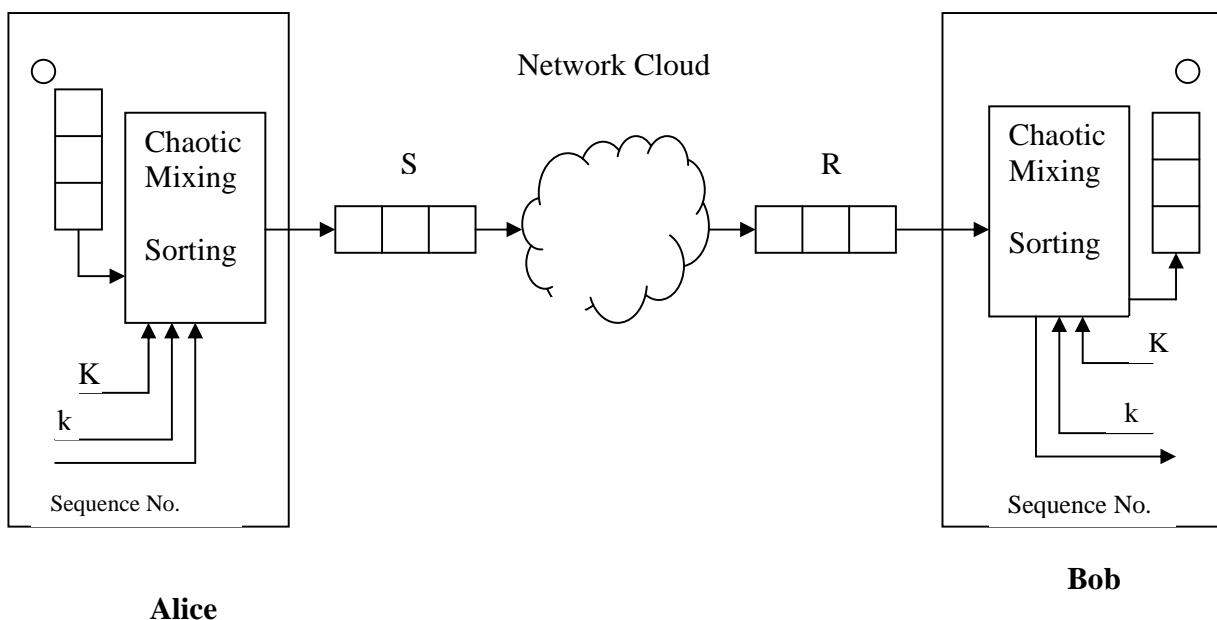


Figure 7 – Sorting & Unsorting process in data ordering

has been done in academic and government domain. No unified method of measuring the lethality (capacity and capability) of a covert channel exists. Some of the detection techniques involve the isolation method and the ACK filters (Singh, 2005). IPsec (Internet Protocol Security) is a set of mechanisms aiming to protect the traffic transmissions at IP level. Connectionless integrity, data origin authentication, protection against replies and confidentiality

are the key security services offered by the IPSec protocol. The IPSec implementation is optional for IPv4 but mandatory for IPv6 implementation and thus can be seen as the current as well as the future security architecture for IP traffic (Ashan, 2002).

Two security mechanisms are used in the IPSec to meet its security objectives and these involve the AH (Authentication Header) and the ESP (Encapsulating Security Payload). While the AH is used in ensuring the authenticity and integrity of the IP datagram, ESP is mainly used in ensuring confidentiality. These two mechanisms incorporate a new AH header and an ESP header to the respective datagrams. These protocols can operate in two modes – the tunnel mode and the transport mode. In the case of transport mode, the AH and the ESP provides protection through interception of packets flowing from the transport layer to the network layer. The tunnel mode is utilized when the final destination of the packet is different from the security termination point.

References

1. Lampson, W. A. (1973). A note on the Confinement Problem. Proceedings Communications of the ACM. No. 16.10. pp.613-615.
2. Ashan, K. (2002). Covert Channel Analysis and Data Hiding in TCP/IP. M.A.Sc. thesis, Dept of Electrical and Computer Engineering, University of Toronto.
3. Wolf, M. (1989). Covert Channels in LAN Protocols. Workshop on Local Area Network Security (LANSEC'89).
4. Rowland, C. K. (1997) Covert Channels in the TCP/IP protocol suit. Tech Rep. 5, First Monday, Peer Reviewed Journal on the Internet.
5. Handel, T. & Sanford, M. (1996). Hiding Data in the OSI network model. Information Hiding. Vol.1174 of Lecture Notes in Computer Science.
6. Girling, C.G. (1987). Covert Channels in LANs. *IEEE Transactions on Software Engineering*, vol. SE-13 of 2.
7. Singh, P. (2005). Whisper on the wire – Network based covert channels Exploitation and Detection. HITBS Conference, Bahrain. Retrieved on August 24, 2005, from http://www.hitbsecconf.com/Presentations/covert_channels_pukhraj_HITB.ppt.pdf#search='covert%20channels%20in%20IPv6'

8. Pitas, I & Voyatzis, G. (1996). Chaotic Mixing in digital images and applications to watermarking. European Conference on Multimedia Applications Services and Techniques. Vol. 2.pp.687-695.
9. Mogul, D.J. (1992) Observing TCP dynamics in real networks. IEEE/ACM Transactions on Networking. Vol. 7, pp. 305-317.
10. Ashan, K. & Kundur, D. (2002). Practical Data Hiding in TCP/IP. ACM Workshops on Multimedia and Security. Retrieved August 24, 2005, from <http://www.tamu.edu/deepa/pdf/acm02.pdf>
11. Katzenbeisser, S. & Petitcolas, S. (2000). Information Hiding Techniques for Steganography and Digital Watermarking. Computer Security Series. Massachusetts: Artech House Inc.
12. Graph, T.(2003). Messaging over IPv6. Retrieved on August 26, 2005, from <http://net.suug.ch/articles/2003/07/06/ip6msg.html>