AN INNOVATIVE APPROACH TO ANALYZE AND DETECT A BROAD CLASS OF TIMING-BASED COVERT COMMUNICATIONS

by

Pradhumna L. Shrestha

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Engineering

Under the Supervision of Professor Hamid Sharif

Lincoln, Nebraska

November, 2014

AN INNOVATIVE APPROACH TO ANALYZE AND DETECT A BROAD CLASS OF TIMING-BASED COVERT COMMUNICATIONS

Pradhumna L. Shrestha, Ph.D.

University of Nebraska, 2014

Adviser: Hamid Sharif

Transmitting information by hiding it in order to evade detection has been practiced since ancient times. In the modern age of computing, digital objects and resources—such as images, video and text files—are used as carriers of hidden information. Recently, an entirely different method of information hiding that leverages existing network resources as side channels for transmitting secret messages has received considerable attention. Since these network resources were not even designed for the purpose of communication, traditional network security elements such as firewalls cannot detect them. These side channels are called covert channels. Covert channels can be used for leaking information and exchanging messages between maligned parties without being detected. This makes covert channels a serious security concern and hence it is imperative to prevent, detect and disrupt them.

Due to the sheer number of covert channel algorithms, it is impossible to deal with them on a case-by-case basis. In this research, an analytical framework that can broadly define all covert timing channels through a mathematical model has been proposed and investigated. From this model, equations have been derived to characterize covert communications in terms of bit error rate under different channel conditions for four popular and diverse covert timing channels. The model was verified by implementing the same algorithms in MATLAB and on a test-bed of real network traffic.

A machine learning-based generic detection mechanism has also been proposed. Statistical fingerprints were derived from the traffic under investigation, which served as feature points for training a Support Vector Machine-based framework. Four types of fingerprints—Kolmorov-Smirnov test score, Regularity score, Entropy and Corrected Conditional Entropy—were used for this purpose. The presented model was then tested against an independent set of feature points derived from an arbitrary traffic under investigation. Results show that the mechanism is very efficient in blind and generalized detection of covert channels.

The presented approach and results have been published in national and international conferences and journals.

Dedicated to my parents

ACKNOWLEDGMENT

First of all, I would like to express my sincere gratitude to my supervisor Dr. Hamid Sharif for his guidance, suggestions and encouragement in this research endeavor and beyond. I owe all of my academic accomplishments to his strong support and excellent supervision during my graduate studies. Dr. Sharif has not only provided great mentoring support to my research and scholarly activities, but also has gone above and beyond to help me in all aspects of my professional life. I would be forever indebted to his supervision, advice and kindness.

I would like to give special thanks to Dr. Michael Hempel for his tremendous support and advice in completing this research and all of my other research activities. Dr. Hempel has invested a great amount of time and effort to help me with my scholarly activities. He has always been approachable and available whenever I needed guidance and help with my research. His modesty and incredible dedication to research and the laboratory has really inspired me.

I would also like to thank Dr. Dongming Peng for his guidance and advice on my research activities and serving on my supervisory committee. His valuable support has been instrumental in helping me solve several research problems in my graduate studies. Also, I would like to thank Dr. Yaoqing (Lamar) Yang and Dr. Jong-Hoon (Jon) Youn for serving on my committee and providing valuable advice on writing this dissertation.

Finally, I would like to thank my parents for their unconditional love and support that has helped me to stay focused on my research. I would like to thank my colleagues in my research group—Ting Zhou, Puttipong Mahasukhon, Tao Ma, Fahimeh Rezaei and Sushanta Mohan Rakshit—for their great camaraderie. Last but not least, I would like to thank everyone at the University of Nebraska and Peter Kiewit Institute for their help throughout my graduate studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 History of Information Hiding

The art of hiding information in innocuous mediums has been practiced for ages and dates back to ancient Greece, when the information was embedded in physical carriers. For instance, there are legends of passing secret messages written on wooden tablets by covering it with wax, tattooing the secret information such as maps on the head of the carrier so that hair would conceal it, using the so-called invisible inks and secret masks for location of messages on texts and letters, etc. The possibility of transmitting messages using musical scores and modifying angles and line lengths of geometrical objects was discussed during the 14^{th}-16^{th} centuries. The World Wars, particularly the second one, played a crucial role in modernizing this field. The Germans developed the microdot technology whereby the physical size of the message, such as a standard page, could be reduced to the size of a printed period and later developed to reproduce the original message. With the advent of digital communications and the spread of computer networks after World War II, the carriers of stenographic messages have shifted from physical carriers to digital carriers.

The possibility of hiding information in digital multimedia content such as images, videos, text and audio files—along with the spread of computer technology and its decreasing cost—brought the science of information hiding from government and military laboratories to the homes of common people. With the arrival of instant global connection brought by the Internet, embedding and transmitting secret messages became possible between any two parts of the world. From the point of view of intended

application, the science of embedding information in digital media can be divided into two domains—watermarking and steganography.

1.2 Watermarking

The concept of watermarking digital media was introduced largely for content and ownership verification and protection. A legitimate entity embeds an image (referred to as a watermark) into the media being produced. The legitimacy of the item under evaluation depends on the presence of the watermark on it. Financial documents like paper currency and checks, postal stamps, important texts and documents bear such watermarks. With the advent of digital photography and image editing tools, watermarks started being embedded on images for copyright protection. Initially, the legitimate owners used to simply put a small image or a signature on top of the image to establish their ownership over the content. However, this created a visible impact on the image, often reducing its aesthetic quality and monetary value. To improve upon this, researchers and scientists designed the so-called digital watermarking, in which the content ownership information embedded in the image is imperceptible to human vision, and hence the quality of the image is not affected. Furthermore, the watermark information no longer had to be in a pictorial form, as any information source embedded was eventually converted to a binary data stream. The concept of watermarking has since been extended to the domain of audio and video, as well.

1.3 Steganography

Historically and, to a large extent technically as well, the art of information embedding was solely called steganography. The crucial difference between watermarking and steganography is the purpose of information hiding. Steganography,

still following its historical roots, is used for secret message transfer. The message, as well as the fact that any form of communication is taking place, is not known to the external parties. Due to this variation in application, watermarking strives for robustness whereas steganography requires stealth. In modern times, steganography is being used to transmit messages secretly by embedding them into digital content like images, audio and video in the form of binary streams, and distributing them via the Internet. A vast amount of research has been done in the area of steganography.

Due to the possibility of transmitting and sharing messages without being detected, steganography is a grave security threat. Government and military database and trade secrets can be leaked by simply embedding them into innocuous looking digital media. Entities with malign intentions can freely communicate and share sinister plans without being detected by authorities. To counter these issues, the domain of steganalysis has surfaced. Steganalysis researchers devise methods to detect the presence of steganographic information in digital media.

1.4 Network Steganography and Covert Channels

An entirely unique method of hiding messages in existing computer network resources has received considerable attention in the past few decades. Interestingly, these network resources were not even designed for the purpose of communications, and traditional network security components like firewalls cannot detect or handle these events. These side channels are appropriately termed covert channels, and the process of transmitting messages across these channels is called covert communications.

The term “covert channel” was first coined by Lampson in 1973 [1], where he defined it as a channel “not intended to transmit information at all.” In 1985, the

Department of Defense published the famous Orange Book [2]. It defined a covert channel as “any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy.” Covert channels and the process of covert communications can be explained in analogy with the famous prisoner’s problem described by Simpson in 1983. He describes a scenario of two prisoners who plan to devise a plan to escape from prison. However, they can only communicate by letters carried by a prison warden from one to the other. Therefore, in order to communicate they need to create a “subliminal channel” to embed messages in the traded letters in such a manner that the warden is unaware of the existence of any secret communication between the prisoners. In case the warden detects or suspects the presence of any such activities, he may possibly stop the communication process altogether. The process can be analyzed from the warden’s perspective as well. The warden may decide to tamper with all letters being transmitted in the prison in order to insure that no secret information gets transmitted. However, in doing so, legitimate communications between innocuous parties will also be adversely affected, which may be against prison policies. Furthermore, it may not be an efficient use of time and resources to look at every letter in order to detect secret communications.

Covert communications using computer network resources can be similarly described. The communicating parties assume the role of the prisoners and create a subliminal channel within the traffic between them and use it to embed their messages. The entities could be two machines physically separated by thousands of miles or two processes residing inside the same machine. The security policies of the network, like firewalls and operating systems, represent the prison warden. For the same reason, these

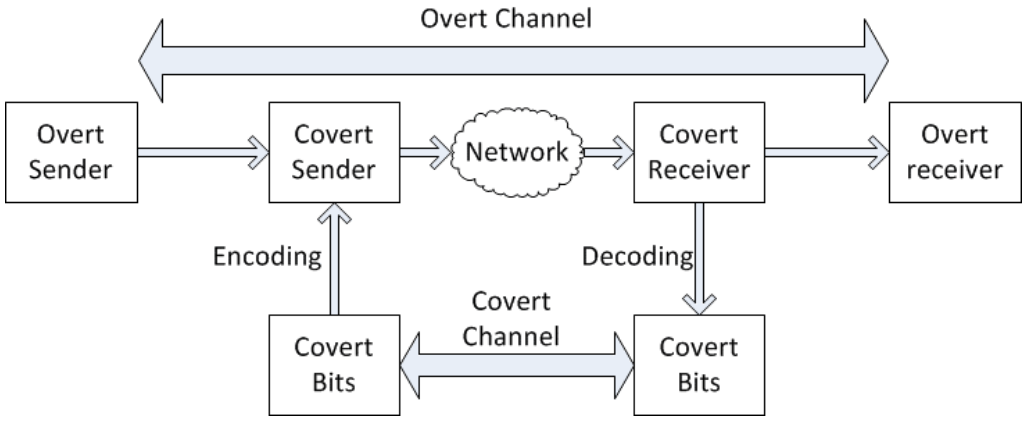

Figure 1.1: Conceptual Block Diagram Representation of a Covert Communications System

firewalls are also referred to as the network wardens. All network traffic must pass through the network warden, and the warden is unaware if any traffic flow has secret messages embedded in it. An active warden will attempt to change all traffic flow patterns in order to destroy the possibility of any transfer of messages. However, this will also impact innocuous traffic and will also lead to an inefficient use of computing resources. Therefore, a passive warden approach is preferred in which the focus is on traffic analysis and detection.

1.5 The Covert Communication Process

The process of covert communications can be generically modeled using a block diagram representation as shown in Figure 1.1. As shown in the figure, the overt sender and receiver are engaged in legitimate communications represented by the overt traffic channel between them. The covert sender hijacks the traffic flow and embeds his own message into it. The traffic, now categorized as covert, passes through its intended route. The intermediate network could be as small as a local network spanning across a few hops, or as large as a global network comprised of hundreds of nodes. The covert receiver uses a suitable decoding algorithm to extract the message from the traffic. It may or may

not attempt to remove the hidden information from the traffic flow. The traffic finally reaches its intended destination where the overt receiver is possibly completely unaware of all the modifications made to the traffic by the covert channel. Figure 1.1 only represents a logical separation between the overt and covert entities. The overt and covert senders could be two different machines, as well as two processes residing on the same machine. This is also true for the two receivers. A network warden, if present, will either be an integral part of the network between the covert entities or a separate process that taps into the traffic flow for analysis.

From the perspective of intended purpose and implementation, there are two types of covert communication models. The first one is used for leaking information, in which the covert sender is a machine or a process designated with a higher security level, e.g., a computer with access to a research institute's intranet and servers. In this case, the covert machine is a machine outside the aforementioned intranet, or a process that does not enjoy the security privileges as the sender process does. By using a subliminal channel, like opening a traffic port between the machines or changing the state of some local variable like processor temperature, the covert sender can leak the information it has accessed to the receiver, which does not have access to such computers. The second implementation models a secret communication between two parties. The covert sender embeds its message into an innocuous looking traffic channel, and the receiver extracts the message from the same traffic flow. Figure 1.1 models both of these implementations of covert channels.

1.6 Types of Covert Channels

Based on the type of system resources utilized for covert communications, there are two basic forms of covert channels—storage channels and timing channels.

1.6.1 Covert Storage Channels

In Covert Storage Channels (CSCs), the covert sender directly writes in the covert message (or some function of it, like an encrypted message) as values of some objects of the system resource. System resources—such as unused protocol header fields and extensions such as initial sequence numbers, IP identification, TTL, etc.—are used for embedding the covert information. The covert receiver is capable of directly reading out those values to extract the secret message.

An example of a typical CSC is leveraging unused and redundant TCP header fields, as shown in Figure 1.2. As explained in [1], the TCP initial Sequence Number can be used to embed covert messages. Whenever a host initiates a TCP session, a 32-bit random number is selected as the initial Sequence Number. A covert sender can send a character as a Sequence Number by multiplying the ASCII value of the character by 16777216.

<table>
<tr><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Acknowledgement Number</td></tr>
<tr><td>HLEN</td><td>Reserved</td><td>Code Bits</td><td>Window</td></tr>
<tr><td colspan="2">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="3">Options</td><td>Padding</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

Figure 1.2: TCP Header Format

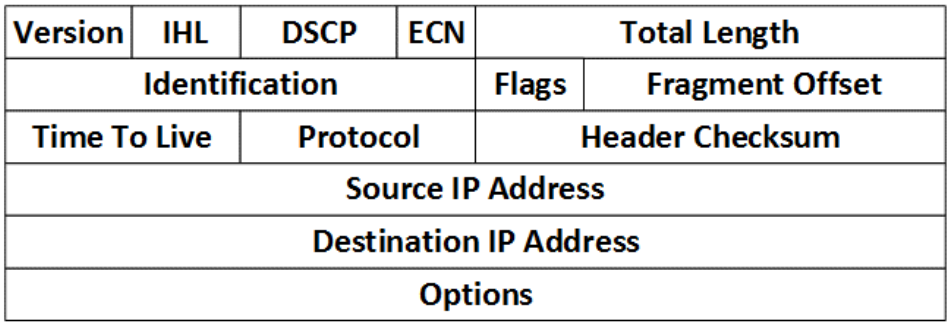

Figure 1.3: IP Header Format

Another popular method to transmit is using the Time To Live (TTL) field [2] of the IP header as shown in Figure 1.3. In computer networking, the TTL field is used to limit the lifetime of an IP packet, preventing it from living forever during routing loops. The TTL is set by the sender and decremented by one by each network element that the packet traverses through. The packet is dropped whenever the TTL value becomes zero. The covert sender chooses two TTL values—a high TTL to represent covert bit '1' and a low TTL to represent covert bit '0'. If the covert sender generates the packet itself, the default initial TTL value is used as the high TTL value. Similarly, if the covert sender lies in the middle of the path, it uses the lowest TTL of the intercepted packets as the high TTL value. The low TTL value is simply high TTL value minus 1. The receiver can decode the bits by observing the TTL values of the received packet.

1.6.2 Covert Timing Channels

On the other hand, in Covert Timing Channels (CTCs), the covert sender modulates the time-related properties of the system in accordance with the covert message. Modifying inter-packet arrival time, packet rate, hardware interrupt timing, etc., are usual methods of embedding secret messages using CTCs. The receiver observes the corresponding events with respect to time and decodes the message.

A typical method of implementing CTCs is changing the rate of transmission of packets by modulating the delay between two packets. It is possible to encode a binary channel by designating two inter-packet delays, one for each bit, e.g., 50 ms for covert bit '0' and 60 ms for covert bit '1'. The receiver simply has to observe the delay between the received packets to decode the bits. This process can be implemented as shown in Figure 1.4.

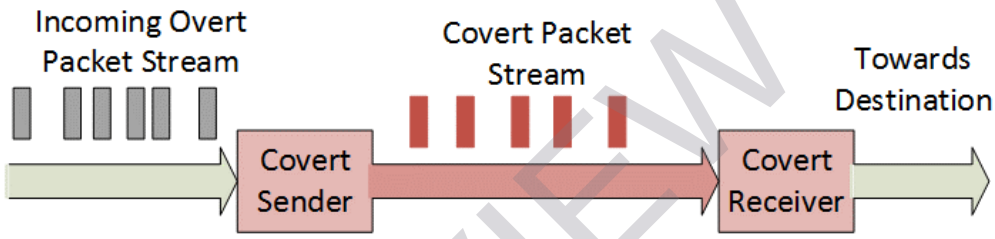

Figure 1.4: A Simple Implementation Covert Timing Channel

Timing channels are known to have more stealth, but storage channels provide higher capacity and robustness. Some researchers have designed a hybrid class of covert channel that combines both timing and storage channels. In terms of performance, hybrid channels fall between the other two types. The different varieties and implementations of covert channels are thoroughly reviewed in Chapter 3.

CHAPTER 2

PROBLEM STATEMENT

In this chapter, a description of the rationale behind undertaking this research project is provided.

As explained in Chapter 1, covert communications utilize side channels that were not even designed for the purpose of information transfer. Because of this, they can evade detection by conventional network security tools like firewalls and other intrusion detection systems. For example, a covert sender might decide to utilize the Initial Sequence Number (ISN) field to embed his secret message. A conventional firewall is not designed to classify traffic with respect to the content of the ISN field and hence it will not restrict the flow of such traffic. Similarly, a covert sender might embed data into the inter-packet arrival time. Again, the firewall will not restrict traffic based on the packet rate, and the traffic is not blocked.

The ability to use covert channels to transmit data without being detected makes them a grave security concern. Using such side channels, classified information such as government and military databases and secrets, industrial blueprints, research results, etc., can be leaked by simply establishing a connection from a machine belonging to the internal network with higher security privileges to an external network without access to such information. For example, a machine with access to the internal server storing the Social Security numbers of all customers can easily leak that information by establishing an HTTP session with an external webserver. To any conventional network warden, the session looks like an innocuous web surfing, but underneath, very confidential information will be leaked in a short interval of time. Similarly, covert channels facilitate

two-way communication between any two parties connected over the Internet. This means people with malign intentions, like terrorists, can readily communicate and exchange messages without being detected by authorities. This will again have serious implications for security. Therefore, it is of utmost necessity to detect and disrupt such communications.

In Chapter 3, a comprehensive survey of different types of covert channel algorithms is presented. Due to the sheer variety and number of covert channel algorithms, it is not possible to design detection mechanisms to deal with covert channels on a case-by-case basis. Such a design will be resource inefficient and impossible to implement, with unacceptable latency in detection. Furthermore, as newer algorithms of covert channels are introduced, the detection mechanism has to be updated regularly, which further worsens its performance. Therefore, it becomes necessary to design a mechanism that can be applied to detect any covert channel without the prior knowledge about the algorithm being used to embed the secret message. Unfortunately, no such detection mechanism has been reported yet. As shown in the survey in Chapter 3, the published detection mechanisms are domain limited and can be applied to a few covert channel algorithms. Furthermore, they lack extensibility and cannot be modified to detect a different type of covert channel, other than the one they were designed for.

To design a complete detection mechanism that can be applied to all covert channels, it is necessary to define them using a common framework. Only a detection mechanism built on the base of a common generalized framework that describes the covert communication process will realize in a detector that can be applied to all covert

channel algorithms residing in that common domain. However, no such common framework has been reported either.