Behavioral Mimicry Covert Communication

by

Seyed Ali Ahmadzadeh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Seyed Ali Ahmadzadeh

Abstract

Covert communication refers to *the process of communicating data through a channel that is neither designed, nor intended to transfer information.* Traditionally, covert channels are considered as security threats in computer systems and a great deal of attention has been given to countermeasures for covert communication schemes. The evolution of computer networks led the communication community to revisit the concept of covert communication not only as a security threat but also as an alternative way of providing security and privacy to communication networks. In fact, the heterogeneous structure of computer networks and the diversity of communication protocols provide an appealing setting for covert channels. This dissertation is an exploration on a novel design methodology for *undetectable* and *robust* covert channels in communication networks.

Our new design methodology is based on the concept of behavioral mimicry in computer systems. The objective is to design a covert transmitter that has enough degrees of freedom to behave like an ordinary transmitter and react normally to unpredictable network events, yet it has the ability to modulate a covert message over its behavioral fingerprints in the network. To this end, we argue that the inherent randomness in communication protocols and network environments is the key in finding the proper medium for network covert channels. We present a few examples on how random behaviors in communication protocols lead to discovery of suitable shared resources for covert channels.

The proposed design methodology is tested on two new covert communication schemes, one is designed for wireless networks and the other one is optimized for public communication networks (e.g., Internet). Each design is accompanied by a comprehensive analysis from undetectability, achievable covert rate and reliability perspectives. In particular, we introduced turbo covert channels, a family of extremely robust model-based timing covert channels that achieve provable polynomial undetectability in public communication networks. This means that the covert channel is undetectable against any polynomial-time statistical test that analyzes samples of the covert traffic and the legitimate traffic of the network. Target applications for the proposed covert communication schemes are discussed including detailed practical scenarios in which the proposed channels can be implemented.

Acknowledgements

I would like to thank my advisor, Professor Gordon Agnew, for all his support and guidance throughout my study and the development of this dissertation. His invaluable insight, broad vision, and continuous support helped me pass through difficulties and obstacles in my doctoral program. He has always seen merit in my work, and encouraged me to challenge myself by exploring new ideas in my research. He showed me how a great teacher can positively influence his students and taught me to be objective and constructive at the same time. I am truly honored to work with him, and I do appreciate all that he has done for me.

I also offer my sincere gratitude to the members of my thesis examination committee, Prof. Otman Basir, Prof. Mohamed Oussama Damen, and Prof. Eihab Abdel-Rahman for their kind advice and support. Indeed, it is my pleasure to acknowledge my external examiner, Prof. Scott Knight, for his comments on my PhD dissertation and his outstanding contributions to covert communication that inspired me during this project.

I had the privilege to be motivated by exceptionally talented colleagues, in particular members of the high speed and secure communication lab, Saad Alaboodi, Abdel Maguid Tawakol, Daniel Miller, Michal Skiba, and Jeffrey Woo. Thank you my friends for your companionship and the abundantly helpful discussions.

A big thank you goes to my family, specially my father, Seyed Ahmad Ahmadzadeh, and my mother, Bibi Ozra Nabavi, who have always valued education, and taught me perseverance, honesty, and integrity. I am grateful for your unconditional love, and it gives me great pleasure to present my finest work to you.

Last and foremost, I would like to express my deepest gratitude beyond words to my beautiful wife, Azin Ashkan, who stood by my side throughout this journey, and made me believe there is nothing I can't reach. Azin, you are my inspiration, and I cherish every moment spent with you. Without you, none of this would be possible.

Dedication

To

Azin

my love and true friend who has brought joy to my life!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer systems and the digital revolution undoubtedly have changed modern history. Computer networks, distributed systems, and cloud computing are symbols of an era in which instant access to information and the ability to process, store, and share huge volume of data are no longer optional properties for modern computer systems. However, such enormous flow of information requires extra attention to data security and privacy preserving technologies.

It is often believed that using *encryption* can provide security in computer systems. However, encryption only limits the access to the content and in turn gives away the fact that there exist some sensitive data that needs to be protected. In some cases, just a mere hint about the existence of information or a communication channel is enough to raise suspicion or trigger adversarial actions. Hence, consideration should be made in order to protect data by hiding signs of existence of sensitive information or communication channels.

The art of information hiding is about concealing the very existence of a message or a communication channel in order to protect data and prevent being detected by possible system observers. As an example, one can consider the application of spread spectrum techniques in military radio communications. The objective in such applications is to prevent the enemy to identify and locate the transmitter as it may direct attack towards

the signal source. *Steganography* and digital watermarking are other examples of information hiding techniques that are used in particular in multimedia and copy-write protection applications. In this way, the message that is intended to be hidden (e.g., sensitive information, or copy-write data) is embedded within contents of an innocent message often called *cover-data*. Thus, as long as the adversarial entity or the system observer (e.g., a censorship authority) does not have the key that is used in the embedding process, no information on the hidden message can be gained from the knowledge on the cover-data and the embedding process.

Another important sub-discipline of information hiding is covert channels. Covert communication often refers to the process of communicating data through a channel that is neither designed, nor intended to transfer information. In other words, a *covert channel* is a communication channel that violates the normal specifications of a system by using a shared resource or a communication algorithm in such a way that is not initially designed for data transfer purposes. Thus, the channel can evade detection since it uses resources that are not intended for data transmission and it should not exist in the first place. These channels are of a particular importance in evaluating the security of computer systems. In fact, covert channels have been traditionally used to leak sensitive information from processes with high access privileges (i.e., high process) to unauthorized recipients that normally do not have security clearance to access such information (i.e., low process).

Consider Alice and Bob to be two users of a multi-level secure (i.e., MLS) system. In a multi-level secure structure, the access privileges of each user are thoroughly specified by a security architecture. Hence, users with low access privileges (e.g., Bob) can not access data or services that are marked only for users with high clearance (e.g., Alice). Any attempt to access high level data by low processes will be monitored and blocked by the access control mechanism. In addition, Alice can not send high level information directly to Bob as such action is clearly against the access policy of the system. However, if the reverse case is not prohibited by the access control policies (i.e., if the low process is allowed to send messages to the high process), Alice can leak sensitive information to the low process (i.e., Bob) through the delay between consecutive acknowledgments that it sends to confirm the correct reception of Bob's messages. In this way, a long delay between

Ack messages represents covert bit-1, and a short delay denotes covert bit-0. Thus, a covert channel is formed in a direction that was initially disallowed by the access policy of the system.

It is worth noting that in contrast to the cryptographic games in which the ethical position of players is usually straightforward i.e., the "good" guys want to protect the data from "bad" eavesdroppers or adversarial entities, the situation is not as clear when it comes to covert channels. In more detail, while in some cases a user wants to detect and eliminate covert channels in order to prevent unauthorized information leakage in the system, there are cases in which hiding the communication channel can be used in order to improve user privacy, or to reduce the risk of adversarial attacks against sensitive information. It is even suggested to trace back attackers or malicious users of a system by secretly marking traffic flows in and out of the system using covert channels

1.1 Overview

Covert channels are usually created by exploiting weaknesses in conventional computer systems or communication protocols. A crucial step in designing a covert channel is to identify a resource that is (i) shared between the covert transmitter and the covert receiver, and (ii) can be modified by the covert transmitter in order to modulate the covert message. The shared resource should also play a major role in normal operation of the original system (e.g., randomness in resource sharing protocols). Thus, the access control mechanism should not be able to remove or strictly control the shared resource that is used for covert communication, without imposing drastic performance degradation on the system.

In general, two major forms of covert channels are defined based on the type of the shared resource, and how the transmitter uses that resource for covert communication. One category involves direct or indirect storage of the covert message (i.e., *storage channels*). The *file-lock* covert channel is a typical example of a storage covert channel in which the covert transmitter operates on a shared file in the system and changes the file access status to locked/released in order to modulate the covert bit-0 / bit-1, respectively. The

other category (i.e. *timing channels*), targets typical time-based properties of the system. These channels often exploit the resource consumption pattern of a user or a process in the system in such a way that the covert receiver interprets the covert message by analyzing the covert transmitter behavior. For instance, a simple timing covert channel can be formed by congesting the memory access bus with a burst of access requests in a known period of time to modulate covert bit-1 or leave the bus open by remaining idle to modulate the covert-bit0. It is noted that splitting covert channels into timing and storage channels is not a clear cut classification and can be extended by identifying new channels (e.g., *counting channels*) in which the count, the frequency, or the ordering of system events are used to embed the covert message into the overt channel.

The importance of covert channels becomes more noticeable when they are extended into computer networks. Modern communication networks benefit from relatively high data rate and cover a wide range of users that may be geographically spread across the globe. The large number of users, the diversity of services and the huge volume of network traffic make it relatively difficult for system observers to effectively monitor and process all communication channels in computer networks. In fact, the heterogeneous structure of communication networks and the diversity of communication protocols that are used in such systems provides an appealing setting for covert channels. This dissertation is an exploration of a novel methodology that is suitable to design *undetectable*, *robust* and *high rate* covert channels over public communication networks.

1.2 Motivation and Research Objectives

In traditional covert communication schemes (e.g., the file-lock channel) the undetectability of the channel mostly relies on the obscurity of the shared resource that is used as the medium for covert communication. For instance, in the file-lock covert channel, the channel is undetectable as long as the system observer does not consider the status of shared files as a potential medium for information leakage. However, if the shared medium between covert transmitter and the covert receiver is identified by the access control mechanism,

the channel may be compromised and it can be removed from the system. The situation is more critical in network covert channels, where the shared medium (i.e., the network traffic) is clearly used for communication purposes. Network traffic is easily accessible by monitoring entities that may record, process or even modify the traffic in order to thwart covert channels. In fact, there are numerous methods that are designed to detect, eliminate or limit the capacity of network covert channels. Therefore, *undetectability* becomes one of the main requirements of covert channels over communication networks.

Channel *robustness* is another important feature of a covert communication scheme, specifically for network covert channels. In communication networks, the covert channel shares network resources with many different traffic sources. Resource sharing is a key element in distributed systems (e.g., computer networks) in order to spread system resources in an efficient and fair manner to all users and services in the system. However, contention over the same resources may produce a temporal resource shortage that would turn into transient performance degradation of the system. Such unpredictable events change network characteristics and potentially affect the shared medium that is used for covert communication. For instance, it is known that the TCP/IP connections over the Internet are served with *best-effort* quality of service. It means that the network does its best to deliver the traffic to the destination, however it does not provide any guarantee on its service level. Hence, data packets may be delivered out-of order, in delay, or they may not be delivered at all. Now consider a covert transmitter that modulates the covert information over inter-packet delays (i.e., IPD) of its traffic over the Internet. Thus, any variation in the IPDs of the covert traffic can potentially force the covert receiver to an erroneous detection of the covert message. Furthermore, active adversarial entities may attempt to disrupt the channel by introducing jamming noise into the channel. In fact, it is shown that adding random delays into inter-packet delays of the covert traffic can effectively diminish the throughput of timing covert channels in communication networks. The goal in designing a robust covert channel is to deal with the inherent network noise (e.g., network jitter), and prevent active adversaries from disrupting the covert channel by introducing additional noise into the channel (i.e., jamming).

The exchange rate of covert information i.e., channel *covert rate*, is also of a great im-

portance in evaluating the effectiveness of covert channels. In fact, the reason to establish a covert channel is to exchange information without being detected. Thus, if the channel can not handle enough throughput to the covert receiver, even with high levels of undetectability and robustness, it can be hardly beneficial. However, little emphasis has been given to the covert rate and even robustness of covert channels. On the contrary, these properties are often scarified in order to achieve higher levels of stealthiness for covert channels.

Our research is inspired by the challenges that exist in designing robust, undetectable, and high rate covert channels in communication networks. To this end, the process of selecting shared resources and means of exploiting such resources to establish covert channels are studied. Then, novel design strategies for covert channels over public communication networks and wireless environment are explored. Each design is accompanied by a comprehensive analysis from different perspectives including undetectability, achievable covert rate and reliability of the covert channel. Target applications for the proposed covert communication schemes are presented including detailed scenarios for practical implementation of the proposed channels. In what follows, a more detailed introduction of each contribution of the thesis is presented.

1.2.1 Covert Channel Design Methodology

In this dissertation we explore a novel design methodology for covert channels over communication networks. Our methodology is based on the concept of behavioral mimicry covert communication in which the covert transmitter is expected to mimic the behavioral characteristics of a normal transmitter. The challenge is to design a covert transmitter that has enough degrees of freedom to behave like an ordinary transmitter in the system and react normally to unpredictable network events, yet it has the ability to modulate a covert message over its behavioral fingerprints in the network. To this end, we first highlight the requirements of a proper resource that can be used for covert communication, and describe our approach on how to find such resources in computer systems. In fact, we have identified the inherent randomness in communication protocols and network environments to be the key element in finding and exploiting the proper medium for covert channels in com-

munication systems. It is noted that randomness is used in the design of communication protocols in order to achieve fairness, stability, and scalability. Such sources of random behavior in communication systems can be exploited for covert communication in order to facilitate the modulation of the covert message over the behavioral characteristics of the covert transmitter. We have identified several sources of randomness in communication protocols (e.g., random elements in cryptographic algorithms), and provided a step-by-step approach on how to design a covert communication scheme according to the proposed design methodology.

1.2.2 Wireless Covert Channel

The emergence of mobile devices and wireless networks urges extra attention to the security of wireless communication. Wireless covert channels are rapidly gaining attention both as security threats and also as means of protecting sensitive information in wireless environment. It is noted that the wireless channel is broadcast in nature, hence all activities over the wireless channel can be monitored by receivers that are in the range of the transmitter's radio. In this way, covert channels have found applications in privacy preserving, and improving the reliability of the channel by hiding the communication channel from possible adversarial entities that would disrupt the channel otherwise. As an example consider implanted medical devices that use wireless channels to communicate information with outside world. Due to the critical role of such devices, they may be subject to adversarial attacks if their presence are revealed to the adversary. Covert channels can be a valuable tool in fortifying the defenses of such vital devices against security threats.

In this thesis we present a new wireless covert communication scheme according to the concept of behavioral mimicry and the proposed design methodology for undetectable, robust and high rate wireless covert channels. The objective is to adopt the medium access control protocol (i.e., CSMA protocol) of a normal wireless transmitter, and modify it in such a way that gives the covert transmitter enough freedom to embed a covert message into its overt traffic. The proposed covert channel is basically a timing covert channel that benefits from other user's activities in the system in order to transmit the covert message.

Since the packet transmission mechanism of the covert transmitter is influenced by channel activities of other elements in the system, the covert transmitter mimics not only the long term characteristics of a legitimate node, but also it reacts to the temporal changes in the system similar to a typical network transmitter. Thus, the covert channel remains hidden due to the indistinguishable packet transmission pattern of the covert transmitter as compared to any other node in the network. A salient feature of the proposed wireless covet communication scheme is that it can sustain relatively high covert rate that is scaled linearly with the overt rate of the communication channel.

1.2.3 Turbo Covert Channel

Timing covert channels over public communication networks (e.g., Internet) have also attracted significant interest recently. The enormous volume of information over the Internet, and the diversity of protocols and services make the Internet an ideal framework for covert channels. On the other hand, the unpredictable nature of public communication networks necessitates robust and reliable covert channel designs that are accustomed to dealing with extremely high level of network noise and adversarial disruptions. In this dissertation, a novel design approach for robust and undetectable covert channels over public communication networks is investigated. The proposed framework is based on modeling the timing covert channel as a differential communication channel and using the new communication model in order to derive the formulation for modulation/demodulation processes of the channel. The key element in this design strategy is the use of a trellis structure in modulating the covert message over inter-packet delays of the covert traffic. The trellis structure also enables the covert receiver to perform iterative demodulation and decoding of the covert message, and form a turbo covert channel that significantly improves the reliability of the channel.

Further analysis of the proposed turbo covert channel leads to introduction of an adaptive modulation strategy that improves the channel robustness even further. To this end, the covert transmitter takes advantage of the error-resistant portion of the IPD spectrum in modulating the covert message. However, in order to keep the hidden status of the

covert channel, the modulation scheme is designed such that it does not force the pattern of IPDs of the covert traffic to deviate from the traffic pattern of the legitimate traffic of the channel. Thus, the channel achieves *provable undetectability* which means that the covert channel is undetectable against any efficiently computable statistical test. The aforementioned modulation strategy combined with the proposed trellis structure gives the covert channel considerable flexibility to achieve a wide range of covert rates with extremely low decoding error rate at the covert receiver. In fact, the performance analysis of the channel reveals that the proposed covert communication scheme withstands extremely high levels of channel noise and adversarial disruption, while it maintains an excellent undetectability level and high covert rate.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows: Chapter 2 pretenses a detailed background on covert communication, the state of the art in covert channel design, and countermeasures for covert channels. The proposed design methodology for covert communication schemes is described in Chapter 3. The chapter also includes theorems and definitions that are used in this dissertation. Chapter 4 is dedicated to the design of a wireless covert channel. In this chapter a new wireless covert communication scheme is introduced and analyzed from robustness, stealthiness, and achievable covert rate perspectives. The chapter also includes a discussion on target applications for wireless covert channels in communication networks. The turbo covert channel design is detailed in Chapter 5. Finally, Chapter 6 concludes the dissertation and provides insights for future directions one could take to continue the contributions of this thesis.

Chapter 2

Background and Related Work

Computer systems depend on resources which are extremely limited when compared to the growing demands of users. Therefore, it is often suggested to implement resource sharing mechanisms in order to maximize resource utilization and efficiency of the system. In this way, computer systems have evolved into complex structures that share same resources among various users and services with different privileges and sensitivity levels. A very simple example of such systems can be found in a single CPU computer architecture in which different users (with different permission levels) share the same processing unit for their individual tasks. However, sharing resources enables users to open unauthorized communication channels (e.g., covert channels) in order to exchange information without proper supervision of the system. Hence, sensitive information could be leaked from a high level process or user into lower levels that are not authorized to access such information. Through the rest of this dissertation, it is always assumed that a *high* level user has higher clearance level than a *low* level user. This means a low level user is not allowed to access high level information.

Since the majority of applications of covert channels are of a malicious nature (e.g., leaking information, Botnet command and control channels), these channels are often considered as security threats in computer systems. To avoid such security threats, one can simply move to a totally isolated system and disallow the sharing of all resources between

different security levels [3]. However, this solution is simply too costly and inefficient for advanced computer systems. Alternatively, one can develop a controlled resource sharing mechanism like *multi-level secure systems* (i.e., MLS), in which the transactions between different security levels are well-defined and monitored [3]. In MLS systems classification labels are assigned to all objects (e.g., files) in the system and clearance levels are defined for all subjects that operate on system objects (e.g., users or processes). When a subject tries to access an object, it is granted access if and only if the clearance level of the subject supersedes the label of the target object. Such multilevel secure system is used by Bell and Lapadula [4] as an example for access control mechanisms.

The goal of access control systems is to *confine* users to their security levels and control user interactions with other users and processes. This problem is first stated by Lampson as *confinement problem* which is defined as the problem of preventing a service leaking information when it is serving a legitimate client of the system [5]. Lampson presented several examples of such information leakage scenarios and classified them into three general categories i.e., *legitimate channels*, *storage channels*, and *covert channels*. Linper [6] showed that information leakage through the storage and legitimate channels can be effectively countered by access control mechanisms in which access to the system resources are carefully evaluated and enforced. However, thwarting covert channels may be costly as it moves the system toward the total isolation strategy (e.g., one CPU should be dedicated to each service). In fact, many different solutions to the confinement problem, including *virtualization* [7] or *sand-boxing* [8], are practically simulations of totally isolated systems as they attempt to restrict the direct access between users of different security levels. Nevertheless, these methods lack the protection against indirect communication methods (e.g., covert channels) that may exists between different users in different security levels.

In practice, resource sharing mechanisms are the main target for covert channels. As an example, consider the *bus contention channel* [9], a covert channel that is created when multiple processes share a same memory bus to access a shared memory bank. Consider two processes with a high and low security levels that are running concurrently on two distinct processing units. It is noted that the access control mechanism prevents the low level process to directly access the high level process data even though they share the same

memory bank module. Despite the existence of the access control mechanism, the high level process can leak information to the low level process through the following process.

Bus Contention Covert Channel: In this design the time domain is sliced into intervals of length t milliseconds. During each time period, the high level process floods the bus contention system with access requests in order to transmit covert-bit 0, or it leaves the bus open with no access request in order to modulate the covert-bit 1. The low level process, simply generates a constant number of bus access requests and monitors how many of its requests are actually serviced by the system during each period. If the number of serviced requests are small/large, the receiver gleans the covert-bit 0/1 according to the known behavior of the high process thread. In this way, high level data can be send to the low level process with relatively high rates (i.e., $1/t$ *kbps*).

2.1 Covert Communication

It is often believed that encryption is sufficient to secure a communication channel. However, encryption only prevents unauthorized access to the channel content. In many cases, the existence of a communication channel, or even sudden changes in the traffic volume of the channel are enough to signal an event or raise suspicion. On the other hand, the art of information hiding and in particular *steganography* is about concealing the very existence of a message by means of hiding the sensitive information in the context of other information [10]. In fact, steganographic techniques usually aim to hide information within resources that are directly accessible by the receiver. Covert channels are also considered as an alternative form of information hiding methods, however they are different from steganographic techniques in the sense that they are usually used in systems where direct access to the actual resource (e.g., file content, or packet payload) is disallowed or at least controlled by access control mechanisms.

2.1.1 Covert Channel Definition

Covert communication refers to *the process of communicating data through a channel that is neither designed, nor intended to transfer information* [3]. Alternatively, the Orange book [11] defines the covert channel as *any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy*. The domain of this dissertation however, is focused on covert channels that exists in communication networks, where two distant nodes attempt to communicate covertly by modifying the communication protocols or node behaviors.

An *overt channel* is an authorized communication channel that transfers information according to a legitimate communication protocol (e.g., TCP/IP over the Internet). An overt channel may contain an embedded covert message (i.e., covert channel), or it may carry no covert message (i.e., legitimate channel). Figure 2.1 depicts the *prisoner's problem* that is proposed by Simmons [12]. This model is widely considered as the *de-facto* model for covert channel in communication networks. In brief, the prisoner's problem is defined as the challenge in front of Alice and Bob, two prisoners, who try to negotiate an scape plan. However, all communication between prisoners has to go through the warden (i.e., Wendy) that puts the prisoners into solitary confinement if she detects any suspicious behavior. The solution is for Alice and Bob to begin an overt communication channel with innocuous contents while a covert message is embedded into the overt traffic. In this way, the covert message could be carried to the receiver if the warden does not notice the existence of the covert message and forward the overt traffic to the receiver.

This model later was extended to communication networks in which Alice and Bob could be two, possibly distant, computer nodes in a public communication network (e.g., Internet) [13]. To this end, Alice and Bob use common network protocols (e.g., TCP/IP) in order to establish the overt channel and then modify some characteristics of the overt traffic (e.g., inter-packet delays) in order to modulate a covert message over the overt traffic. In practice, it is often assumed that Alice and Bob share a secret which is used in defining the covert channel parameters, and perhaps encrypting/authenticating the covert message.

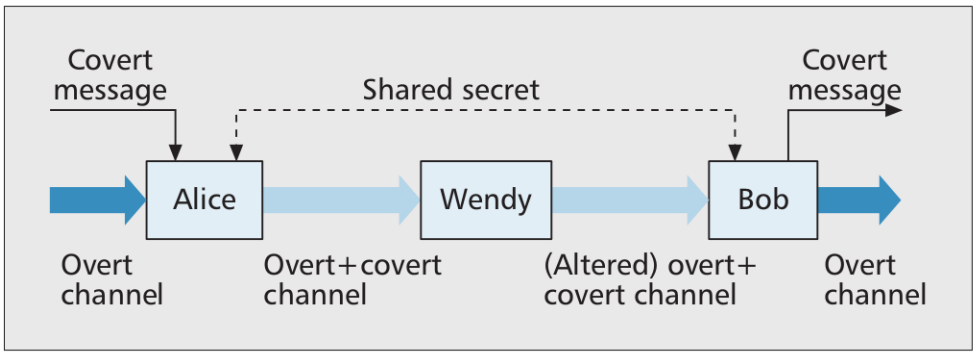


Figure 2.1: The prisoner's problem and covert channels [1].

2.2 Covert Channel Classification

Covert channels can be characterized based on many different features of the channel including shared medium that is used for the channel (e.g., Internet, or wireless channel), warden type (e.g., active or passive), or the type of resource that is used for covert communication (e.g., timing or storage channels). Figure 2.2 depicts a few possible parameters that may be used for covert channel classification.

Kemmerer [14] identified three necessary conditions for existence of a covert channel in a computer system. These conditions cover a wide variety of covert channels from those that exist in single host computer systems to more advanced covert channels in communication networks. In brief, these conditions can be described as follows:

Kemmerer Principle: A covert channel can exist between a covert transmitter and a covert receiver if the following conditions are met:

1. *There exist a global resource that is shared between the covert transmitter and the covert receiver.*

   For instance, the wireless medium is broadcast in nature and all the nodes in the wireless channel can intercept transmitted packets. Memory bus is also another resource that is shared among users of the same system.

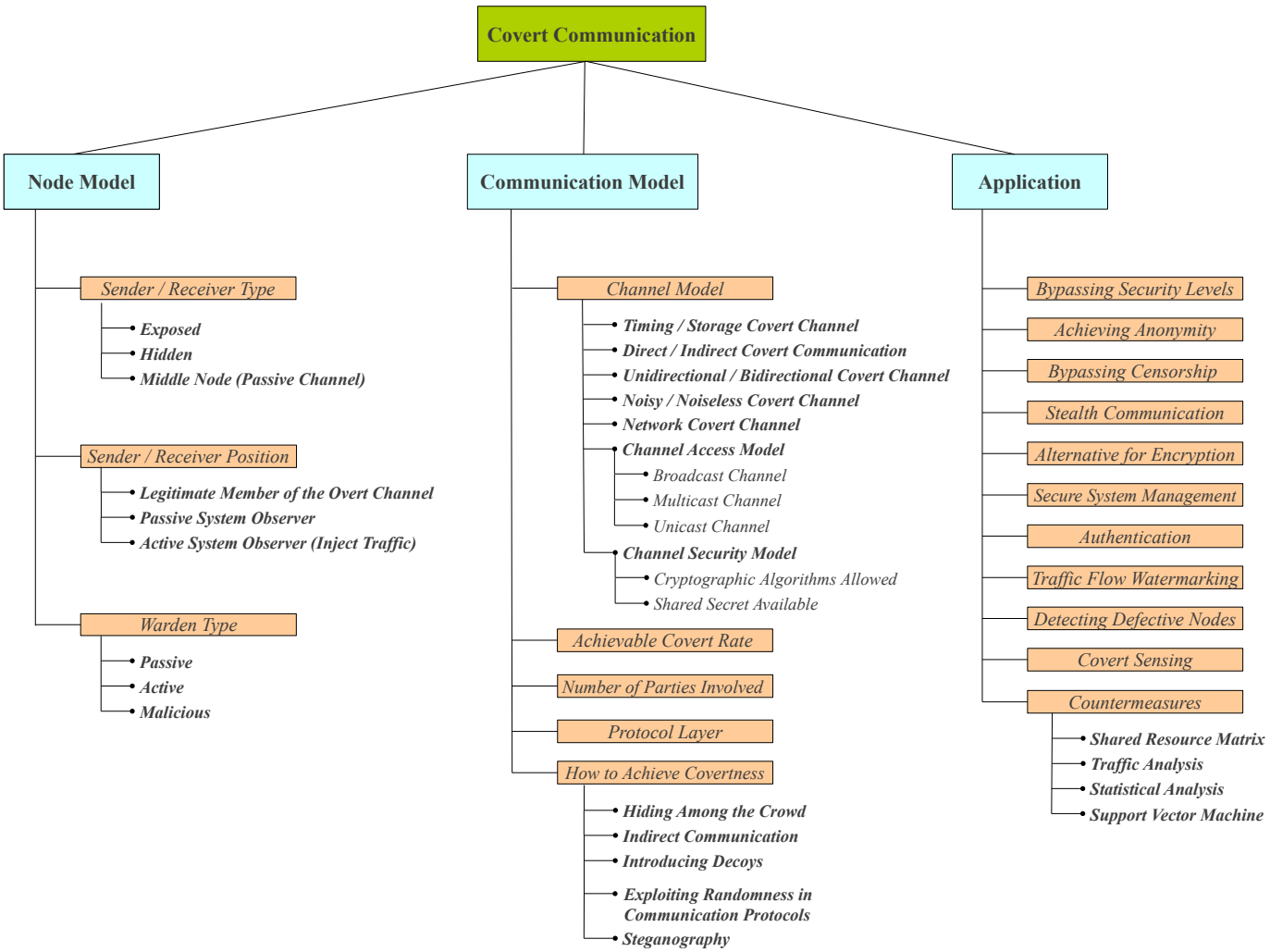


Figure 2.2: A sample classification of covert channels.

2. *Covert transmitter should be able to modify the shared resource such that the covert receiver can detect and identify such modifications.*

   As an example, a wireless capable node can transmit packets into the wireless channel and modify the state of the channel from idle to occupied. The header part of packets in communication networks (e.g., IP header) is also another resource that can be modified by the covert transmitter in order to transmit covert messages.

3. *The covert transmitter and the covert receiver need to have a method to achieve synchronization in order to communicate covertly.*

   The synchronization criterion covers the ability of the covert transmitter and the covert receiver to initiate the covert communication, and remain in synchronization state during the communication period. This also includes the ability to determine how long it takes for each bit of information to be transmitted from the covert transmitter to the covert receiver.

Using the Kemmerer's principle, one can classify covert channels based on how each channel design satisfies the aforementioned requirements for covert channels. In what follows, a few common examples of such classes of covert channels are presented.

2.2.1 Storage and Timing Covert Channels

It is a common classification approach to divide covert channels into storage and timing channels [1]. This classification is based on how the covert transmitter modifies the shared resource in order to embed the covert message into the overt channel. *Storage covert channels* often involve direct or indirect storage of the covert message into certain portion of the overt traffic (e.g., modify the header of an IP packet) [13]. The receiver look for the same section of the overt traffic in order to read the transmitter's fingerprint and decode the covert message. On the other hand, a *timing covert channel* targets typical time-based properties of the system, that are visible to the covert receiver (e.g., inter-packet delays, CPU usage), in order to modulate the covert message. The covert receiver decodes the

covert message by observing the behavioral fingerprints of the covert transmitter and its resource consumption pattern [15].

Dividing covert channels into storage and timing channels is not a clear cut classification and different definitions exist in the literature. However, this classification highlights two distinct approaches on achieving synchronization according to the Kemmerer's principle. In other words, timing covert channels, such as the bus contention channel, require a *reference clock* in order to measure time and define the pattern of resource consumption by the covert transmitter. However, storage timing channels overcome this obstacle by forcing a writable property of the system to change and remain constant until the covert receiver observes the modification.

The domain of storage covert channels is extensive. One of the earliest known storage channel in a single host computer system is called *file-lock* channel [16]. In this design the covert transmitter locks a shared file to transmit the covert-bit 1, and releases the file to send the covert-bit 0. Hence, the shared medium is actually the file, and the covert transmitter stores the covert message in the access flag of the shared file. The receiver, decodes the covert message by initiating an access request for the file and records if its request is granted/denied to access the file. In communication networks, storage covert channels can be established by modifying unused header bits of data packets such as reserved bits of IP header [13]. Other designs suggest to use parts of the overt traffic that does not have standard initial values. For instance, the don't fragment bit (DF) of the IP header [17], unused code fields of the internet control message protocol (ICMP) [18], IP identification (ID) field [19], or initial sequence number (ISN) field of TCP packets [20] are used as storage mediums for covert communication. A comprehensive list of such channels can be found in [1]. However, storage covert channels suffer from a fundamental design flaw as they systematically change the normal format of the overt traffic (e.g., header fields). Such sever modifications enable access control authorities to detect or at least limit the capacity of storage covert channels. In fact, in [21] it is suggested that using pattern recognition methods, a majority of storage covert channels in communication networks can be detected and removed from the system.

Timing covert channels on the other hand, are created by modulating the covert message over the timing of a sequence of events that are observable at the covert receiver. The bus contention covert channel is an example of timing covert channels in which the covert message is embedded in the state of the shared memory bus during fixed periods of time. Wray observed that timing channels crucially depend on the existence of a reference clock in order to achieve synchronization between the covert transmitter and the covert receiver [22]. It is noted that any sequence of events can be used to form a reference clock. For instance, a reference clock can be created by monitoring the instructions that are generated by the system CPU and consider each instruction as an event that increments the clock (i.e., clock ticks) [22]. Another example is to build a reference clock using an asynchronous disk controller. To this end, a large volume of access requests are issued to the disk controller in such a way that the I/O completion interrupts arrive at a regular pace to the issuing process. These interrupts can be used as clock ticks in order to form a reference clock for a timing covert channel [9].

With the reference clock in place, a timing covert channel needs a secondary sequence of events whose timing contains the covert message. This sequence of events is called the *signaling event stream*. For example, in the bus contention covert channel, the signaling events are the bus access response times at the covert receiver (i.e., low process). It is noted that the covert transmitter (i.e., the high process) controls the signaling events by flooding the bus with its requests (i.e., covert-bit 0) or remaining idle (i.e., covert-bit 1). Gray identified two classes of timing channels according to the relation between the reference clock and the signaling event stream [9]. The first class consists of a reference clock which has a much higher rate as compared to the signaling events. In this way, due to the high accuracy of the reference clock each signaling event can be individually timed. Therefore, covert information can be modulated on each individual event in the signaling event stream. A timing covert channel that encodes covert messages directly into inter-packet delays of an IP traffic flow is an example of such timing channels [23].

In the second class, the signaling event stream has a much higher rate than the reference clock. Hence, the timing of individual signaling events can not be measured accurately. In such cases the channel is exploited by counting the number of signaling events that

are observed during a known period of time according to the reference clock. In this way, the covert transmitter can modulate its covert message by controlling the number of signaling events that happens during each interval. These channels are called *counting covert channels* [9]. An on/off network covert timing channel [15] is an example of such class of covert channels in which the transmitter either transmits packets or remains silent in each time interval in order to transmit the covert message. The receiver counts the number of observed packets in each interval and records covert-bit 0 if its packet count is zero and records covert-bit 1 otherwise.

As a final remark, it is worth mentioning another variant of timing covert channels called *sorting channels* [19]. In this design the covert information is embedded in the ordering of the received signaling events during one or multiple timing intervals. Consequently, if the receiver can observe n different event during a given time interval, it has a message space of $n!$ depending on the ordering that these events are observed.

2.2.2 Noiseless and Noisy Covert Channels:

A covert channel is noiseless if the shared medium is dedicated to the covert transmitter and the covert receiver. Hence, there is no interference from other parties on the covert channel. In contrast, if the shared medium is accessible by other legitimate users or access control mechanisms, the covert channel is *noisy* meaning that there is a chance that the covert receiver decodes a message other than the covert message that is intended by the covert transmitter. For instance, in a file-lock covert channel, if another process can access the file that is used as the shared medium for the covert channel, it may toggle the state of the file forcing the covert receiver to record a wrong covert-bit.

It is noted that the noise described here (i.e., the *covert noise*) is different from signal noise of the communication channel. In more detail, the covert noise is generated based on other user's contention to access the shared resource or by an active adversarial entity who deliberately injects noise into the system in order to cause errornous detection of the covert message at the covert receiver. As another example, in the bus contention covert channel, memory access requests of other processors in the system may reduce the response time

at the covert receiver despite the fact that the covert transmitter is trying to keep the bus open by remaining idle.

The effect of covert noise on the capacity of covert channels has been vigorously studied in the literature [24]. In [24] the authors discussed the effect of jamming noise on the capacity of timing covert channels. They also proposed noise generating strategies according to a game theoretic analysis of the timing covert channels that significantly reduces the available bandwidth of the channel. Another example of using noise to control timing covert channels is an approach called *fuzzy time* [25]. In this approach, the jamming noise is introduced to the reference clock by means of random interrupt requests in different security levels. Random interrupts cause random clock shifts that disrupt the system clock in different security levels. Thus, the synchronization between the covert transmitter and the covert receiver is lost which makes the covert channel unreliable and extremely noisy.

2.2.3 Covert Channels in Computer Networks

The file-lock covert channel or the bus contention channel are examples of covert communication schemes in single host systems. However, the importance of covert channels becomes more noticeable when these channels are extended beyond the limited scope of a single host environment and into computer networks. The diversity of existing network protocols, the variety in resource sharing algorithms, and the high data rate of communication networks make them an appealing setting for covert channels. In fact, the scale of the shared medium (i.e., the network) makes it extremely difficult to have a global access control mechanism in the system. More importantly, the enormous number of users and services in communication networks necessitates the existence of resource sharing algorithms in order to efficiently allocate system resources to all network users. Thus, there is more possibility to create covert channels while the risk of being detected or denied by a global access control mechanism is significantly reduced. The majority of the contributions of this dissertation are focused on *network covert channels*.

In communication networks, information are often transmitted in data packets. A packet is a sequence of information bits (i.e., packet payload) that is augmented by one or

multiple protocol headers. The information in protocol headers are used to deliver each packet to its destination in the network. Hiding information in the packet payload is often regarded as a subset of information hiding through steganography which is out of the scope of this thesis. Designing storage covert channels using unused or uninitialized portions of protocol headers also received significant attention in the literature [1,13,26–28]. However, as suggested by Murdoch, the majority of these channels are vulnerable to the pattern analysis detection schemes [21]. Furthermore, the network routers and firewalls are usually configured to overwrite unused and reserved fields of packet headers in order to prevent covert channels.

To bypass these countermeasures, it is suggested to use the variable fields of IP or TCP headers that change from one connection to another. To this end, Rowland proposed to use IP identification fields or TCP initial sequence number fields as the medium for covert communication [29]. A more advanced version of such network covert channels benefits from a bounce back strategy in which the covert transmitter uses a middle, a bounce proxy, to communicate to the covert receiver. In this way, the transmitter generates a TCP SYN packet with an spoofed source IP address field that contains the address of the covert receiver. The ISN field in the packet header is set according to the value of the covert message and the packet is sent to the bounce proxy. Upon reception of the SYN packet, the bounce proxy sends a SYN/ACK or SYN/RST message to the covert receiver. The receiver simply decrements the value of the ISN field of the acknowledgment packet by one unit and decodes the covert message. Rutkowska [20] proposed to use encrypted covert messages in order to generate ISN fields with uniformly distributed values. Nevertheless, it is shown that the distribution of initial sequence numbers that are generated according to the encryption approach is notably different from the distribution of sequence numbers that a real operating system generates [21]. To tackle this problem, in [21] an implementation of an ISN based covert channel called *Lathera* is presented in which the distribution of the ISN header fields of the covert traffic mimics the same distribution as the traffic that is generated by a normal operating system (i.e., Linux open BSD).

Checksum fields and in general message integrity protection headers are also considered as possible medium for network storage covert channels. In fact, since the Checksum field

is calculated based on the packet payload, its contents can be considered random with no particular initialization value. Hence, it is a suitable field for storage covert channels as it can not be reset at the middle nodes in the network (e.g., routers and switches). Abad described how IP header Checksum fields can be used for covert communication [30]. In this approach, the header of IP packets are extended using IP header extension fields such that the calculated code redundancy check (i.e., CRC) of the header matches the value that corresponds to the covert message. The same technique is applicable to the TCP header Checksum. Rivest extended this idea to the message authentication codes by introducing *chaffing and winnowing* [31]. In this model, the transmitter and the receiver share a secret key that is used to generate keyed-hash digests of packet payloads. The transmitter can choose to send a packet with a matching hash digest or just generate a packet with random bits. The receiver verifies the hash-digest of each packet and drop all packets except those with matching hash digests. Although the original application for this scheme was to achieve confidentiality without encryption, this scheme can be modified to form a network covert channel. To this end, the covert transmitter appends the matching hash digest to its data packet in order to modulate the covert-bit 1, or it sends a packet (random or data packet) with a random hash digest to represent covert-bit 0. It is noted that as long as the shared secret remains secure, given the collision-resistant property of the hash function in use [32], there exist no system observer that can distinguish between a matching hash digest and a random bit stream. Hence, the covert channel remains undetected.

The potential of timing covert channel for stealth communication over data networks is known for more than three decades [15, 33]. Early timing covert channels were based on a simple idea in which the transmitter either transmits a packet or remains silent during a known period of time in order to communicate covert-bit 1 or bit 0, respectively [33]. An implementation of such timing covert channel was demonstrated by Cabuk [34] in which the synchronization between the covert transmitter and the covert receiver is achieved using a special preamble sequence at the start of each transmission period. However, the author explained that the simple on-off strategy is extremely sensitive to the accuracy of the reference clock and the synchronization between the covert transmitter and the covert receiver. More importantly, it was shown that due to the radical changes in the packet

transmission pattern of the covert transmitter, an on-off timing channel can be easily detected by a system observer.

Girling proposed to encode the covert message directly into the inter-packet delays (i.e., IPD) of the overt traffic in a LAN environment [15]. He also suggested to increase the inter-packet delays in order to mitigate the effect of network noise on the error rate of the covert receiver. Although this strategy can improve the channel robustness, it is mentioned that the capacity of the channel is inversely linked to the length of traffic inter-packet delays. Hence, there exists a tradeoff between the capacity and reliability of the channel. A salient feature of encoding covert information into the IPDs of the overt traffic is that it does not require accurate synchronization between the covert transmitter and the covert receiver. In fact, as long as the receiver can distinguish between long delays and short delays that are intended to modulate different covert bits, the channel operates normally. Berk *et al.* designed a variant of this strategy in which multiple delays can be used in order to modulate more information bits in each packet transmission interval [23]. They also proposed a mechanism in which the transmitter can pick the optimal encoding strategy based on properties of the channel (e.g., noise level, adversary model, packet transmission rate) in order to maximize the covert channel data rate.

Network timing covert channels are usually implemented in public communication networks such as the Internet. Hence, these channels should be designed based on the assumption that there exists a system observer with access to the covert traffic and also samples of the legitimate traffic of the same network. Based on such assumptions, in advanced timing covert channels the covert transmitter is designed to mimic the normal transmission pattern of a legitimate node of the network. Therefore, it would be less likely that the existence of the covert channel be detected due to the difference between the behavior of the covert transmitter and the ordinary elements of the system. Such design methodology is used by Cabuk [35]. To this end, a covert channel is designed in which the empirical distribution that characterizes the IPDs of the legitimate traffic is split into *small delay* and *large delay* halves. Then, the transmitter generates its covert traffic according to the large delay/small delay patterns in order to transmit the covert-bit 1/0, respectively. The keyboard Jitter Bug [36] uses a similar technique to leak the information entered through

the keyboard into a third party receiver over a public network such as the Internet. Compared to the simple on-off strategy of the elementary timing channels, these designs provide much higher level of stealthiness. Nevertheless, modulating the covert message over the IPDs of the overt traffic, affects key statistical attributes of the overt traffic such as distribution [35], entropy [37], and temporal characteristics of the covert traffic [38]. Such changes can be used in order to detect the covert channel.

To prevent detection, *model-based* covert channels were designed that aim to mimic the statistical behaviors of the legitimate traffic [39, 40]. In this way, the transmitter estimates a model that fits the pattern of IPDs of the legitimate traffic and uses this model to generate the covert traffic. The design in [40] incorporates a linear modulation scheme in which the parameters of the modulation formula are calculated based on the distribution of the inter-packet delays of the legitimate traffic of the channel. Hence, the covert transmitter can mimic the first order statistical behaviors (e.g., distribution shape) of the legitimate transmitter of the channel. The channel also benefits from a shared secret between the covert transmitter and the covert receiver that enables them to dynamically update the modulation parameters. In this way, the transmitter can influence higher order statistical properties of the covert traffic (e.g., traffic correlation) and blend itself among the legitimate users of the network. To achieve robustness against network noise and adversarial disruptions, channel coding (i.e., spreading code) is applied to the covert message prior to modulating the message over inter-packet delays of the covert traffic. However, in order to best fit a non-stationary traffic model of the legitimate traffic, this approach requires periodic updates of the model parameters between the covert transmitter and the covert receiver that may reduce the achievable rate and channel robustness.

In [41] a network timing covert channel called *liquid* is proposed. In this approach, the IPDs of the covert traffic are split in two parts. The first part is used to modulate the covert message in a similar approach as Jitter Bug [36]. The rest of the inter-packet delays (i.e., shaping IPDs) are used to mimic the statical behavior of the legitimate traffic according to the estimated model of the normal traffic of the channel. However, liquid is capable of exploiting only a fraction of the channel capacity as the shaping IPDs can not carry covert information. To tackle the shortcomings of *liquid*, Kothari [42] suggested to use a regularity

tree structure in order to control the correlation among inter-packet delays of the covert traffic in combination with a shape enforcing module. The shaping module forces the distribution of the covert traffic to mimic the statistical behavior of the legitimate traffic. Meanwhile, the regularity tree sets constraints on the acceptable sequences of inter-packet delays that the covert transmitter is allowed to use in modulating the covert message. In this way, the channel can evade detection by covert channel detection schemes that are based on the average conditional entropy of the covert traffic [37].

Recently, the focus of model-based covert channels has shifted towards designs with *provable undetectability*. This means that the covert channel should be undetectable against any efficiently computable statistical test that has access to samples of the legitimate traffic and covert traffic in the network. Sellke *et al.* [43] pioneered this design methodology and introduced a covert channel that is provably undetectable against any polynomial-time statistical test, assuming that the inter-packet delays of the legitimate traffic are independent and identically distributed (*i.i.d.*) random variables. The main idea is to use the output of a cryptographically secure pseudo random number generator (i.e., CSPRNG) and feed the result to the Quantile function of the distribution of the legitimate traffic. The result is a sequence of inter-packet delays that has a similar probability distribution function as the normal traffic of the network. In order to increase the achievable rate of the channel, geometric codes were designed that maximize the covert rate of the channel. However, such improvement is only possible under the strong assumption that the channel noise (i.e., network jitter) is bounded.

A high portion of the Internet traffic is made of HTTP traffic that reflects long range dependent (LRD) characteristics. In [44] a covert channel was presented that aims to mimic not only the marginal distribution of the legitimate traffic but also the long range dependency and correlation of the traffic samples. The design is based on modeling the web traffic connection start times using Fractional Auto-Regressive Integrated Moving Average (FARIMA) time series. The FARIMA model enables the covert traffic to achieve desirable LRD characteristics without disturbing the shape of the marginal distribution of the inter-packet delays of the covert traffic. However, due to the implementation of *persistent connection* functionality in modern web browsers and web servers, the TCP

connections that carry HTTP traffic last much longer than before. Hence, the rate of the new connection request packets in HTTP traffic has reduced significantly which in turn diminishes the achievable covert rate of the aforementioned covert channel.

Achieving robustness and high covert rate along with high levels of undetectability has been proved to be a challenge in designing covert channels [45]. In fact, in many covert channel designs there exists a tradeoff between the level of stealthiness and the effect of network noise on the robustness of the covert channel. In [46] the effect of different channel codes (e.g., convoloutional codes, linear codes) on the reliability of covert channels is analyzed. The same analysis using fountain codes and LDPC codes are presented in [47] and [48], respectively. However, the aforementioned schemes trade channel undetectability in favor of robustness which is undesirable in network covert channels. The design in [49] incorporates spreading codes in order to thwart the effect of the channel noise while maintaining provable polynomial undetectability. Nevertheless, this scheme requires a very strong channel code that significantly reduces the covert rate of the channel.

Smith and Knight [50] studied the application of convoloutional codes, a very well known family of channel codes [51], for improving the reliability of covert communication schemes. They argued that the traditional implementations of convoloutional codes may lack enough resistance against some of the error events that could occur in covert channels. In particular, they observed that the symbol insertion error event i.e., the error event in which a symbol that is exploited as a covert channel signaling element occurs due to the natural operation of the channel, can push the decoder at the covert receiver into a wrong path. Such an error event may cause loss of synchronization between the covert transmitter and the covert receiver that would eventually turn into high message decoding error rates at the receiver. To tackle this challenge, in [50] the authors proposed *toroidal codes*, a family of trellis codes based on a ring structure. In this way, the encoder state remains within a local ring if the next input bit is 0, and it jumps to an outer ring if the input to the encoder is a bit 1. This coding scheme is much more efficient in dealing with symbol insertion error events as compared to the traditional implementations of convoloutional codes. However, this approach is mostly effective for covert communication schemes that are vulnerable to symbol insertion errors (i.e., passive covert channels). An example of

such covert channels can be constructed by injecting a subset of naturally occurring events of the legitimate traffic (i.e., signal events) in order to modulate the covert message over the overt traffic of the network.

2.2.4 Wireless Covert Channels

Covert channels can also exist in wireless networks as the wireless medium offers unique properties for covert communication. In fact, the wireless environment satisfies all necessary conditions for existence of a covert channel according to the Kemmerer's principle. The wireless channel is broadcast in nature, meaning that the covert transmitter and the covert receiver can both detect channel activities that happen within their radio ranges. Thus, the wireless medium can be used as the shared medium for the covert channel. Moreover, the transmitter can change the state of the channel from idle to occupied simply by accessing the channel for packet transmission or jamming. Such modifications are noticeable at the covert receiver which monitors the status of the wireless channel. The broadcast nature of the wireless medium also enables the covert transmitter and the covert receiver to observe a wide variety of events (e.g., activities of other nodes in the network), and use those events as clock ticks in order to create a reference clock and achieve synchronization. Given such desirable properties, and with unprecedented growth in the number of wireless users and devices, designing wireless covert channels has attracted a lot of attention in the literature [52–54].

On the other hand, the same characteristics that make the wireless medium such a perfect environment for covert communication impose unique challenges in designing stealthy communication channels. For instance, the broadcast nature of the wireless channel enables a system observer to monitor all channel activities of the transmitters in the network. Therefore, detecting a covert transmitter based on its abnormal communication behaviors is much easier in wireless networks. The noise level in wireless channels is higher as compared to wired communication networks. Hence, consideration should be made in order to compensate for the error-prone nature of the shared medium in wireless covert channels. Finally, since the channel is easily accessible to other nodes in the network, performing

active adversarial attacks (e.g., jamming, packet injection, selective packet drop) is much more convenient for a malicious network element.

One approach in designing wireless covert channels aims to conceal the very existence of the communication signal by forming a signal that is similar to the background noise of the channel [55]. This technique is similar to the direct sequence spread spectrum (i.e., DSSS) communication system [56] in which the information signal is multiplied by a spreading signal that runs in a much higher rate as compared to the information signal. Since the spreading signal has larger bandwidth as compared to the information signal, it spreads the bandwidth of the transmitted signal. Hence, the peak power of the transmitted signal is reduced considerably. This reduction in signal power can be used in order to hide the wireless signal behind the background noise of the channel. However, the DSSS signal relies on a periodic carrier wave for transmission over a wireless channel. The periodic carrier forces the transmitted signal to become cyclostationary. Thus, squaring the signal makes it much easier to detect which is a drawback for a covert channel design. In [55] it is suggested to use chaotic systems [57] to generate the carrier signal as an alternative for the traditional periodic carrier waves.

A chaotic system is a non-linear dynamic system that has extreme sensitivity to its initial conditions [58]. Two autonomous identical chaotic systems that start in *nearly* same initial conditions, would choose two completely different trajectories and become uncorrelated very quickly. The well-known Lorenz system is a series of differential equations with chaotic characteristics [59]. Chaotic systems have many intrinsic behaviors that makes them specifically attractive for secure communication. These systems provides *noise-like* behaviors and are extremely sensitive to initial conditions. Therefore, a considerable research was done on developing a real world circuits with chaotic behaviors [60]. Followed by the discovery of the self-synchronized chaotic circuits by Pecora and Carroll [61] the application of chaotic systems for secure and covert communication has been intensely studied and few schemes were developed [55, 58, 62]. However, the need for perfect synchronization between the transmitter and the receiver in chaotic covert communication schemes limits the application of such schemes in wireless environment due to the high noise level of the wireless channel.

Storage wireless covert channels follow the same design criteria as storage channels in wired networks. In addition to the existing storage covert channels that exploit header fields in higher protocol layers, the MAC layer of wireless communication protocols (e.g., IEEE 802.11) can also used for covert communication [52, 63]. In [53] it is proposed to use the destination address field of unsolicited IEEE 802.11 acknowledgment frames as the medium for covert communication. The authors also suggested to use invalid frames (i.e., frames with incorrect checksums) as an enhancement to their covert communication scheme. Li *et al.* proposed to exploit the routing protocols in ad-hoc wireless networks and create covert channels with applications in authentication services [64]. In this approach, the covert message can be modulated over different fields of the route request or route reply frames of the routing protocol. The authors proposed to use the source sequence number, request lifetime, and also the destination ID section of the route request messages. Nevertheless, wireless storage covert channels are vulnerable to pattern matching detection schemes as the transmitted packets are easily accessible by network monitoring entities.

Wireless network protocols often allow high degrees of randomness in order to share system resources in an efficient and fair manner. Combine such randomness with the diversity of network protocols, number of users, and heterogeneous system architectures, wireless networks form a ubiquitous environment that offers a wide variety of resources for covert communication schemes. A covert channel based on exploiting randomness in binary collision resolution algorithm was described in [65]. To this end, the covert message is transmitted by the number of collisions that is observed at the covert receiver during the collision resolution period. A similar approach based on splitting tree collision resolution algorithm is introduced in [66]. In this approach, the covert transmitter is configured to choose a particular path in the splitting tree according to the covert message. The receiver on the other hand, decodes the covert message through the relative position of the covert transmitter in the tree. Later, Wang *et al.* [67] extended the aforementioned design into an anonymous covert channel in which the receiver decodes the covert message using a specific voting approach that considers the probabilistic decisions of multiple covert transmitters in the network. Rate switching protocols [68], packet retransmission strategies [69], and the size of back-off timer in the contention resolution algorithm of IEEE 802.11 protocol [70]

are among other wireless protocols/schemes that can be used for covert communication in wireless environment. In [71] a covert channel based on signal jamming over slotted ALOHA was presented. To this end, the covert transmitter influences the collision rate of the network by introducing jamming noise into the wireless channel. The receiver decodes the covert message by analyzing the packet loss pattern in the channel that is controlled by the covert transmitter. The authors also presented a detailed information theoretic analysis of their covert communication scheme and derived the upper bound on the capacity of the covert channel.

2.3 Covert Channel Countermeasures

In designing a covert channel special consideration should be made in order to maintain the stealthiness property of the channel and protect the identity of the covert transmitter and the covert receiver. Therefore, understanding covert channel countermeasures and detection techniques play a major role in designing undetectable and efficient covert communication schemes. On the other hand, from a secure system design point of view, identification of covert channels is equally important in order to prevent information leakage from high priority level processes to the lower levels. Thus, a great deal of attention has been focused on developing efficient methods for identification of weaknesses and design oversights that may be used for covert communication [72].

A majority of covert channel identification methods were first developed for single host systems. Such identification schemes include lattice based formal method identification techniques [73], covert flow tree (CFT) data structures [74], and shared resource matrix (SRM) methods [14]. The SRM method is important since it can be used to visually identify system dependencies that may be exploited by covert channels in the system [75]. To form a shared resource matrix structure of a particular system, each attribute of the system is assigned to a row in the matrix while the columns of the matrix are dedicated to the operations on each attribute of the system. Hence, the relation between operations and attributes are reflected in the elements of the shared resource matrix. In this way, one

can focus on identifying an implicit flow of information by calculating a transitive closures of the resource matrix of a particular system. Kemmerer argued that this approach can be used to identify timing channels provided that the system time is included as one of the attributes in the matrix. The application of the SRM approach in identification of network covert channels was discussed in [76], where the authors proposed to split the communication channel into host-by-host segments and inspect each segment separately. However, most of covert channel identification methods rely heavily on information that is manually acquired and processed. Hence, it is highly probable that some attributes of the system are neglected and overlooked during the design phase of the system.

In general, covert channels exist due to either design oversights or inherent weaknesses of the system [77]. The design oversights can be fixed once they are identified [72]. Therefore, covert channels that are designed based on such weaknesses can not have long life spans and will be removed easily when their target systems are upgraded. However, the intrinsic vulnerabilities of the system can not be removed without redesigning the system which may be costly or impractical in some scenarios. Thus, it is often conjectured that covert channels can not be completely removed from computer systems due to the implication of inherent design flaws in distributed systems [78]. Alternatively, one can aim to limit the capacity of covert channels, or provide means of channel detection in order to thwart the unwanted effects of such security threats in computer systems. In particular, detecting network covert channels has attracted a lot of attention since the diversity of network structures and users makes it increasingly difficult to control covert communication schemes over computer networks. In what follows, a brief discussion on elimination, limitation, and detection schemes for covert channels is presented.

2.3.1 Eliminating Covert Channels

Eliminating a covert channel is possible by removing or blocking access to the shared resource that is used by the covert communication scheme. For instance, filtering the ICMP network traffic can effectively eliminate covert channels that rely on ICMP packets as the medium for covert communication (e.g., *Loki* channel [18]). Obviously, applying

such policy is not possible on other network protocols (e.g., IP, TCP) that form the basic building blocks of public communication networks. As another example, it can be observed that bouncing covert channels, including the channel that is described by Rowland [29], exist only if the network allows packets with arbitrary source address fields. However, it is a common practice to configure routers and switches in public communication networks to drop packets with source address fields that do not belong to the network path in which the packets are received. Hence, the covert channel is removed as the proxy node can not be driven properly. Storage network covert channels that exploit unused or uninitialized packet header fields are also subject to elimination by traffic normalization. In this way, network middle nodes (e.g., routers, firewalls) overwrite header fields of the overt traffic in order to eliminate network storage covert channels. For instance, network routers can be configured to set the identification header field of IP packets to zero if the DF bit of the header is set. Also, time-to-live (i.e., TTL) header fields can be set to arbitrary values for destinations that are known to be close in the network. Thus, the covert channel would be eliminated as the shared resource does not exist anymore.

2.3.2 Limiting the Capacity of Covert Channels

Removing covert channels is not always the best possible solution as it may cause extreme performance degradation in the system. In some scenarios limiting the capacity of the covert channels in the system may be a better choice. Thus, one can reduce the risk of leaking information through covert channels, yet assure certain level of system performance. In order to effectively limit the capacity of a covert channel it is often useful to calculate the formulation of the channel capacity or at least identify the major parameters that dictate the rate of the channel.

Millen estimated the capacity of timing covert channels by modeling timing covert channels as legitimate communication channels and applying normal information theoretic approaches [79]. Moskowitz extended this analysis to the discrete memory-less timing channels [80]. Another formulation for the capacity of timing covert channels and its application on limiting covert channels is studied in [24]. In this approach, the timing

channel is modeled as a game between the covert transmitter who intends to modulate the covert message over inter-packet delays of the overt traffic and a jammer (i.e., an active adversary) who wants to limit the capacity of the covert channel through jamming. The objective of the game is the mutual information between the input and the output of the chanel, where the covert transmitter/jammer attempts to maximize/minimize the game objective. The authors also provided coding strategies for the covert transmitter and the jammer in different game setups and channel configurations.

Gray studied the capacity of bus contention covert channel in the presence of *fuzzy time* disruption [9]. In fuzzy time attacks, the accuracy of the system clock is randomly affected by random interrupts at different security levels [25]. Gray also proposed another capacity limiting approach (i.e., *probabilistic partitioning*) in which the accuracy of the reference clock remains intact, however the bus controller randomly changes the access policy for the shared bus from a *secure mode* to a *normal mode* and vice versa. In the secure mode different security levels have dedicated time partitions to access the bus, while in the normal mode processes from different security levels share the bus in order to improve system performance by resource sharing.

In the bus contention covert channel, the high process controls the response time of the low process by selectively congesting the access bus or remaining idle. A more direct approach in designing covert channels can be used if the message from low level process has to be acknowledged by the high level process. In this way, the high process (i.e., the covert transmitter) can create a timing covert channel by manipulating the timing of its Ack messages. To prevent such covert channels, a middle node is placed between the low and high processes that accepts the messages from the low process, sends acknowledgment to the low process, forwards the message to the high process, and removes the message from its buffer when the high process acknowledges the message. In this way, the middle node aims to break the dependency between the timing of Ack messages that are generated by the high process and the Ack messages that are received at the low process.

However, this store and forward protocol (SAFP) is limited by the size of the buffer at the middle node. In other words, if the buffer is full, the middle node has to wait

until it receives an acknowledgment (from high process) for one of the messages in the buffer before it can accepts another message from the low process. Hence, the dependency between the Ack messages transmitted by the high process and the ones that are received at the receiver can be restored. The low process floods the middle node by a sequence of messages until it does not receive any more Acks (i.e., the buffer is full). From this point forward, the middle node can accepts new message, from the low process, only if it receives acknowledgment for previous messages from the high process. Therefore, the high process can control the rate of sending Ack messages to the low process by manipulating the rate of generating Ack messages. In this way, to send a covert-bit 1, the high process sends an Ack to remove a packet from the buffer and allows the middle node to accept (and acknowledge) one packet from the low process. To send a covert-bit 0, the high process remains idle and leaves the buffer in full state.

Data PUMP [81] is a store and forward buffer structure that is designed to eliminate such covert channels by adding probabilistic delay to the pattern in which the low process receives its Ack messages. In brief, the PUMP uses the statistical behavior of the covert transmitter (i.e., the high process) in sending Ack messages and mimics the same behavior when it acknowledges the message that are received from the low process. To this end, upon receiving a packet from the low process the trusted low process puts the message in queue and wait for a random amount of time before sending the Ack to the low process. The delay is based on a modified exponential distribution with the same mean as the average inter-acknowledgment delay of the high process [81]. The mean value is updated every time a new acknowledgment is received at the PUMP. Hence, the low process can not overflow the buffer and the PUMP decouples the timing of acknowledgments that are received from the high process and the ones that are sent to the low process. Figure 2.3 depicts a schematic view of a simple PUMP system. The trusted low and the trusted high nodes are gateways that control the input to the PUMP.

The requirement to acknowledge the received packets exists in computer networks as well. In fact, the reliability of the TCP protocol, which handles most of the Internet traffic (including web traffic), is based on the receiver to send acknowledgments for the packets that are received correctly. Network PUMP [82] is the network version of the simple PUMP

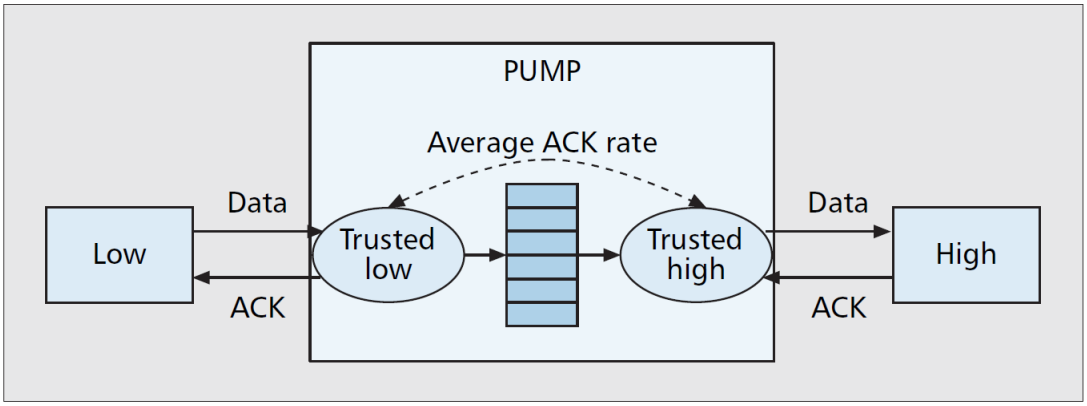


Figure 2.3: The simple PUMP architecture [1].

that is designed to decouple the timing of Ack messages in order to limit the capacity of network timing covert channels. The design of network pump is similar to the simple PUMP. However, the network PUMP is capable of handling multiple data links from a group of source nodes to different destinations in the network. The design also provides fairness, congestion control capability, and it is resilient against denial of service (i.e., DoS) attacks.

The network pump model is used in to analyze the effect of active adversarial disruptions on network timing covert channels [83]. It is noted that an active adversary can effectively reduce the capacity of timing covert channels by introducing random noise into the inter-packet delays of the overt traffic [43]. However, such disruption may have a catastrophic effect on the overall capacity of the network [24]. In Chapter 5, a design methodology for network covert channels is presented that can tolerate extremely high levels of channel noise and adversarial disruptions while it maintains an excellent undetectability level and covert rate.

2.3.3 Detecting Covert Channels:

Covert channel identification often requires manual system analysis or inputs that have to be prepared according to the system configuration and network scenarios. Since human

factor is involved in this process, some aspects of the design may be overlooked and a number of covert channels remain undetected. Channel elimination and control techniques may impose high performance degradation on the system that is not acceptable for high performance and distributed systems. Covert channel detection is an alternative in thwarting unauthorized communication channels in computer systems. The key in covert channel detection is to pinpoint anomalies in the overt channel that signal the existence of a covert channel. From the covert channel design point of view, it is often assumed that the system observer (passive or active) is aware of the covert communication scheme, and it knows the normal behavior of the legitimate channel. Therefore, the covert traffic should have similar characteristics as compared to the legitimate traffic in order to prevent detection.

To detect abnormal behaviors of a covert traffic flow, a system observer may use statistical analysis tools to compare the legitimate traffic with the covert traffic. These tools are often classified in two different categories, namely *shape tests* and *regularity tests*. Shape tests study first order statistical metrics of the covert traffic (e.g., mean, variance, *etc*) and compare them with similar results that are calculated from samples of a legitimate traffic flow of the same system. On the other hand, regularity tests are mostly focused on higher order statistical characteristics of the traffic flow such as correlation metrics.

Kolmogorov-Smirnov Test:

The Kolmogorov-Smirnov test (i.e., KS-test) is a well-known statistical measure that determines if a set of sampled data comes from a particular distribution or not [84]. In other words, given the distribution of the legitimate traffic of the network (e.g., the distribution of inter-packet delays of a normal traffic flow), the KS-test distinguishes traffic samples that follow the distribution of the legitimate traffic from samples that have slightly different distribution due to the existence of an embedded message (e.g., covert channels). A salient feature of the KS-test is that it is not based on any *a priori* assumption about the distribution of the legitimate traffic and the covert traffic. Hence, it can be effectively used against any arbitrary distribution. Let $S(x)$ be the empirical distribution of the samples that are taken from the target traffic (i.e., possible covert traffic flow). Let $F(x)$ be the

cumulative distribution function of the legitimate traffic in the same system. The KS-test is defined as follows:

$$H_s = \sup_x |S(x) - F(x)|. \tag{2.1}$$

The application of the KS-test in detecting timing covert channels was demonstrated by Peng *et al.* [85]. The authors showed that using KS-test it is possible to detect watermarked inter-packet delays (i.e., a network timing covert channel) that is used to trace back a traffic flow to its original source.

Regularity Score:

In addition to the first order statistical tests (e.g., KS-test) another statistical measure called *regularity score* was introduced in [35]. The regularity score is designed to detect the temporal abnormal behaviors of the covert transmitter. In general, public communication networks (e.g., Internet) are designed based on a stateless architecture. This means that the network condition can change suddenly and drastically at any time. For instance, a sudden surge in the number of users may cause network congestion that reduces the available bandwidth dramatically. Network jitter, packet loss, and out of order delivery also contribute to the unpredictable nature of communication networks. A normal user reacts to these unexpected interrupts according to the network protocol specifications (e.g., a TCP connection drops its rate to prevent congestion). However, a covert transmitter is committed to modulate a covert message over its traffic flow. Hence, it may not be possible for the covert transmitter to react freely to such unpredictable network events. The regularity score is designed to catch such behavioral differences in order to detect covert channels.

To calculate the regularity score of a given traffic flow, samples of the traffic are distributed into different sets, where each set contains γ samples. Then, the variance of the samples in each set is calculated. Let σ_i denotes the variance of the samples in the i^{th} set. In this way the regularity score (i.e., H_r) of the traffic flow is derived as follows:

$$H_r = std(\frac{|\sigma_i - \sigma_j|}{\sigma_i}, \quad \forall i, j, i < j). \tag{2.2}$$

Where, *std* is the standard deviation operation. In general, a high regularity score signals a noticeable change in the variance of the samples in different sets. This is most likely due to variable network conditions that affect the behavioral characteristics of normal users of the system. On the other hand low values of the regularity score, depicts highly regulated traffic samples that may contain embedded information, perhaps a covert message.

Entropy Based Detection:

Gianvecchio and Wang [86] proposed a covert channel detection method based on the entropy of traffic samples. The idea is to compare the randomness of the samples of the covert traffic and the legitimate traffic of the same system. This approach is primarily designed to detect network timing covert channels in which the covert message is modulated into the inter-packet delays (i.e., IPDs) of the covert traffic. Furthermore, the authors proposed another detection scheme based on the conditional entropy measure of the sampled traffic. The latter approach aims on detecting irregular correlation between consecutive IPDs of the covert traffic as compared to the samples of the legitimate traffic of the overt channel. It is noted that the term *entropy* refers to the Shannon's measure of entropy [87] which represents the average amount of information one requires to determine a random variable without any ambiguity.

Let $X = (X_1, X_2, ..., X_m)$ be a sequence of random variables with n possible outcomes from the alphabet $\mathcal{X}$ (i.e., $|\mathcal{X}| = n$). The entropy of X (i.e., $H(X)$) is defined as follows:

$$H(X) = -\sum_{X_1, X_2, ..., X_m} P(x_1, x_2, ...x_m) \log(P(x_1, x_2, ...x_m)), \tag{2.3}$$

Where, $P(x_1, x_2, ...x_m)$ is the joint probability mass function of the elements of X (i.e., $P(X_1 = x_1, X_2 = x_2, ..., X_m = x_m)$, $x_i \in \mathcal{X}$). It is easy to verify that the entropy measure is maximized when all possible sequences are equiprobable. A very complex process often has a high entropy signature (e.g., a sequence of (*i.i.d.*) random numbers). In contrast a rigid and highly regulated process has much lower entropy. The conditional entropy of sequence X is also defined using Equation (2.3) as follows:

$$H(X_m | X_1^{m-1}) = H(X_1^m) - H(X_1^{m-1}) \tag{2.4}$$

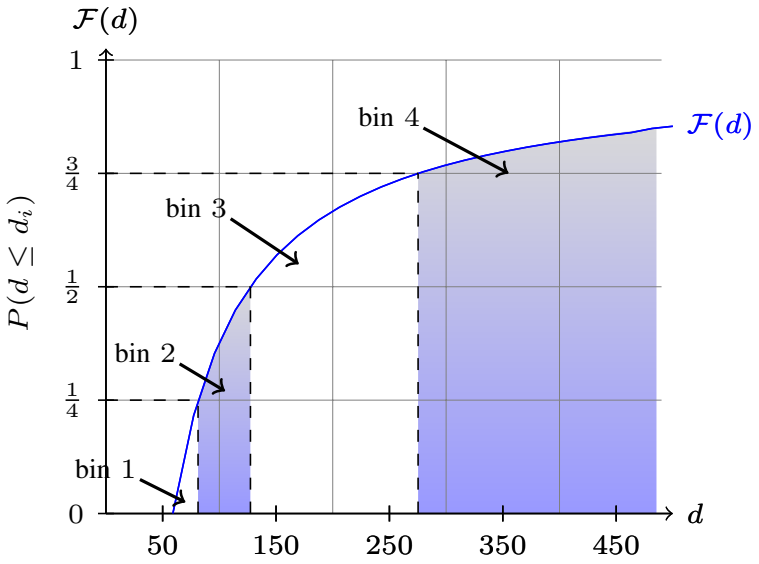


Figure 2.4: Equiprobable bins of inter-packet delays for $Q = 4$.

Where, $X_i^j = (X_i, X_{i+1}, ..., X_j)$, $i \leq j$. In other words, the conditional entropy $H(X_m|X_1^{m-1})$ denotes the amount of information requires to determine sample X_m, given complete knowledge on all previous elements in X (i.e., X_1^{m-1}).

To perform entropy based detection, the probability of occurrence of the sequence X should be derived. To this end, the domain of the sampled traffic (e.g., inter-packet delay) is partitioned into Q equiprobable bins, each of which is identified by a unique number (i.e., bin ID). Each bin covers a non-overlapping range of possible values for the sampled parameter Thus, the corresponding bin ID to the sample value d_i (i.e., q_i) can be calculated as $q_i = \lfloor \mathcal{F}(d_i) \times Q \rfloor$, where $\mathcal{F}(d_i) = P(d \leq d_i)$. Figure 2.4 depicts the binning strategy for samples of inter-packet delays. According to the aforementioned binning strategy, one can map a sequence of samples (i.e., X) to a sequence of bin IDs that cover the elements of X. To derive the estimated probability of a specific sequence of samples (i.e., X), the number of occurrences of the sequence of bin IDs that correspond to the elements of X is divided by the total number of unique bin ID sequences of size $|X| = m$ that are observed according to the set of sampled traffic.

2.4 Chapter Summary

In this chapter we reviewed the state of the art in covert communication. We introduced the general idea, definition, and the early versions of covert channel designs in computer systems. The traditional view on covert channels was to consider them as security threats. Hence the focus was initially on the countermeasures for eliminating, limiting, and detecting covert channels. We discussed different classification methods for covert channels and studied few categories that are commonly used in the literature, including the storage and timing covert channels. The early design approach for covert communication schemes were based on the concept of *security through obscurity*. This means that the covert channel is hidden since the communication medium that is exploited by the channel is not expected to be used to exchange information. Thus, it is not being monitored by access control authorities. However, if the channel design and the resource it uses for covert communication is identified, opposing the covert channel will be straightforward. We also reviewed the requirements for a proper resource that can be used as the covert communication medium, and provided several examples of such designs in computer systems.

Network covert channels have attracted a lot of attention due to the popularity and the scale of communication networks. In fact, network covert channels force the network security community to revisit the idea of covert communication, however this time the trend is to consider these channels as an alternative solution to provide security and privacy in communication systems. We presented several examples of covert channel designs in computer networks, and discuss different challenges that exist in designing robust and undetectable network covert communication schemes. Finally, we turned our attention toward countermeasures for covert channels specifically in computer networks, and provided a detailed description of methods that are designed to eliminate, control, and detect network covert channels. In the next chapter, we present our design methodology for network covert channels in public communication networks.

Chapter 3

Covert Channel Design Methodology

In Chapter 2 we discussed the variety of covert channels and the diversity of methods that can be used in order to open a covert channel in computer systems. Considering the stealthiness property of covert channels, in this chapter we identify two main categories in designing covert channels. The first group involves covert channels that achieve undetectability based on the concept of *security through obscurity*. In other words, the covert channel is hidden because the shared resource between the covert transmitter and the covert receiver (e.g., the status of a file on disk) is not being monitored or controlled by access control authority of the system. The file share covert channel, or network covert channels that use reserved bits of header fields of data packets are examples of these channels. However, once the nature of the shared resource is identified, it is a matter of simple pattern analysis techniques to uncover the existence of the covert channel or at least control the channel.

The second category of covert channels are designed based on the assumption that the adversary (e.g., the system observer) is completely aware of the covert communication scheme. In other words, the shared resource between the covert transmitter and the covert receiver, and also the exploitation method of the resource is completely known to the adversarial entity. The key in designing such channels relies on a shared secret between the covert transmitter and the covert receiver. The shared secret defines how the covert

transmitter modulates covert information over the channel, and how the receiver decodes the covert message. However, it is shown that these channels are vulnerable to statistical analysis detection methods, since modulating a covert message could drastically change statistical characteristics of the covert channel traffic. For instance, the on/off strategy in designing network timing covert channels [15] forces the pattern of inter-packet delays of the covert traffic to deviate significantly from the normal behavior of the legitimate traffic of the network. Such clear behavioral mismatch could be easily picked up by a system observer in order to uncover the covert channel. Knowing the structure of the covert channel and the shared medium also enable active adversarial entities to disrupt the covert channel by directing noise into the channel. In other words, one can reduce the available bandwidth of the covert channel by jamming the shared resource such that the channel becomes practically unusable due to its limited capacity.

A majority of timing covert channels over communication networks, including the covert channels that are discussed in this thesis, fall in the second category as the shared resource (i.e., the communication network) is known to be easily accessible to adversarial entities. Thus, in order to remain hidden the covert channel should be designed such that the behavioral characteristics of the covert transmitter, and the statistical attributes of the traffic that carries the covert message resemble the legitimate elements of the same network. In brief, we have identified the following challenges in designing a covert channel in communication networks.

1. *The identity of the covert transmitter, the privacy of the covert receiver, and the stealthiness of the channel should be protected against passive and active detection schemes, given the assumption that the adversarial entity is completely aware of the covert communication scheme.*

2. *The covert channel should be designed such that it can tolerate network noise and adversarial disruptions, while it provides high covert communication rate.*

3. *The shared resource that is used for covert communication has to be selected such that the access to the resource can not be eliminated without drastic measures to the system performance.*

As an example, consider storage network covert channels in which the unused header bits in data packets are used as the shared resource for covert communication. In fact, one can disrupt the normal operation of such covert channels by simply reset the unused portion of packet headers into a predefined value (e.g., reset all bits to zero). This action practically reduces the achievable bandwidth of the covert channel to zero and eliminates the channel form the system. It is noted that the aforementioned elimination process does not have any effect on the overall network performance. In addition, as the covert transmitter constantly changes the unused portion of the header fields, its network traffic exhibits different characteristics as compared to a normal traffic flow in the network. Thus, a simple pattern analysis attack could reveal the existence of the channel and compromise the privacy of the covert transmitter.

3.1 Behavioral Mimicry Covert Communication

To address the aforementioned design requirements, in this thesis we explore different ways of exploiting system resources according to the concept of behavioral mimicry covert communication. In principle, the behavioral mimicry covert communication is based on adopting one or more behavioral characteristics of normal elements in a communication network (e.g., packet transmission pattern, inter-packet delay, payload size, CRC calculation, *etc*), and use them in order to modulate a covert message and create a covert channel. It is noted that the behavioral fingerprints that are used for covert communication should be selected such that the covert communication scheme binds to a normal communication process of the network (e.g., access control, packet generation, acknowledgment, or error detection processes). In this way, the covert channel can not be removed unless drastic measures are taken in removing/controlling the related services in the network.

In addition to binding the covert channel to the critical elements of the communication process, consideration should be made of the exploitation method that is used at the covert transmitter. In fact, modulating the covert message over behavioral characteristics of the covert transmitter should not force the transmitter to exhibit abnormal behaviors as

compared to a normal transmitter in the network. Any irregularity in the behavioral fingerprints of the covert transmitter may lead to detection of the covert channel by monitoring entities in the channel.

The challenge in using behavioral attributes of the covert transmitter for covert communication lies in the unpredictable nature of communication networks (e.g., Internet). On the one hand, the covert transmitter is bound to transmit a particular covert message that has to be decoded with no ambiguity at the covert receiver. On the other hand, the transmitter has to react to unpredictable network events (e.g., packet loss, collision, or network congestion) that demand specific responses according to the communication protocol (e.g., CSMA, TCP). It is noted that while the former constraint forces the covert transmitter to have a regular and controlled behavior (i.e., to represent the covert message at the covert receiver), the latter case involves reaction to random events that may happen at any time. Thus, one can view the design of a network covert channel as *the challenge of finding the correct balance between the controlled nature of the covert message modulation process, and the uncertainty in the forthcoming actions that are expected from the covert transmitter as a part of the communication network.*

3.2 Randomness in Communication Protocols

The behavioral characteristics of a normal node in a communication network are influenced by the events that require direct responses from the elements of the network. For instance, a collision in a WLAN network can happen at any time when two or more transmitters decide to send their packets simultaneously. The reaction to this event is to expand the size of the contention resolution window and wait for a random amount of time before retransmitting the packet. The unpredictable nature of network events, forces the transmitter to exhibit random and unforeseeable behavioral characteristics.

Randomness can also be found in design of communication protocols in order to achieve fairness, stability, and scalability. This alternative source of randomness can be exploited for covert communication, in particular to modulate the covert message over the behavioral

characteristics of the covert transmitter. To this end, the covert transmitter combines the covert message with the output of a random generating function (e.g., a pseudo-random number generator) that can be traced by the covert receiver seamlessly. The result of this combination replaces the original random source and takes its role in controlling the behaviors of the covert transmitter according to the communication protocol. Thus, the modulating process at the covert transmitter is transformed from a rigid and predictable procedure into a process that contributes a considerable amount of randomness to the behaviors of the covert transmitter. On the other end of the channel, the covert receiver removes the effect of random function from the observed attributes of the covert transmitter and recovers the covert message.

The first step in finding a proper shared resource for covert communication is to identify sources of randomness in the target system. Then, the modulation process of the covert message is combined with the communication protocol such that it inherits the randomness that is involved in the normal behaviors of a network transmitter. Binding the covert modulation process with inherent randomness in the communication protocol gives the covert transmitter enough degrees of freedom to respond to network events similarly to a normal transmitter in the network, while it maintains a covert channel with the covert receiver through its behavioral fingerprints in the network. Furthermore, as the covert modulation process has been integrated with the normal communication process in the network, eliminating the covert channel would likely require major modifications in the communication protocol (e.g., removing randomness) which in turn imposes a significant performance degradation or complexity escalation to the communication protocol.

3.2.1 Sources of Randomness in Communication Systems

As discussed before, the first step in the behavioral mimicry covert communication design methodology is to discover a proper source of randomness that can be exploited by the covert modulation process at the covert transmitter. Here, we provide a few examples of such random sources that we have identified in computer and communication systems.

Resource Sharing Algorithms: In communication networks, in particular those

that involve a broadcast communication medium (e.g., wireless channel), the communication resources have to be distributed to all users of the channel in a fair and efficient manner. To this end, resource sharing algorithms (e.g., carrier sense multiple access) often rely on randomness in order to provide fair resource allocation to all users of the system. In more detail, to prevent frequent collision among users of the network, the access control mechanism requires each node to schedule its channel access event for a random time slot in the future. Thus, it is less likely for two nodes to select the same time slot to access the channel and cause a collision in the network. The advantage of such a strategy is its ease of implementation and scalability which is of a great importance in distributed systems. Without such randomness, the performance of the system is reduced dramatically and a drastic increase in complexity and communication overhead could occur.

The same random behavior of resource sharing algorithms can be used to open a covert channel between a covert transmitter and a covert receiver. In Chapter 4 a detailed description of a wireless covert channel is presented in which the exploited randomness is inherited from carrier sense multiple access protocol (i.e., CSMA). The design also includes a set of design criteria for the covert transmitter that is capable of modulating a covert message in its channel access pattern, yet it exhibits a similar packet transmission pattern as compared to a normal user of the network.

Initial Sequence Number: In order to provide reliable and in order delivery of data packets in communication networks, in some implementations of network communication protocols (e.g., TCP) each packet carries a sequence number field that denotes the relative position of information bits that are in the packet as compared to other packets in the flow. In this way, the receiver can identify lost packets and request retransmission from the transmitter. Moreover, the receiver can use sequence number information to reorder the received packets. It is noted that data packets may be delivered out of order due to different paths that each packet may have taken to reach the destination.

The transport communication protocol (i.e., TCP) is an example of such communication protocols that incorporate sequence number fields for reliable delivery of data packets over communication networks. The TCP sequence number is usually initialized with a

random number at the start of a new connection. Thus, each connection can be identified from previous reincarnations of a similar communication channel between the transmitter and the receiver. In this way, one can take advantage of the random behavior of the initial sequence number (i.e., ISN) and exploit such source of randomness for covert communication over computer networks. This method has been explored by Rutkowska and Murdoch in order to design storage network covert channels [20, 21].

Cryptographic Algorithms: *Pseudo-random number generators* (i.e., PRNG) are one of the major components of cryptographic algorithms. These sources of randomness are often used to increase the uncertainty of the attacker in guessing the correct parameter (e.g., password) to violate the security of the cryptographic algorithm. Random numbers are utilized in key exchange protocols (e.g., Diffie-Helman algorithm [88]) as the contribution of each party in the final session key. In symmetric encryption schemes random numbers are used both as keys and also as initial vectors in block cipher modes. The so called *salt* parameter in crypto-systems is basically a random number that is combined with user passwords before the hash of the password is saved in the database. The salt parameter is used in many operating system (e.g., Linux) to prevent dictionary attacks on password databases in computer systems. In general, any challenge and response protocol that is used in cryptographic algorithms (e.g., Key exchange, authentication, zero knowledge proof) relies on PRNGs for generating random numbers. Digital signature schemes (e.g., DSA) also depend on random number generators to increase the entropy of the generated signatures and protect the private key of the signing entity. Without such random parameters, the adversary can glean information about the private key that is used in signing process and compromise the security of the signature scheme.

The output of cryptographic functions (e.g., encryption) can also be considered as a source of randomness in computer systems. An encryption scheme is designed such that its output (i.e., *cipher text*) does not reveal any information about the original *plain text* unless the encryption key is available. One way functions (e.g., cryptographic hash) are another frequently used functions in cryptographic algorithms. These functions are designed such that the output of the function (e.g., hash digest) can not be associated to any arbitrary text message with non-negligible probability. Thus, one can consider the

result of a cryptographically secure hash functions (e.g., SHA-256) or an encryption scheme (e.g., AES) as a random sequence of bits until more information about the cryptographic process (e.g., original message, or encryption key) is disclosed.

As an example, consider a transmitter and a receiver that use a keyed hash function as message authentication code (i.e., HMAC) to provide integrity and authenticity assurance for the transmitted message. In this way, the transmitter sends the hash digest of the original message along with the message to the receiver. The receiver verifies the message as it possesses the key that is used in the authentication process. However, without access to the key, the HMAC portion of the message seems as a random sequence of bits. This communication process can be simply modified to open a covert channel between the transmitter and the receiver. To this end, the covert transmitter appends the matching hash digest to its packets in order to modulate the covert-bit 1, and it sends a packet with a random HMAC to represent covert-bit 0. Since the monitoring entity has no way of verifying the hash digests without the key, the covert channel is undetectable as long as the hash function is secure and the shared secret is not exposed to any unauthorized party.

To thwart such covert channels, access control mechanisms may forbid the use of keyed hash functions for message authentication, or prevent strong encryption schemes for data communication. In other words, the monitoring authority demands to be able to verify the authenticity of the cryptographic algorithm at all times. In the following section we describe a covert communication scheme that exploits randomness in the digital signature algorithm (i.e., DSA) [89]. In fact, it can be shown that while the signature scheme can still be verified by any monitoring entity, the covert channel between the covert transmitter and the covert receiver remains undetected.

3.2.2 Exploiting Randomness

Once the proper source of randomness is identified, it is the matter of designing an exploitation method that combines the random behavior with covert modulation process. In this section, we provide an example of such covert channel design process according to the method by Simmons [12], where the target system is the digital signature algorithm (i.e.,

DSA). The digital signature algorithm is a signature scheme based on asymmetric cryptography, that is used in order to verify the authenticity of a digital message. By verifying the signature on a particular message, one can assure that the message is created by a known sender in which he can not deny signing the message. Moreover, the receiver can verify that the message has not been tampered with during the transmission. The digital signature algorithm (i.e., DSA) is based on the difficulty of the discrete logarithm problem with respect to a primitive element in a finite field. In the interest of both brevity and completeness, we briefly describe the essential elements of the signature scheme.

1. *Public elements of the signature scheme:* These elements are publicly known and are defined as follows:

Signature Scheme Public Elements

- A cryptographically secure hash function $\mathcal{H}(.)$ (e.g., SHA-256).
- A prime number q with length N-bits, where N is at most equal to the size of the hash digest.
- An L-bit prime p such that q divides $p-1$.
- A base generator element $g > 1$, such that $g \equiv \rho^{\frac{p-1}{q}} \mod p$, where $1 < \rho < p-1$.

2. *Key generation:* Each transmitter has a unique pair of public and private keys. The private key (i.e., x) is kept secret, while the public key (i.e., u) is advertised along with the public elements of the signature scheme.

Key Generation

- The private key x is chosen according to a random process, where $1 < x < q$.
- The public key is calculated as $u \equiv g^x \mod p$.

To generate the signature on the message m, the transmitter performs as follows:

Signature Generation

- Calculate the hash digest of the message m (i.e., $\mathcal{H}(m)$).
- Choose a random number k and calculate $r \equiv (g^k \mod p) \mod q$, where $0 < k < q$.
- Find s such that $s \equiv k^{-1}[\mathcal{H}(m) + xr] \mod q$.

The pair (s, r) forms the signature on m. To verify the signature, one can execute the following steps:

Signature Verification

- The verifying process receives the original message m, the signature (r, s), the public elements of the digital signature scheme (i.e., p, q, g), and the public key of the transmitter (i.e., u).
- Let $\omega \equiv s^{-1} \mod q$.
- Let $v_1 \equiv \omega\mathcal{H}(m) \mod q$.
- Let $v_2 \equiv r\omega \mod q$.
- Let $w \equiv g^{v_1}u^{v_2} \mod q$.
- The signature is valid if and only if $r = w$.

It is worth noting the effect of random parameter k in the security of the signature scheme. In fact, it can be easily shown that if two distinct messages (i.e., m_1, m_2) are signed with the same private key (i.e., x) and no additional randomness in the signature scheme, one can solve a system of linear equations to get the private key from the generated

signatures and the hash digests of the original messages that are signed. This requirement for additional random parameter can be used as a source of randomness to create a covert channel in the system.

To this end, in addition to the public elements of the signature scheme and the public key, the covert transmitter shares x with the covert receiver as well. In this way, when the transmitter sends the signature of the message m through the overt channel (i.e., the channel that is observable by monitoring entity), it can also modulate a covert message (i.e., m_c) that is only visible to the covert receiver. Meanwhile, any party with access to the public elements of the signature scheme can still verify the correctness of the digital signature on the original message (i.e., m). To modulate the covert message, the covert transmitter modifies the signing process. It is noted that that the entropy of the covert message is assumed to be at least equal to the entropy and the size of the random parameter k. In case m_c does not have the same entropy as the random parameter k, one can use a random interleaver, or a symmetric encryption scheme (e.g., AES) in order to encode the covert message into a bit-stream with higher entropy factor. The modified signature scheme at the covert transmitter can be illustrated as follows:

Signature Generation and Covert Message Modulation

- Given the covert message m_c, the overt message m, and public elements of the signature scheme.
- Let $r \equiv (g^{m_c} \mod p) \mod q$, where $0 < m_c < q$.
- Find s such that $s \equiv m_c^{-1}[\mathcal{H}(m) + xr] \mod q$.

The pair (s, r) is the signature on the overt message m and can be verified using the same verification process as the one that is used in the original signature scheme. On the other hand, the generated signature at the covert transmitter contains an embedded covert message that can be decoded at the covert receiver as follows:

Decoding Covert Message

- Given $x, m, (r, s)$, the covert receiver recovers the covert message m_c as follows:

$$m_c = \left([\mathcal{H}(m) + xr]^{-1}\right)^{-1} \mod q.$$

By combining the covert modulation process with the inherent randomness of the original signature scheme, the covert transmitter is capable of generating a signature that is verifiable by any node in the network, given the verification process has access to the public elements of the signature scheme. Meanwhile, the generated signature carries an embedded covert message that is visible only to the covert receiver. In other words, modulating the covert message does not change the behaviors of the signature scheme from a third party perspective. Furthermore, due to the vital role of the exploited randomness in the security of the signature scheme, the access control entity can not remove the random parameter from the signature scheme in an attempt to eliminate the covert channel. The expected covert rate of the presented covert channel is equal to $\log_2(\max(k)) = \log_2(q-1)$ bits per signature.

In the following chapters, we demonstrate how to exploit different sources of randomness in communication networks in order to design robust and undetectable network covert channels. First, let's define the system model and the elements of the communication network that are used in this dissertation.

3.3 System Model

3.3.1 Channel Model

A covert communication scheme consists of a covert transmitter, a covert receiver, and a shared medium that is used for covert communication in the system. An *overt channel* is defined to be an authorized communication channel in the network that may contain an

embedded covert message. The term *covert channel* refers to an overt channel that has an embedded covert message in the channel traffic. In contrast, the *legitimate channel* is an overt channel that carries no covert message. The objective of the covert transmitter is to modulate the covert message and at the same time mimic the behavioral characteristics of a normal node in the system. In this thesis, we are focused on designing *active covert channels* over communication networks. In an active covert channel the covert transmitter is actively involved in generating the traffic of the covert channel (i.e., *covert traffic*), with respect to the covert message it intends to send to the covert receiver. On the other hand, in a *passive covert channel* the covert transmitter intercepts a legitimate traffic flow and modifies the traffic in order to modulate the covert message. The covert receiver is assumed to be aware of the identity of the covert transmitter, the covert communication scheme, and it shares a secret with the covert transmitter. It is noted that the terms covert receiver and receiver and also covert transmitter and transmitter are used interchangeably.

The *covert rate* of the covert channel denotes the rate in which covert information is exchanged between the covert transmitter and the covert receiver. Also, the *robustness* of the covert channel is primarily defined based on the error-rate of the covert receiver in decoding the transmitted covert information. Through the rest of this dissertation the following notations are used:

- The term $[n]$ is used to represent the set of all positive integers smaller or equal than n (i.e., $[n] = \{1, 2, ..., n\}$).
- A sequence of elements is illustrated with letters in bold case. Hence, $\mathbf{v} = (v_1, v_2, ...v_n)$ depicts a sequence of n numbers, where $v_i,\ i \in [n]$, is an element of the sequence $\mathbf{v}$.
- Given a sequence of n elements (i.e., $\mathbf{v}$), the term $\mathbf{v}_i^j,\ i \leq j \leq n$,represents the subset of $\mathbf{v}$ that contains elements from index i to j (i.e., $\mathbf{v}_i^j = (v_i, v_{i+1}, ..., v_j)$).
- The term $\mathbf{v} \sim \mathcal{D}$, expresses that the elements of $\mathbf{v}$ are sampled according to the probability distribution function $\mathcal{D}$.

3.3.2 Adversary Model

The adversarial entity is a network node that can monitor the overt traffic of the network, including the traffic from the covert transmitter and the covert receiver. The adversarial entity is assumed to be aware of the covert communication protocol, and it has access to samples of the legitimate traffic and the covert traffic. The primary goal of the adversary is to detect the covert channel and distinguish the traffic of the covert transmitter from a legitimate traffic flow in the network (i.e., passive adversarial attack). In addition, the adversary is assumed to be capable of actively disrupting the communication channel by introducing random noise into the covert channel (i.e., active adversarial attack). For instance, the adversary can deliberately increase the packet delay or drop some of the packets from the traffic flow. The objective of an active attack is to limit the effect of the covert communication scheme by reducing the capacity of the covert channel.

However, the adversary does not have any a priori knowledge about the existence of the covert channel in the system, or the identity of the network nodes that may be involved in the covert communication process. Furthermore, the shared secret between the covert transmitter and the covert receiver, and any information that may be derived from the secret key, is assumed to be unknown to the adversarial entity in the system.

3.3.3 Undetectability

The undetectability of a covert channel is defined according to the ability of the adversarial entity to distinguish the covert traffic from the legitimate traffic in the network. In other words, the more the covert transmitter behaves like a legitimate node in the network, it is less likely that the covert channel be discovered and the covert message becomes exposed. Probabilistic statistical tests provide a set of reliable metrics in order to study the stealthiness property of covert channels. In this way, one can compare the statistical characteristics of the covert traffic with the same attributes of the legitimate traffic of the network, and look for abnormalities that may be the result of embedding a covert message. A probabilistic statistical test analyzes a set of traffic samples and determines a unique

identifier that fits the statistical characteristics of the input sequence. Since in analyzing covert channels often large volume of samples are used as the input to the test functions, in this dissertation only a subset of statistical tests that have polynomial-time complexity are considered. The following definition adequately describes the statistical tests that are used in this thesis for the purpose of detecting covert channel.

Definition 3.1 (Probabilistic statistical test)**.** *A probabilistic statistical test is a polynomial-time function (i.e.,* $\mathcal{T}(.) : \mathbb{R}^* \rightarrow [0,1]$*) that accepts an arbitrary number of input elements and assigns a real number in the range* $[0,1]$ *to the input set.*

It is worth noting that this definition easily covers the covert channel detection tests that are mentioned in Section 2.3.3 (e.g., KL-test, regularity test). In fact, one can consider the result of the statistical test $\mathcal{T}(.)$ as the probability that the detection scheme marks the sampled traffic as a covert traffic. In addition, one can define an extreme case of undetectability in which the covert channel can not be detected by any polynomial-time statistical test.

Definition 3.2 (Polynomial undetectability)**.** *A covert channel is polynomially undetectable with respect to the parameter* κ*, if for any number of sampled traffic from the covert channel (i.e.,* $\mathbf{d}$*) and the traffic from the legitimate channel (i.e.,* $\tilde{\mathbf{d}}$*), given any probabilistic polynomial-time statistical test* $\mathcal{T}$*, there exists a negligible function*[1] $\nu(\kappa)$ *such that:*

$$|\mathcal{T}(\mathbf{d}) - \mathcal{T}(\tilde{\mathbf{d}})| \leq \nu(\kappa) \tag{3.1}$$

As mentioned before, pseudo-random number generators (i.e., PRNG) play a major role in communication systems. In fact, PRNGs are considered one of the main sources of randomness in computer systems. A pseudo-random number generator is a deterministic algorithm that accepts a sequence of truly random bits (i.e., the seed parameter) of length κ (i.e., $\mathcal{K} \in \{0,1\}^\kappa$), and generates a longer sequence of numbers. The function is called *deterministic* since it generates a same sequence of numbers if the seed parameter (i.e., $\mathcal{K}$)

[1] $\nu(\kappa)$ is called *negligible* if for any positive polynomial $p(.)$, and for all sufficiently large κ, $\nu(\kappa) \leq \frac{1}{p(\kappa)}$.

is the same. A cryptographically secure pseudo random number generator (i.e., CSPRNG) is a PRNG in which its output can not be distinguished from a sequence of truly random numbers by any polynomial-time statistical test. By true random sequence we mean a sequence of independent and identically distributed numbers (*i.i.d.*) that are chosen randomly according to a uniform distribution. In a more mathematical form, a CSPRNG can be defined as follows:

Definition 3.3 (CSPRNG)**.** *A deterministic polynomial-time algorithm* $G : \{0,1\}^\kappa \to \mathbb{R}^n$ *is a cryptographically secure pseudo-random number generator, if for all positive integers* $0 < \kappa < n$*, for all (secret) seed sequences* $\mathcal{K} \in \{0,1\}^\kappa$*, and any probabilistic polynomial time statistical test* $\mathcal{T}(.)$*, there exists a negligible function* $\nu(\kappa)$ *such that:*

$$|\mathcal{T}(\mathbf{v}) - \mathcal{T}(\tilde{\mathbf{v}})| < \nu(\kappa) \tag{3.2}$$

Where, $\mathbf{v} = G(\mathcal{K})$*, and* $\tilde{\mathbf{v}}$ *is a sequence of* n *elements generated according to a truly random distribution uniformly distributed over* $\mathbb{R}$*.*

In other words, the output of a *CSPRNG* is indistinguishable, with respect to the parameter κ, from a true sequence of random numbers by any polynomial-time statistical test.

Theorem 3.1. *Let* $\mathcal{F}(.) : \mathbb{R} \to [0,1]$ *be an invertible function (i.e.,* $\mathcal{F}^{-1}(.)$ *exist). Consider* $\mathbf{v}$ *and* $\tilde{\mathbf{v}}$ *to be two sets of (i.i.d.) random numbers in the range* $[0,1]$*, generated by a CSPRNG and a truly random uniform distribution, respectively. Let* $\mathbf{d} = \mathcal{F}^{-1}(\mathbf{v})$*, and* $\tilde{\mathbf{d}} = \mathcal{F}^{-1}(\tilde{\mathbf{v}})$*. Then, for any probabilistic polynomial time statistical test* $\mathcal{T}$*, there exists a negligible function* $\nu(\kappa)$ *such that:*

$$|\mathcal{T}(\mathbf{d}) - \mathcal{T}(\tilde{\mathbf{d}})| = \nu(\kappa) \tag{3.3}$$

Proof. The proof is by contradiction. Assume there exists a polynomial time statistical test (i.e., $\mathcal{T}$) that can tell apart $\mathbf{d}$ and $\tilde{\mathbf{d}}$ with non-negligible probability. Hence, one can define another polynomial time statistical test $\mathcal{T}^* = \mathcal{T} \circ \mathcal{F}^{-1}$, that can distinguish $\mathbf{v}$ and $\tilde{\mathbf{v}}$ accordingly. However, this contradicts with the fact that $\mathbf{v}$ is generated by a CSPRNG according to Definition 3.3. □

3.4 Chapter Summary

In this chapter we presented our design methodology for robust and undetectable covert communication channels in communication networks. The objective of the proposed design methodology is to systematically construct a covert transmitter that is capable of modulating a covert message over its behavioral fingerprints in the network (e.g., packet transmission pattern), yet it exhibits the same behavioral characteristics as compared to a normal node in the system. To this end, we identified the randomness in communication protocols and network environment as the key to find the proper resource that can be exploited as the medium for covert communication. By combining the covert modulation process with the inherent random behavior in communication protocols, the covert transmitter gains enough degrees of freedom to release itself from the constraints that are imposed by the existence of a fixed covert message in its behavioral fingerprints. Thus, the covert transmitter is capable of acting like a normal transmitter of the system and the covert traffic resembles the properties of a normal traffic flow in the network. A salient feature of such design methodology is to provide a path for designing covert channels with high levels of stealthiness that is a vital advantage for covert communication over tightly monitored environments like communication networks. To elaborate more on this matter, we presented an example of a covert channel design based on the proposed design methodology.

The chapter also includes the definitions and the notations that are used in the rest of this thesis. A detailed description of the system model that is studied in the rest of this dissertation is presented, including the parameters of the overt channel, the normal traffic of the network, and the ordinary transmitters in the system. The adversary model and the attack scenarios are also described in detail. Furthermore, a formal definition on the notion of undetectability is illustrated that is used to evaluate the stateliness of the proposed covert communication schemes in the following chapters.

Chapter 4

Wireless Covert Communication

4.1 Introduction

Wireless channels offer unique and peerless properties for modern communication networks. The wireless medium can reach beyond physical boundaries of the surrounding environment, and serve any user that is in the range of the transmitter's radio. Such unique characteristics enable wireless devices to play major roles in mission critical applications (e.g., health devices), and distributed systems (e.g., sensor networks). However, the heterogeneous structure of wireless networks, the limitations of wireless devices, and the diversity of wireless services make it a challenging task to provide security and privacy to wireless users. As an example, a majority of mobile wireless devices have limited processing and power resources. Thus, it is extremely difficult to rely merely on cryptographic primitives to achieve confidentiality and privacy in wireless networks.

Covert channels provide an alternative solution to improve the security of wireless devices by hiding the communication channel from adversarial entities. The wireless medium is broadcast in nature which means all activities over the wireless channel can be observed by other users in the network. The channel can be easily modified just by sending a signal over the channel, and such modification can be detected by any receiver in the range of the transmitted signal. Therefore, one can observe that wireless networks satisfy all conditions

for existence of a covert channel according to the Kemmerer's principle. Resource sharing is also a fundamental part of most wireless communication architectures. The adaptive nature of wireless medium necessitates the existence of rather complex and flexible channel access mechanisms in order to serve an ever changing number of users in the system. Thus, behavioral mimicry covert communication can be applied to design undetectable covert channels in the wireless environment.

Inspired by this observation, in this chapter a new covert communication scheme is presented which is based on mimicking the structural behaviors of multiple access protocols in wireless networks. In particular, the proposed covert channel targets the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) algorithm [90], which is used in popular wireless architectures such as Wireless LAN, Bluetooth, and Zigbee. In brief, the proposed covert channel is based on adopting the structure of the CSMA/CA algorithm that controls the packet transmission process of ordinary users of the system. In this way, the channel access mechanism of the covert transmitter is modified in such a way that gives the covert transmitter enough freedom to embed the covert message into its overt traffic. However, the modification is done under the constraint that the packet transmission pattern of the covert transmitter remain the same as compared to ordinary users of the network. The key element in this design is to use the randomness that is brought into the system by the multiple access protocol in order to provide each user with a fair share of network resources (i.e., channel bandwidth). By adopting the structure of the communication protocol, one can exploit the aforementioned random behavior and establish a hidden communication channel in the system.

4.2 Preliminaries and System Model

4.2.1 Shared Medium and Multiple Access Protocols

Carrier Sense Multiple Access/Collision Avoidance algorithm is one of the most common multiple access protocols that is used in modern communication networks specifically in

the wireless environment. IEEE 802.11 [91] (i.e., WLAN) is an example of wireless communication standards that uses the CSMA/CA algorithm to control the access to the wireless medium. Due to the popularity of IEEE 802.11, the proposed covert communication scheme is described according to the WLAN specifications, however our design can be easily extended to any communication model that involves a multiple access protocol and a shared broadcast medium similar to the wireless channel.

In CSMA/CA, the wireless channel is divided into small time periods called time slots. In order to access the wireless channel, network users constantly check the channel and monitor network activities (i.e., packet transmission) of other users of the wireless channel. If the channel is busy (i.e., a user is transmitting information over the channel), other users postpone packet transmission and randomly select a backoff time (measured in slot times) in the interval [0, W), where W is the size of the contention window. The backoff timer is decreased any time that the channel is sensed idle for a specific period of time called distributed inter-frame space (i.e., DIFS). The timer stops if the channel gets busy again (i.e., another node uses the channel for transmission). When the backoff timer reached zero, the user transmits its packet and the receiver acknowledges the packet, if the transmission was successful, after a predefined period of time called short inter-frame space (i.e., SIFS).

The size of the contention window (i.e., W) is initialized at W_{min} and it is doubled after each unsuccessful transmission attempt. The expansion process continues until the size of the contention window reaches W_{max}. Then, the window remains unchanged until the transmission is successful or the number of retransmission attempts reaches a predefined number. In either case, the size of the contention window is reset to W_{min} for the next packet transmission round. This mechanism is called exponential binary backoff algorithm which is used in IEEE 802.11 communication protocol. Figure 4.1 depicts the structure of CSMA/CA and the binary back off mechanism according to the specifications in IEEE 802.11 standard.

An important feature of the CSMA/CA algorithm is the randomness that is involved in the binary backoff algorithm. Suppose all the nodes that share the wireless channel select a predetermined time slot. Since all of them monitor the same channel, their backoff

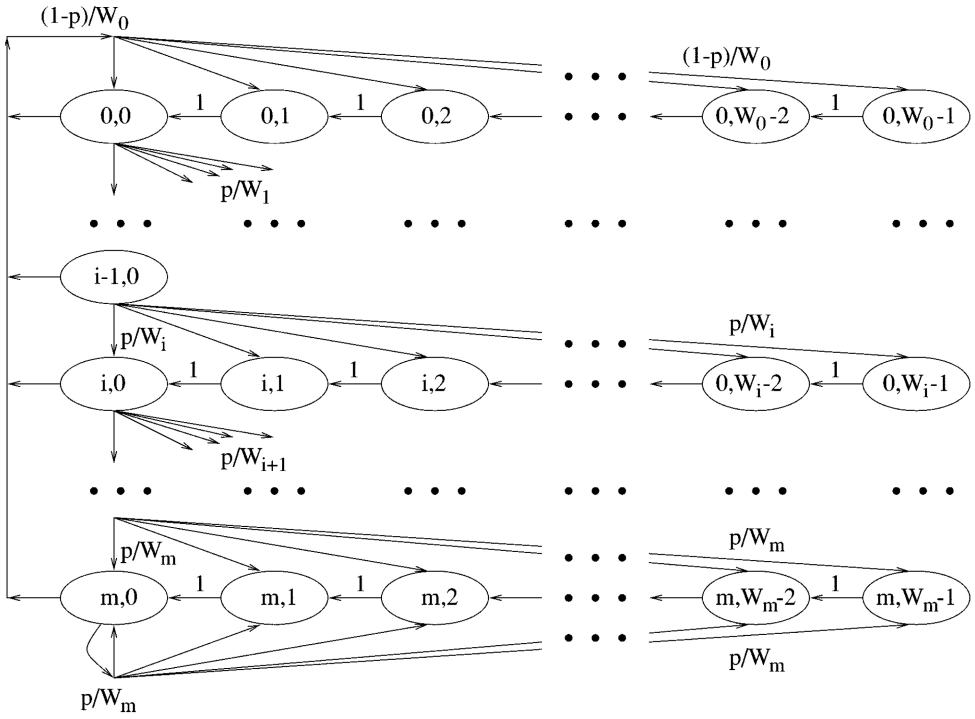


Figure 4.1: The CSMA/CA and the binary backoff algorithm. In each stage W_i is the size of the contention window, where $W_i = 2^i W_{min}$ and p is the probability of unsuccessful transmission attempt [2].

timers would reach zero simultaneously and a collision is inevitable. However, with random selection of the backoff time, it is possible that only one node attempts to access the channel in each particular time slot. Hence, the probability of a successful packet transmission is improved significantly. In what follows we demonstrate how to exploit such random characteristic of the channel access mechanism in order to create an undetectable wireless covert channel.

4.2.2 System Model

The proposed wireless covert channel is designed based on the IEEE 802.11 (i.e., WLAN) environment, where a group of nodes (including the covert transmitter and the covert

receiver) share a wireless channel using the CSMA/CA and the binary backoff algorithms. The network parameters (e.g., SIFS, DIFS, slot size, ...) are assumed to be known at the covert transmitter and the covert receiver. All users in the network are assumed to be backlogged, and always ready to send packets through the channel (i.e., saturated network). This assumption makes the communication model tractable in order to facilitate the analytic study of the covert communication scheme. However, due to the adaptive nature of the proposed scheme, relaxing this assumption does not affect the ability of the covert transmitter and the covert receiver to establish a covert channel in the system. It is also assumed that each packet contains the identity of its transmitter (e.g., the source address field in the IEEE 802.11 frame).

4.3 Fixed Rate Covert Channel (FRCC)

The FRCC scheme is basically a timing covert channel in which covert information is modulated over packet transmission pattern of the covert transmitter. It is noted that the normal operation of the CSMA/CA algorithm depends on the binary backoff algorithm according to the discussion in Section 4.2.1. The key element in this process is the backoff timer that defines the wait times for each node before it tries to access the channel. By adopting this timing process, it is possible to design a timing covert channel that allows the covert transmitter and the covert receiver to communicate without being detected.

4.3.1 Covert Clock

Synchronization is one of the main building blocks of a timing covert channel according to the Kemmerer's principle [14]. In timing covert channels, synchronization can be achieved using a reference clock that is accessible to the covert transmitter and the covert receiver. It is noted that in addition to the normal system clock, any sequence of events can establish a reference clock [22]. In this way, each event is considered as one clock tick that increments the clock by one unit.

The *covert clock* is the reference clock of the FRCC scheme that uses the activities (i.e., packet transmission) of a subset of nodes in the network (i.e., the *covert set*) as the clock ticks. Hence, every time a member of the covert set transmits a packet, the covert clock is advanced by one unit. It is noted that due to the broadcast nature of the wireless environment, all users that share the same channel can overhear the transmitted packets. Therefore, as long as the covert transmitter and the covert receiver track the same covert set, their covert clock would remain synchronized.

The covert set can be preset in the covert transmitter and the covert receiver, or it may be generated during the channel operation. For instance, the covert set can be defined as the set of users that their identity (e.g., MAC address) satisfies a specific condition. As another example, a user can be included in the covert set based on its location, or signal strength. It is worth noting that the covert communication scheme should be equipped with proper error control mechanisms in order to deal with possible mismatches in the covert clocks of the covert transmitter and the covert receiver (e.g., in case of packet loss on one end of the covert channel). The analysis of the effect of such error events on the performance of the covert channel will be covered in Section 4.5.

4.3.2 FRCC Channel Design

Let S be the subset of network users in which their packet transmissions are considered as clock ticks for the covert clock (i.e., covert set). Let C_t, and C_r represent the covert clocks at the covert transmitter and the covert receiver, respectively. The covert transmitter of the FRCC scheme uses a channel access mechanism that is very similar to the traditional CSMA/CA and binary backoff algorithms. However, the clock that controls the binary backoff algorithm at the covert transmitter is replaced by the covert clock (i.e., C_t). In other words, the covert transmitter measures its waiting time before each packet transmission attempt according to the value of its covert clock instead of the actual reference clock of the system. The selection of the backoff time at each stage is also modified to be controlled by the covert message. The combination of these two modifications enables the covert transmitter to modulate covert information over its packet transmission pattern.

Figure 4.2 depicts the structure of the covert transmitter contention resolution mechanism that is modified in order to facilitate the FRCC scheme. It also illustrates how the transmitter embeds the covert message into its transmission window that is controlled by the covert clock. It is worth noting the similarity between the structure of the contention resolution algorithm at the covert transmitter (i.e., Figure 4.2) and the traditional binary backoff algorithm that is shown in Figure 4.1. In more detail, both methods incorporate a number of *stages* where each stage consists of a sequence of serially connected *states*. The transmitter selects one state in the first stage (randomly or according to the covert message) and follows the algorithm flow until the packet is successfully transmitted. The only difference between these two contention resolution algorithms lays in the event that triggers the move from one state to the next one. In other words, while in the CSMA/CA algorithm the state transitions are controlled by observing the channel to be idle for a known period of time (i.e., DIFS), the state of the algorithm in the covert transmitter is changed when a packet transmission by members of the covert set is detected.

Covert Message Modulation

Each covert communication block starts with a successful packet transmission by the covert transmitter. The covert transmitter then loads the next covert message and resets its covert clock (i.e., $C_t = 0$). Let ω be the covert message from the alphabet set Ω. Each covert message is associated with a unique state in the first stage of the transmitter's transmission window . Hence, the size of the alphabet set is equal to the initial size of the transmitter's transmission window (i.e., $|\Omega| = T_0$). For simplicity and without loss of generality, let's assume $\Omega = \{0, 1, ..., T_0 - 1\}$. Thus, the message $\omega \in \Omega$ corresponds to the state m_ω^0 in Figure 4.2. It is noted that for any alphabet set Γ of size T_0, one can find a one-to-one mapping that transforms Γ to Ω.

Given the covert message ω, the covert transmitter moves to the state m_ω^0 and begins to monitor the communication channel to catch successful packet transmission by members of the covert set (P_S is the probability of a successful transmission by members of S). For each packet, the transmitter's covert clock is incremented by one unit and the contention

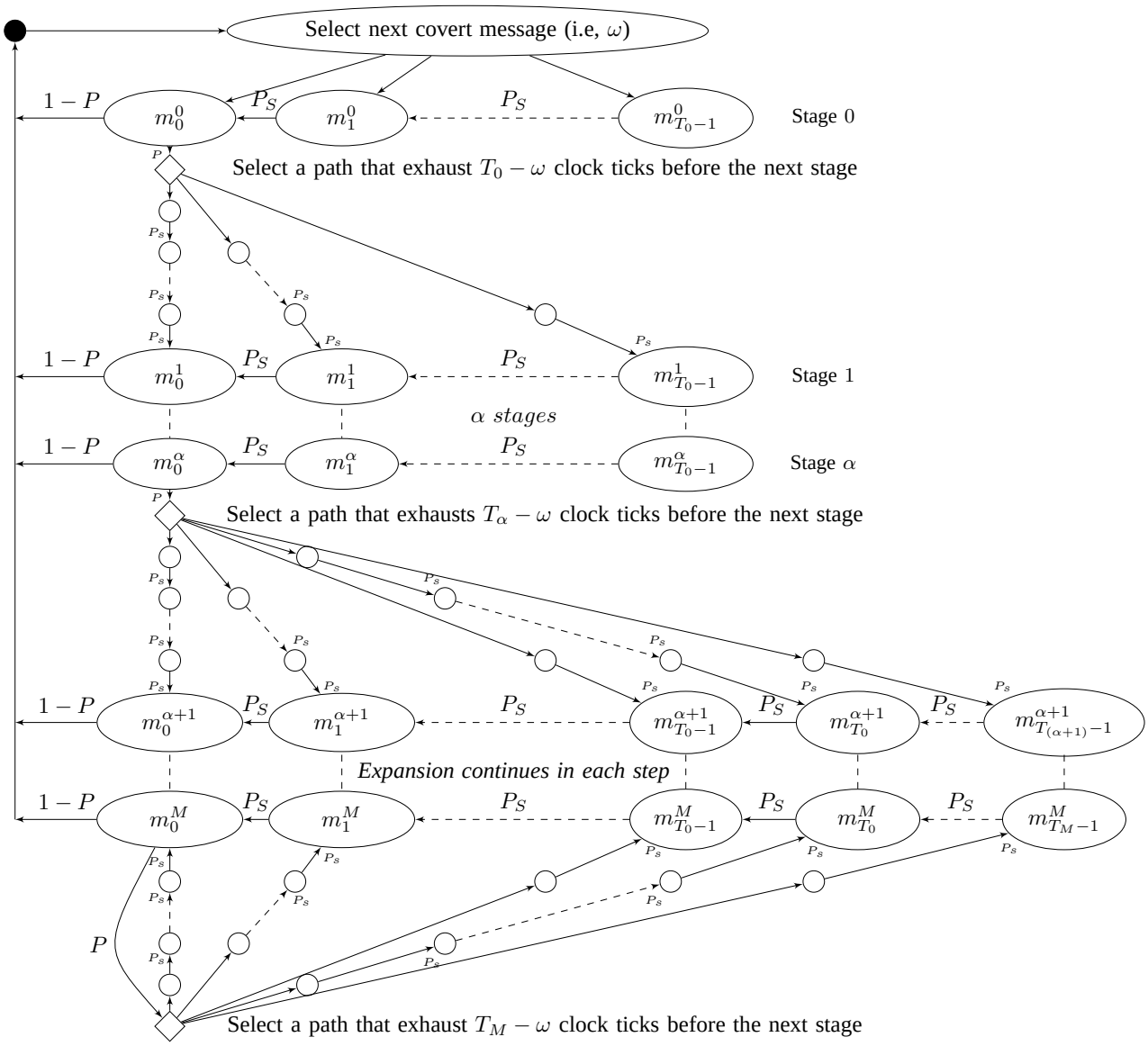


Figure 4.2: Covert message transmission. P_S is the probability of a successful transmission by members of the covert set.

resolution algorithm moves towards the last state of the current stage of the transmission window (i.e., one state to the left in Figure 4.2). It is noted that the covert transmitter can recognize multiple transmissions of a single packet using the packet's sequence number field. When the transmitter reaches the last state (i.e., m_0^0), the covert transmitter sends its next packet in order to mark the value of its covert clock which represents the covert message.

For instance, suppose $T_0 = 4$, and the transmitter intends to transmit the binary sequence $\mathcal{B} = \{01110011\}$ over the covert channel. The transmitter generates the alphabet set $\Gamma = \{00, 01, 10, 11\}$ and splits $\mathcal{B}$ into smaller items of size two, where each item is a member of Γ. It is noted that $|\Gamma| = T_0$, and the binary to decimal conversion is a mapping that transforms any elements in Γ to a unique element in $\Omega = \{1, 2, 3, 4\}$. Hence, the covert message is mapped to the corresponding states in the contention resolution algorithm as follows:

$\mathcal{B} = \{01110011\}$	Covert information ready to be send at the covert transmitter.
$\Rightarrow \{01, 11, 00, 11\}$	Splitting into elements of size $\lceil \log_2(T_0) \rceil = 2$.
$\Rightarrow \{1, 3, 0, 3\}$	Mapping into the corresponding elements in Ω.
$\Rightarrow \{m_1^0, m_3^0, m_0^0, m_3^0\}$	Mapping into the corresponding states at stage $i = 0$.

In fact, one can recognize the similarity of the above example to the concept of modulation in digital communication [92]. In this way, the covert transmitter selects state m_1^0 for the first packet transmission. Similarly, the next packet transmission will begin by selecting state m_3^0 that correspond to the next covert message in $\mathcal{B}$.

At the other end of the channel, the covert receiver monitors the channel for successful packet transmission attempts of members of the covert set. Thus, the covert receiver maintains its covert clock similar to the the covert transmitter. In this way, when the covert receiver detects a packet from the covert transmitter, it reads the value of its covert clock (i.e., C_r) and decodes the corresponding covert message. The receiver then resets its clock and monitors the channel in order to receive the next covert message. It is worth noting that this scheme does not affect the usual packet transmission of the covert receiver

or any other node in the system. The covert transmitter also is able to maintain its overt communication with an exception that its contention resolution algorithm is controlled by the covert clock instead of a regular binary backoff algorithm.

Packet Loss and Collision Handling

If the covert transmitter fails to transmit its packet on the proper time slot (e.g., due to collision with other nodes), the covert transmitter expands its contention window and selects another transmission slot to send the packet. The reason for the covert transmitter to expand its contention window is two fold. First of all, it is essential for the covert channel to achieve maximum stealthiness. To this end, the covert transmitter should not behave differently as compared to other nodes in the system. In the CSMA/CA algorithm, each node expands its contention window and waits for a random amount of time before retransmitting its packets. The covert transmitter should not be exempted from this rule, otherwise it would be easy to detect the covert channel. Thus, the size of the covert transmitter's contention window in the i^{th} transmission stage is calculated as follows:

$$T_i = \begin{cases} T_0 & 0 \leq i \leq \alpha \\ 2T_{i-1} & \alpha < i \leq M \\ T_M & i > M \end{cases} \tag{4.1}$$

Where, M is the index of the last stage in which the transmitter expands its contention window. The parameter α is a design parameter of the proposed scheme and will be discussed in Section 4.3.3. Algorithm (1) shows how the covert clock is used in order to transmit a covert message. Here, the $wait()$ function exhausts certain number of clock ticks (i.e., packet transmission by members of the covert set) before it returns the control to the main process. The function $transmit\_pkt()$ transmits the packet through the channel and returns true if the actual receiver of the packet acknowledges the reception of the packet (i.e., successful transmission).

In addition to stealthiness, expanding the covert transmitter's contention window plays a major role in synchronizing end peers of the covert channel. In fact, as the covert message

Algorithm 1 FRCC transmission sequence.

```
/* i is the covert transmitter's current stage index.
function Success = sendpkt(T_i, message)
wait(message);
Success = transmit_pkt();
if Success then
  return true
else
  wait(T_i − message);
  return false
end if
```

is embedded in the value of the covert clock, both nodes require accurate knowledge on the current state of the covert clock if they are to communicate effectively. To this end, following each unsuccessful packet transmission attempt, the covert transmitter waits for exactly $T_i - \omega$ clock ticks before expanding its contention window and moving to the next stage. Hence, the covert clock is equal to $\sum_{j=0}^{i} T_j$ at the beginning of the $(i+1)^{th}$ stage regardless of the value of the covert message. This offset is removed from the value of the receiver's covert clock (i.e., C_r) in order to decode the covert message as follows:

$$\hat{\omega} \equiv C_r \pmod{T_0}. \tag{4.2}$$

Where, $\hat{\omega}$ is the decoded covert message at the covert receiver. It is worth nothing that exhausting $T_0 - \omega$ clock ticks (instead of $T_i - \omega$) at the end of each stage also satisfies the synchronization criterion, however it deviates the covert transmitter's behavior from normal characteristics of an ordinary user in the system. Hence, it is not an option for a stealthy channel design.

As the size of the transmitter's contention window increases, there are more states in each stage that correspond to a particular message. For instance, if $T_i = kT_0$, there exist k states in the stage i that corresponds to the message ω (i.e., m^i_ω, $m^i_{T_0+\omega}$,..., $m^i_{(k-1)T_0+\omega}$). In

Algorithm 2 Covert transmitter function of FRCC.

/* T_0 is the initial size of covert transmitter's contention window*/
/* $PRNG(n)$ generates a random bit stream of length n.
$i = 0;\ \omega_s = \omega;\ T_{-1} = T_0;$
repeat
  if $\alpha < i < M$ **then**
    $b_i = PRNG(i - \alpha);$
    $T_i = 2^{i-\alpha}.T_0;$
    $\omega_s = (b_i||\omega);$
  else
    $T_i = T_{i-1};$
  end if
  $Success = sendpkt(T_i, \omega_s);$
  $i = i + 1;$
until $Success$

the FRCC scheme, the transmitter randomly picks one of the aforementioned states and moves to the next stage. Therefore, despite the increase in the number of possible states in the forthcoming stages, the size of the message set (i.e., Ω) remains constant. In Section 4.4 it is shown that with an adaptive modulation algorithm, one can take advantage of the extra capacity due to expansion of the covert transmitter's contention window, and increase the covert rate of the channel even further. Algorithm (2) depicts the transmitter's function of FRCC. In each stage, b_i depicts a random bit stream of size $i-\alpha$ which is used to selects one of the possible states for the covert message ω in the i^{th} stage of the contention resolution algorithm.

4.3.3 Design Parameters

In this section, different parameters of the proposed covert communication scheme are derived. The idea is to optimize system parameters to achieve maximum stealthiness (i.e.,

similar characteristics as compared to other nodes in the system), and improve the covert rate of the channel. According to Figure 4.2, it can be observed that the covert transmitter's contention resolution algorithm mimics the same principles as the CSMA/CA algorithm. In fact, in both schemes the transmitter has to wait for a specific amount of time before each packet transmission/retransmission attempt. Hence, to harmonize the behavioral fingerprints of these two transmitters (i.e., the covert transmitter and normal CSMA/CA users), it is important to derive the parameters of a regular CSMA/CA transmitter and then adapt the covert transmitter to resemble the same characteristics.

Figure 4.1 depicts the two-dimensional Markov chain model for the binary backoff algorithm which is widely used for performance analysis of the IEEE 802.11 MAC architecture [2]. Each state of this Markov process is represented by an ordered pair $(s(t), b(t))$, where $b(t)$ denotes the state of the backoff timer and $s(t)$ represents the current backoff stage of a given station at time t. It is noted that in this model, t and $t+1$ correspond to two consecutive time slots that may contain a packet transmission or an idle time slot. A transition from one stage to another one happens upon a successful packet transmission (i.e., transition from stage $s(t) = i$ to stage $s(t+1) = 0$), or after a failed packet transmission attempt (i.e., from stage $s(t) = i$ to $s(t+1) = i+1$). The model also depicts how the back off timer is decremented each time the channel is sensed idle for a DIFS period of time (i.e., transition from state $b(t) = j$ to $b(t+1) = j-1$).

Let p_c denote the collision probability in a given time slot. In other words, p_c is the probability that at least one of the remaining nodes in the system simultaneously transmits a packet with the transmitter. Thus,

$$p_c = 1 - (1-q)^{N-1} \tag{4.3}$$

Where, N is the number of users in the network, and q is the steady state probability of a packet transmission attempt at a particular time slot. In practice, the transmitter can learn about the result of its transmission attempt only if it receives the acknowledgment message from the actual recipient of the packet. The receiver however, sends the Ack message only if it receives the packet correctly. Hence, if the packet or the Ack message

are lost due to the wireless channel noise, the transmitter considers the transmission as a failed attempt and prepares for retransmission. Let p_e^{data} and p_e^{ack} denote the frame error probability of the data packet and the acknowledgment message, respectively. It is also assumed that each packet in the channel is subject to an independent error event. Thus, the probability of failure in transmitting a packet due to the wireless channel noise can be written as follows:

$$p_e = p_e^{data} + p_e^{ack} - p_e^{data}.p_e^{ack} \tag{4.4}$$

Combining Equations (4.3), and (4.4) one can write the transmission failure probability at a given time slot (i.e., p) as follows:

$$p = 1 - (1 - p_e)(1 - q)^{N-1} \tag{4.5}$$

Define $b_{i,j} = \lim_{t\to\infty} P(s(t) = i, b(t) = j)$, where $s(t)$ and $b(t)$ represent the stage and the state of the Markov chain at time t, respectively. Since a transmission happens only at state 0 of each stage, one can write the probability of transmitting a packet at a given time slot as follows:

$$q = \sum_{i=0}^{M} b_{i,0} \tag{4.6}$$

Where, M is the maximum number of stages that the binary backoff algorithm is allowed to expand its contention window. It is noted that the transition from state $(i-1, 0)$ to the next retransmission state (i.e., $(i, 0)$) happens only if the former attempt fails. Hence,

$$b_{i,0} = p.b_{i-1,0} \qquad 0 < i < M \tag{4.7}$$

Therefore,

$$b_{i,0} = p^i.b_{0,0} \qquad 0 < i < M \tag{4.8}$$

However, there is another possible branch that can lead to state $(M, 0)$ in the last stage of the contention resolution algorithm. In other words, if the transmission attempt in state

$(M, 0)$ fails, the algorithm remains in stage M as it is not allowed to expand its contention window any further. Thus,

$$\begin{aligned} b_{M,0} &= p.b_{M-1,0} + p.b_{M,0} \\ &= \frac{p.b_{M-1,0}}{1-p} = \frac{p^M}{1-p} b_{0,0} \end{aligned} \tag{4.9}$$

In order to calculate the probability of being in other states of the Markov chain (i.e., $b_{i,j},\ j \in [1, W_i - 1]$), one can observe that $b_{0,0} = (1-p)\sum_{i=0}^{M} b_{i,0}$. In other words, a packet should be successfully transmitted before the algorithm can return to stage 0 and start the transmission sequence for the next packet. Based on this observation, the probability of finding the Markov chain in state $b_{i,j}$ for $j \in [1, w_i - 1]$, and $i \in [0, M]$ can be written as follows:

$$b_{i,j} = \frac{W_i - k}{W_i} \times \begin{cases} (1-p)\sum_{i=0}^{M} b_{i,0} & i = 0 \\ p.b_{i-1,0} & 0 < i < M \\ p.(b_{m-1,0} + b_{m,0}) & i = M \end{cases} \tag{4.10}$$

Thus, given Equations (4.8), (4.9), and (4.10) we have:

$$\begin{aligned} 1 &= \sum_{i=0}^{M} \sum_{j=0}^{W_i - 1} b_{i,j} \\ &= \sum_{i=0}^{M} b_{i,0} \sum_{j=0}^{W_i - 1} \frac{W_i - k}{W_i} \\ &= \sum_{i=0}^{M} b_{i,0} \frac{W_i + 1}{2} \\ &= \sum_{i=0}^{M} b_{i,0} \frac{2^i W_{min} + 1}{2} \\ &= b_{0,0} \frac{W_{min} + 1}{2} \sum_{i=1}^{M-1} \big(p^i.b_{0,0} \frac{2^i W_{min} + 1}{2}\big) + \big(\frac{p^m.b_{0,0}}{1-p}\big(\frac{2^m W_{min} + 1}{2}\big) \\ &= \frac{b_{0,0}}{2}\big[W_{min} \times \big(\sum_{i=0}^{M-1}(2p^i) + \frac{(2p)^M}{1-p}\big) + \frac{1}{1-p}\big] \end{aligned} \tag{4.11}$$

Hence, $b_{0,0}$ can be derived as follows:

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W_{min}+1) + pW_{min}(1-(2p)^M)}. \tag{4.12}$$

Combining Equations (4.12), (4.8), (4.9), and replacing them into Equation (4.6) results in a close form expression for the packet transmission probability of the ordinary transmitters of the network. Hence,

$$q = \sum_{i=0}^{M} b_{i,0} = \frac{b_{0,0}}{1-p} = \frac{2}{1 + W_{min} + pW_{min}\sum_{j=0}^{M-1}(2p)^j} \tag{4.13}$$

Equations (4.5) and (4.13) form a system of non linear equations with a unique solution given the system parameters N,W_{min} and M. This system of equations can be solved numerically in order to evaluate p and q.

Remark 4.1. The derivation of Equation (4.13) is according to the work by Bianchi [2] in which the transmitter is assumed to persist on retransmitting each packet until the receiver successfully acknowledges the packet. In many practical scenarios if the channel is not under extreme conditions and the network is not overloaded, the transmitter will eventually manage to transmit its packet after few retransmission attempts. However, in order to prevent a remote possibility of spending unreasonably long periods of time on transmitting a single packet, the retransmission process is set to be terminated after a predefined number of attempts and the packet is ignored. This modification changes the transmission probability of a node in the system (i.e., q) as it is discussed in [93]. However, for the sake of simplicity, we adopt the unlimited transmission model of Bianchi (i.e., Figure 4.1) in our design process. It is worth noting that the other model that limits the number of retransmission attempts at the transmitter can be simply adopted for covert communication through similar procedure as the one that is descried in this chapter.

Transmission Rate Design Criterion

One of the most prominent properties of a transmitter is its transmission rate. Hence, if the covert transmitter has a different transmission rate as compared to a regular user in

the system, it can be easily detected by a system observer. Thus, as a design criterion for the FRCC scheme, the transmission probability of the covert transmitter (i.e., σ) is restricted to be the same as the packet transmission probability of regular users of the system. Therefore, one can restrict the transmission probability of the covert transmitter as $\sigma \equiv q$. In this way, Equations (4.5), and (4.13) can be used in derivation of the parameters of the covert transmitter as the existence of the covert transmitter does not upset the balance between the transmission failure probability (i.e., p) and the packet transmission probability of a node in the network (i.e., q).

Transmission Window Design Criterion

Let d_r^0 be the average number of time slots that each regular user has to wait before its first packet transmission attempt. Thus, given the initial size of the contention window (i.e., W_{min}) and the fact that the backoff timer is selected randomly, one can derive d_r^0 as follows:

$$d_r^0 = \frac{W_{min} - 1}{2}. \tag{4.14}$$

On the other hand, in order to send the covert message $\omega \in \{0, 1, ..., T_0 - 1\}$, the covert transmitter waits until it observes ω packet transmissions from members of the covert set before it sends a packet over the channel. In fact, this is the mechanisms that the covert transmitter uses to mark the value of the covert clock, and communicate with the covert receiver. Let π be the probability of a successful packet transmission by a legitimate node of the system. Thus,

$$\begin{aligned} \pi &= q(1-p) \\ &= q(1-q)^{N-1}(1-p_e) \end{aligned} \tag{4.15}$$

Therefore, the probability of observing a successful packet transmission from members of the covert set can be derived as follows:

$$P_S = |S| \times \pi. \tag{4.16}$$

It is worth noting that in addition to Equation (4.16), the covert transmitter can progressively improve its estimation of P_S during the course of transmission. In more detail, by observing more packet transmission activities over the channel, specially from members of the covert set, the covert transmitter can update its estimation of P_S, and adapt its packet transmission policies accordingly.

In order to emulate an ordinary user of the network, *the covert transmitter has to spend, on average, the same number of time slots before its first transmission attempt as compared to any regular user of the system.* This observation forms the second design criterion for the FRCC scheme. It also dictates the value of the initial transmission window of the covert transmitter (i.e., T_0). Let d_c^0 denote the average number of slots that the covert transmitter has to wait before its first transmission attempt of a packet. In this way, one can derive d_c^0 as the average number of time slots until the covert transmitter observes exactly ω packets from members of the covert set, where $\omega \in \{0, 1, ..., T_0\}$. Thus,

$$
\begin{aligned}
d_c^0 &= \frac{1}{T_0} \sum_{\omega=0}^{T_0-1} \sum_{n=\omega}^{\infty} n. \binom{n-1}{\omega-1} P_S^{\omega-1}.(1-P_S)^{n-\omega}.P_S \\
&\overset{(1)}{=} \frac{1}{T_0} \sum_{\omega=0}^{T_0-1} \sum_{x=0}^{\infty} (x+\omega). \binom{x+\omega-1}{\omega-1} P_S^{\omega}.(1-P_S)^{x} \\
&= \frac{1}{T_0} \sum_{\omega=0}^{T_0-1} \Big[\sum_{x=0}^{\infty} x. \binom{x+\omega-1}{\omega-1} P_S^{\omega}.(1-P_S)^{x} \\
&+ \omega. \sum_{x=0}^{\infty} \binom{x+\omega-1}{\omega-1} P_S^{\omega}.(1-P_S)^{x} \Big] \\
&\overset{(2)}{=} \frac{1}{T_0} \sum_{\omega=0}^{T_0-1} \Big[\sum_{x=0}^{\infty} x. \binom{x+\omega-1}{\omega-1} P_S^{\omega}.(1-P_S)^{x} + \omega \Big] \\
&\overset{(3)}{=} \frac{1}{T_0} \sum_{\omega=0}^{T_0-1} \Big[\frac{\omega(1-P_S)}{P_S} + \omega \Big] \\
&= \frac{T_0-1}{2P_S}.
\end{aligned}
\tag{4.17}
$$

The equality in (1) is according to the change of variables as $x = n - \omega$. Meanwhile, (2) and (3) are based on the definition of the negative binomial distribution function [84]. Combining Equations (4.14) and (4.17), one can state the second design criterion for the FRCC scheme by setting the two parameters d_c^0 and d_r^0 to be equal (i.e., $d_c^0 \equiv d_r^0$). Consequently, the initial size of the transmission window of the covert transmitter (i.e., T_0) is derived as follows:

$$T_0 = P_S(W_{min} - 1) + 1. \tag{4.18}$$

Expansion Postpone Design Criterion

Following each packet transmission attempt, a normal CSMA/CA user expects the receiver to acknowledge the correct delivery of the transmitted packet. If the packet is acknowledged by the receiver, the transmitter resets it binary backoff algorithm, and prepares for the next packet to be transmitted. However, there are cases in which two nodes access the channel simultaneously (i.e., network collision), or the packet is lost due to the error-prone nature of the wireless channel. To handle such scenarios, the CSMA/CA algorithm is designed to enter a retransmission phase in which the transmitter expands its contention window and waits for another random period of time before it retransmits the packet.

The channel access mechanism of the FRCC scheme (i.e., Figure 4.2) is designed to mimic the same behavior as a normal transmitter in the network. Thus, after an unsuccessful transmission attempt, the covert receiver has to enter the retransmission phase, and retry delivery of its packet to the receiver. On the other hand, the covert transmitter is obliged to keep its synchronization with the covert receiver if the two nodes are about to maintain their covert channel. To this end, following each unsuccessful packet transmission attempt and before moving to the next stage of the algorithm, the covert transmitter resets the covert clock to $\sum_{j=0}^{i-1} T_j$ by waiting another $T_{i-1} - \omega$ clock ticks. Where, i is the number of unsuccessful transmission attempts for the current packet. This re-synchronization task accounts for additional delay for the covert transmitter as compared to regular users in the system. Thus, if the covert transmitter doubles the size of its contention window after each unsuccessful transmission (similar to what happens in a normal CSMA/CA user), the

average number of time slots that the covert transmitter waits between consecutive packet transmissions may deviate from the same parameter of ordinary users. Different wait times translates into different packet transmission behaviors which can be detected by a system observer, and reveal the existence of the covert channel.

To handle the mismatch in wait times, the covert transmitter is designed to postpone expanding its transmission window by α stages. For this purpose, the *expansion postpone parameter* (i.e., α) of the covert transmitter is selected such that *the average number of slots between the last successful packet transmission, and the α^{th} re-transmission attempt to send a new packet converges for both ordinary users and the covert transmitter.* The postpone strategy enables the covert transmitter to control the delay between retransmission attempts, and compensate for the additional delay due to the synchronization process.

To find the proper value for the expansion postpone parameter (i.e., α), one has to calculate the average number of slots that each user in the network (i.e., the covert transmitter, and the normal user) has to wait before it completes its α^{th} retransmission attempt. For the covert transmitter, since it does not expand its transmission window up to the stage α, the average number of slots between the last successful packet transmission, and the i^{th} re-transmission attempt for a new packet ($i \leq \alpha$) can be written as the average number of slots to observe $iT_0 + \omega$ packets from members of the covert set. Thus, similar to the calculation of Equation (4.17) we have,

$$\begin{aligned} d_c^i &= \frac{1}{T_0} \sum_{t=iT_0}^{(i+1)T_0-1} \sum_{n=t}^{\infty} n. \binom{n-1}{t-1} P_S^{t-1}.(1-P_S)^{n-t}.P_S \\ &= \frac{(2i+1)T_0 - 1}{2P_S}. \end{aligned} \tag{4.19}$$

A regular user of the network, on average, spends $\frac{W_j-1}{2}$ time slots in the stage j before retransmitting the packet. It also spends one slot trying to transmit the packet at the end of each stage. Hence, the average number of slots between the last successful packet transmission and the i^{th} retransmission attempt for the next new packet (i.e., d_r^i) can be

written as follows:

$$d_r^i = \sum_{j=0}^{i} \frac{W_j - 1}{2} + i$$
$$= \frac{W_{min}(2^{i+1} - 1) + i - 1}{2}. \tag{4.20}$$

Combining the expressions for d_r^i and d_c^i, the postpone expansion design criterion of the FRCC scheme can be applied by finding the index of the last stage in which the covert transmitter spends more time slots, on average, trying to transmit a packet as compared to a regular user. Based on this observation, one can derive the value of the postpone parameter (i.e., α) as follows,

$$\alpha = \max_{i>0} \{i|\ d_r^i \ \leq \ d_c^i\}$$
$$= \max_{i>0} \{i|(\frac{W_{min}(2^{i+1} - 1) + i - 1}{2} \leq \frac{(2i+1)T_0 - 1}{2P_S})\}$$
$$= \max_{i>0} \{i|(\frac{W_{min}(2^{i+1} - 1) + i - 1}{2} \leq \frac{(2i+1)(P_S(W_{min} - 1) + 1) - 1}{2P_S})\}$$
$$= \max_{i>0} \{i|(\frac{2^i - (i+1)}{i} \leq \frac{2 - 3P_S}{2P_S W_{min}})\} \tag{4.21}$$

4.4 Adaptive Rate Covert Channel (ARCC)

In order to achieve stealthiness, the covert transmitter is able to to expand the size of its contention window after unsuccessful transmission attempts. In principle, by doubling the size of the contention window, the covert transmitter may add an additional covert information bit to the original covert message (i.e., ω) and increase the covert rate. This rate increase is due to the availability of more states that corresponds to a particular covert message after each expansion in the transmission contention window. The Adaptive Rate Covert Communication scheme (i.e., ARCC) is designed to exploit the aforementioned extra

capacity in order to increase the covert communication rate between the covert receiver and the covert transmitter.

The transmission functionality in the ARCC scheme is similar to the one that is described for the FRCC scheme. The only exception is on how these two schemes handle failed packet transmission attempts. In other words, while the covert transmitter of the FRCC always works with a fixed covert message (i.e., $\omega \in \Omega$, and $|\Omega| = T_0$), the covert message in the ARCC scheme expands depending on the number of states that are available in each stage (i.e., $\omega_i \in \Omega$, and $|\Omega_i| = T_i,\ i = 0, 1, ..., M$). By expanding the size of the covert message, the covert transmitter can modulate more covert information in each retransmission attempt. Thus, the channel covert rate is improved.

In brief, upon failure in transmitting a packet, if the contention window of the covert transmitter is supposed to be expanded for the next stage (i.e., $i \geq \alpha$, and $T_i = 2T_{i-1}$), the covert transmitter fetches the next covert-bit and appends it as the *most-significant* bit to the current covert message. Hence,

$$\omega_i = (b_i||\omega_{i-1}), \quad i \in \{\alpha, \alpha + 1, ..., M\} \tag{4.22}$$

Where, ω_{i-1} is the covert message from the previous stage, and b_i is the additional covert-bit that will be transmitted using ARCC. This is analogous to the fact that the number of states that are available after each expansion of the contention window are doubled as compared to the number of states in the previous stage (i.e., Figure 4.2). This process is repeated after each failed transmission attempt until the packet is transmitted successfully, or the contention resolution algorithm reaches the stage in which the number of states are no longer expanded (i.e., M). Algorithm 3 depicts the transmission procedure of the ARCC scheme. The function *getbit*() returns the next covert-bit which is being concatenated to the original covert message.

In order to decode the covert message in the ARCC scheme, the covert receiver has to account not only for the value of the covert clock, but also it has to keep track of how many covert information bits are added into the original message. This makes the decoding at the covert receiver a much more complex task as compared to the simple decoding method

Algorithm 3 Covert transmitter function of ARCC.

/* ω is the original covert message.
/* *getbit*() returns extra information bits to be concatenated to the original message.
$i = 0;\ T_{-1} = T_0;$
$\omega_j = \omega, \quad j \in \{0, 1, ..., \alpha\};$
repeat
  if $\alpha < i < M$ **then**
    $b_i = getbit(i);$
    $T_i = 2^{i-\alpha}.T_0;$
    $\omega_i = (b_i||\omega_{i-1});$
  else
    $T_i = T_{i-1};$
  end if
  $Success = sendpkt(T_i, \omega_i);$
  $i = i + 1;$
until *Success*

of the FRCC scheme. To this end, the covert receiver first decodes the original message similar to the decoding process at the FRCC scheme (i.e., Equation (4.2)). Then, it checks the existence of extra information bits by removing the effect of first α stages from its covert clock. It is noted that the transmitter's contention window is not expanded for the first α stages. Hence, there would be no rate increase on those stages. If the value of the covert clock (i.e., C_r) is still positive, it means that the transmitter had more than α unsuccessful re-transmission attempts, and it had to expand its contention window. The receiver counts the number of expansions by removing multiples of T_i from the value of the covert clock and progressively decodes the extra covert bits. Algorithm (4) depicts the ARCC decoding process.

Algorithm 4 ARCC decoding at the receiver.

function $message = decode(C_r)$
/* First decode the original message ω_o
$\omega_o \equiv C_r \pmod{T_0}$;
$C_r = C_r - \omega_o$;
/* Decode the additional message ω_a
if $C_r < \alpha T_0$ **then**
  return ω_o
else
  $C_r = C_r - \alpha T_0$;
  for $i = 1$ *to* $M - \alpha$ **do**
    if $C_r < 2^i T_i$ **then**
      $\omega_a = \frac{C_r}{T_0}$;
      return $(\omega_a || \omega_o)$
    else
      $C_r = C_r - 2^i T_i$;
    end if
  end for
  $C_r \equiv C_r \pmod{2^M T_0}$;
  $\omega_a = \frac{C_r}{T_0}$;
  return $(\omega_a || \omega_o)$
end if

4.5 Performance Analysis

In this section the performance of the proposed covert communication schemes is analyzed. First the undetectability of the proposed covert channel is discussed followed by a thorough analysis of the robustness of the proposed wireless covert communication scheme under various network conditions and error events. Finally, the achievable covert rate of both proposed covert communication schemes (i.e., FRCC and ARCC) are studied. Table 4.1

Table 4.1: Wireless network parameters.

Parameter	Selected value
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
Transmission rate	1 $Mbps$
Payload size	1500 $Bytes$

depicts wireless network parameters that is used in this section. The designated values for system parameters (e.g., DIFS, SIFS) are selected according to the specifications by IEEE 802.11 standard committee [91]. The capacity of the overt channel is selected to be fixed at $1Mbps$ during the reliability and undetectability analysis of the covert channel. It is noted that choosing any other channel rate would not change the result of our analysis on the robustness and stealthiness properties of the proposed covert communication scheme. On the other hand, as the capacity of the wireless channel plays a decisive role in defining the achievable covert rate of the covert channel, the covert rate analysis is performed under a series of different rates for the wireless channel.

The performance analysis is conducted on four distinct scenarios each of which have different network parameters such as number of users, and the size of the covert set. For each scenario, the parameters of normal network transmitters (i.e., W_{min}, W_{max}, M) are selected according to the recommendations by the IEEE standard committee [91]. Meanwhile, the parameters of the covert transmitter (i.e., T_0, α) are derived according to the proposed design criteria for the covert communication scheme (i.e., Section 4.3.3). The simulation model consists of implementations of a covert transmitter and a covert receiver that use the proposed covert communication scheme in order to open a covert channel in the model. Table 4.2 illustrates the corresponding design parameters of the covert channel in each scenario.

Table 4.2: Design parameters for different simulation scenarios.

Parameter	SC1	SC2	SC3	SC4
Number of users (N)	25	35	50	15
Size of the covert set ($\|S\|$)	16	21	33	10
Covert transmitter minimum window size (T_0)	4	7	8	4
Expansion postpone parameter (α)	1	1	1	1
Regular user minimum window size (W_{min})	16	32	32	16
Regular user maximum window size (W_{max})	1024	1024	1024	1024
Number of back-off stages (M)	6	5	5	6

4.5.1 Stealthiness of the Covert Channel

Network covert channels are usually designed in order to hide the fact that information is exchanged in the system. Thus, it is important to design a covert channel such that it is extremely difficult for a network observer, even with complete knowledge of the covert communication scheme, to detect the covert channel. In this section we study the stealthiness property of different elements of the proposed covert communication scheme including the covert transmitter and the covert receiver.

Covert Receiver

As the covert receiver is a completely passive entity in the proposed covert communication scheme, it is undetectable even if the covert channel is detected. Moreover, since the covert transmitter does not need to know the identity of the covert receiver, the covert receiver is safe even if the covert transmitter is compromised or its information is exposed to a system observer.

Covert Transmitter and the Covert Channel

The proposed covert communication scheme is designed based on the concept of behavioral mimicry. In other words, the covert transmitter is designed to mimic the same transmission characteristics as a normal user of the system even though it is committed to maintain the covert communication with the covert receiver. According to the design description of the proposed covert communication scheme, it can be observed that the structure of the contention resolution algorithm of the covert transmitter is specifically designed to approximate the transmission pattern of a regular transmitter of the network. In this way, the covert transmitter is capable of blending itself into the group of ordinary nodes that share the wireless channel and remain undetected.

On the other hand, system observers are equipped with a set of tools that are specifically designed to detect covert channels. Thus, in order to evaluate the undetectability of the proposed covert communication scheme the covert channel is tested against two of the most well-known statistical tests that are used to detect covert channels in communication network. In what follows, the effectiveness of the *Kolmogorov-Smirnov test*, and the *regularity test* in detecting the proposed covert channel is investigated.

Kolmogorov-Smirnov Test Analysis

Kolmogorov-Smirnov test (i.e., KS-test) is designed as a measure to show the difference in the cumulative distribution function of the sampled data from the covert transmitter's traffic as compared to the distribution of the legitimate traffic of the same network. This test has been already used in detecting watermarked inter-packet delays, and it is a major tool in detecting timing covert channels [37]. Let $S(x)$ be the empirical distribution of the inter-packet delays of the covert traffic. Let $F(x)$ be the cumulative distribution function of inter-packet delays of the legitimate traffic in the same system. The KS-test is defined as follows:

$$H_s = \sup_x |S(x) - F(x)|. \tag{4.23}$$

The test result is the answer to the validity of the null hypothesis that the two samples

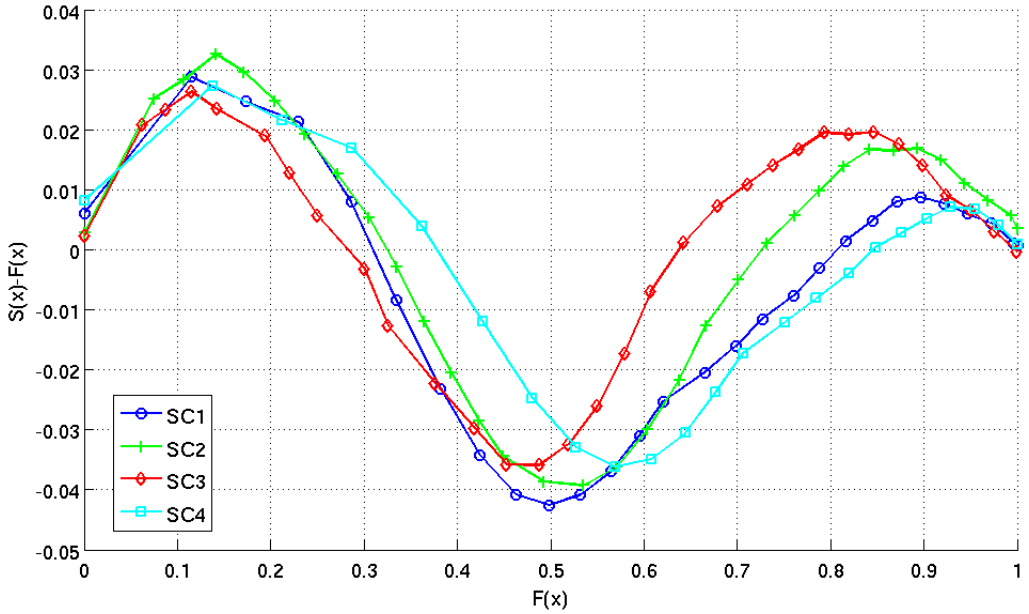


Figure 4.3: Kolmogorov-Smirnov shape test for different simulation scenarios.

are drawn from the same distribution. In this way, the null hypothesis is rejected (i.e., a covert channel may exist) if the value of H_s is beyond a predefined threshold (e.g., 0.05). Figure 4.3 illustrates the difference between the cumulative distribution function of the inter-packet delays of the legitimate traffic and the covert traffic. According to the graph, the peak difference between the two distributions is less than 5% which is an acceptable margin for the KS-test. Such a small difference makes it extremely difficult for an observer to detect any abnormal behavior in the system based on first level statistical tests such as the KS-test. In other words, the test accepts the null hypothesis that the channel access pattern of the covert transmitter follows a similar probability distribution function as compared to the one that is exhibited by normal users of the wireless channel.

Figure 4.3 also highlights how the covert transmitter systematically adapts its behavior in order to emulate the same transmission pattern as an ordinary user of the system. According to the discussion in Section 4.3.3, the covert transmitter postpones the expansion of its transmission window by α stages in order to compensate for the extra delay caused

by the synchronization process. Therefore, the covert transmitter gets to have a slightly higher transmission rate up to the point where it starts expanding its transmission window (i.e., the first peak of the graph). Eventually, by expanding the size of the contention window the covert transmitter has to wait longer before its next retransmission attempt. Thus, the covert transmitter is forced to lag behind an ordinary user of the system until the point in which the contention window of a regular user is large enough such that the wait times for both covert transmitter and regular users converge (second extreme point of the graph). This adapting behavior is the key design feature of the proposed covert channel that enables the covert transmitter to track the transmission pattern of ordinary nodes in the network and mimic their behavioral fingerprints in order to avoid detection.

Regularity Test Analysis

In addition to the first level statistical tests (e.g., the KS-test), the stealthiness property of the covert channel is analyzed against the *regularity test* [35]. In principle, the regularity test is designed to detect the temporal abnormal behaviors of the covert transmitter. More precisely, the variance of the inter-packet delays of a normal network traffic flow changes over time due to different network events and channel conditions (e.g., packet loss, congestion). In fact, regular users of the system have the same reaction to sudden events in the network. Hence, the characteristics of the normal traffic of the network changes constantly during the course of communication. However, as the covert transmitter is committed to transmit a particular covert message, it may not be able to react to the changes in the network condition similar to other nodes in the system. The regularity test is meant to detect such behavioral differences and track down covert activities.

To calculate the regularity test score, samples of the inter-packet delays are collected and then spread into multiple sets of size γ. The standard deviation of each set is calculated to derive the regularity score H_r as follows:

$$H_r = std(\frac{|\sigma_i - \sigma_j|}{\sigma_i}, \quad \forall i, j, i < j). \tag{4.24}$$

Where, *std* is the standard deviation operation, and σ_i is the standard deviation of the

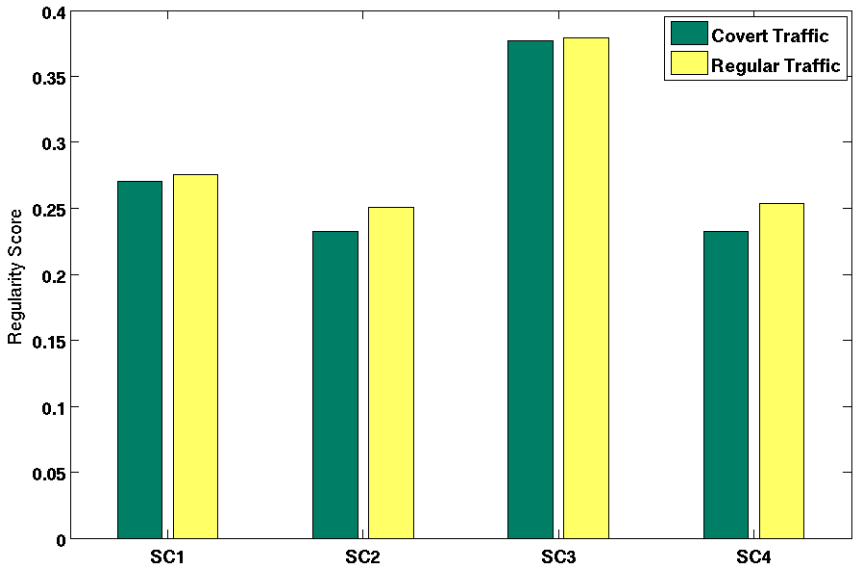


Figure 4.4: Regularity test for different simulation scenarios.

i^{th} set of inter-packet delays. A high regularity score means large variance in the statistical properties of the samples of each set which signals a constant change of behavior in the sampled traffic. In contrast, a low value for H_r depicts a set of regular inter packet delays that is likely being controlled by another process and may contain other information (e.g., a covert message).

Figure 4.4 shows the regularity score of the covert transmitter and the same score for ordinary users assuming $\gamma = 50$. According to the graph, the covert transmitter's regularity score is extremely close to the score of regular users in all four scenarios. In other words, the covert transmitter has managed to blend itself into the crowd well enough such that even its temporal behaviors are matched with normal users of the system.

The key in maintaining the regularity score of the covert traffic lies in the design of the contention resolution algorithm at the covert transmitter, and the effect of the covert clock on the functionality of the covert channel. By design, the covert clock changes according to activities of other nodes in the system. Thus, channel activities (i.e., packet transmission) of members of the covert set are considered as clock ticks for the covert clock. If the channel

condition changes in a way that normal network users have to wait for a longer period of time between consecutive packet transmissions (e.g., reduction of channel capacity, high error rates, etc), the covert clock advances with a slower pace. Consequently, the covert transmitter has to wait longer, in order to capture enough clock ticks, between any two consecutive packet transmission attempts. In contrast, if the wireless channel provides a higher transmission rate to normal users of the network, the covert clock would increase much faster. This immediately leads into a faster packet transmission rate at the covert transmitter.

By coupling the covert message modulation process to the channel activities of normal nodes in the network, the covert transmitter can constantly follow network events and modify its transmission characteristics in order to mimic the transmission pattern of normal users of the system. Such an adaptive behavior is an advantage of the proposed covert communication scheme as compared to other covert channel designs in the literature in which the regularity score is artificially controlled by switching from one transmission mode to another [40], or by replaying previously sampled traffic traces of the legitimate traffic and switch from one sample set to another one periodically [38].

4.5.2 Robustness Analysis

In addition to high undetectability level a covert channel has to be robust as well. This means the covert receiver should be able to decode the transmitted covert message correctly and without any ambiguity. In this section, the performance of the covert receiver in decoding the covert message is analyzed. First the network events that may cause the error at the covert receiver are identified. Then, methods to control the effect of the aforementioned error events on the channel robustness are discussed. In brief, two independent packet loss events are considered in order to evaluate the robustness of the proposed covert communication scheme. Before any further discussion, it is worth noting that by packet loss we refer to the event of losing a packet due to the inherent channel noise in the wireless channel. Hence, packet loss events do not cover the situation in which the transmitter fails to transmit the packet due to collision with other nodes in the channel. However, all

numerical analysis in this section are generated based on simulation models that consider the collision events as well.

The first error event is due to failure in detecting packets from members of the covert set. It is noted that the covert transmitter and the covert receiver rely on the activities of members of the covert set (i.e., clock ticks) in order to achieve synchronization. The covert clock also plays a major role in modulation and decoding of the covert message at the covert transmitter and the covert receiver, respectively. Hence, any mismatch in the covert clocks will directly affect the reliability of the covert channel. The second error event is caused by losing the covert transmitter's packet at the covert receiver. The covert transmitter uses its packets in order to mark the value of the covert clock that corresponds to the covert message. The covert transmitter's packets also trigger the decoding process at the covert receiver. Consequently, if the covert receiver fails to detect the covert transmitters packet, the synchronization between the covert transmitter and the covert receiver is lost which leads to an errornous detection of the covert message at the covert receiver.

Countermeasures for Packet Loss Events

The error-prone nature of the wireless channel necessitates the existence of error control mechanisms virtually in all wireless communication protocols. Thus, covert communication schemes can be designed to take advantage of the aforementioned error control mechanisms in order to improve the robustness of the covert channel. To this end, we have identified some properties of the IEEE 802.11 communication standard which can be used in order to boost the performance of the covert receiver in decoding the covert message.

Acknowledgment Message: In most communication protocols, each successful packet transmission is acknowledged by the actual receiver of the packet. The acknowledgment message usually contains the sequence number of the original packet and also the identification of the packet transmitter. In this way, one can learn about a successful packet transmission (e.g., packets from members of the covert set or the covert transmitter), either by detecting the packet itself or observing the corresponding acknowledgment message.

Packet Sequence Number: The packet sequence number field is also beneficial to the covert receiver, in particular to control the effect of losing packets from the covert transmitter. It is noted that losing the covert transmitter's packet has a pronounced effect on the synchronization between the covert transmitter and the covert receiver. It also may cause error propagation at the covert receiver. In other words, if the covert receiver fails to detect a covert transmitter's packet, it continues on incrementing its covert clock and waiting for the packet from the covert transmitter. On the other hand, the covert transmitter resets its covert clock and prepares for the next round of transmission, therefore the synchronization is lost. The covert receiver will regain synchronization with the covert transmitter when it receives another packet from the covert transmitter. Indeed, the covert receiver can learn about the lost covert transmitter's packets by comparing the sequence numbers of the last two consecutive packets that are received from the covert transmitter.

Using the sequence number field of the packets that are received from the covert transmitter, the covert receiver is able to learn about possible lost packets during the course of communication. Thus, the covert receiver can re-synchronize itself with the covert transmitter, and pinpoint the location of the symbols that are lost due to packet loss events in the wireless channel. This approach prevents error propagation at the covert receiver.

Channel Coding: Channel coding [51] is another approach to combat the destructive effect of packet loss events on the robustness of the covert communication scheme. In this way, the covert transmitter encodes the original message (e.g., the binary sequence $\mathcal{B}$) into a coded covert message by adding extra redundancy information to the original message. Then, the coded message is modulated and transmitted using the proposed covert communication schemes (i.e., the FRCC or the ARCC). Encoding the original message enables the covert receiver to tolerate more error events during the course of communication, and still recover the original message with no ambiguity. However, the reliability is achieved at the cost of the extra bandwidth that is required to send the longer and more redundant coded message.

Numerical Results

In order to evaluate the robustness of the proposed covert communication scheme, a simulation framework is implemented in which a group of normal CSMA/CA users share a wireless channel with a covert transmitter and a covert receiver that are designed based on the specifications of the FRCC scheme. Meanwhile, for the sake of simplicity it is decided to report only the numerical results that are derived according to the first simulation scenario in Table 4.2. The covert transmitter and the covert receiver are equipped with routines to take advantage of the acknowledgment messages and packet sequence number fields in order to handle possible packet loss events in the channel. In addition, channel coding is used at the covert transmitter in order to reduce the likelihood of errornous decoding of the covert message at the covert receiver. In this way, a rate 1/3 convolutional code with the generator matrix $[47; 53; 75]$, and a rate 1/4 convolutional code with the generator matrix $[17; 13; 13; 15]$ are used to encode the covert message at the covert transmitter. The numerical results are collected under two distinct channel configurations from the packet loss point of view.

- *Single error case:* In this setup, it is assumed that only one of the covert transmitter or the covert receiver experiences packet loss from members of the covert set.
- *Equal error case:* For the equal error case setup, the covert transmitter and the covert receiver are assumed to have same failure probability in detecting packets from members of the covert set.

It is noted that the packet loss events at the covet transmitter and the covert receiver are assumed to be independent. In addition, for the single error case simulation setup, it can be observed that applying packet loss events on either end of the covert channel does not change the decoding error rate at the covert receiver.

Figure 4.5 depicts the covert channel bit error rate (BER) based on the packet loss ratio of the packets from members of the covert set. So far, packets from the covert transmitter are assumed to be received error-free at the covert receiver. According to the graph, the

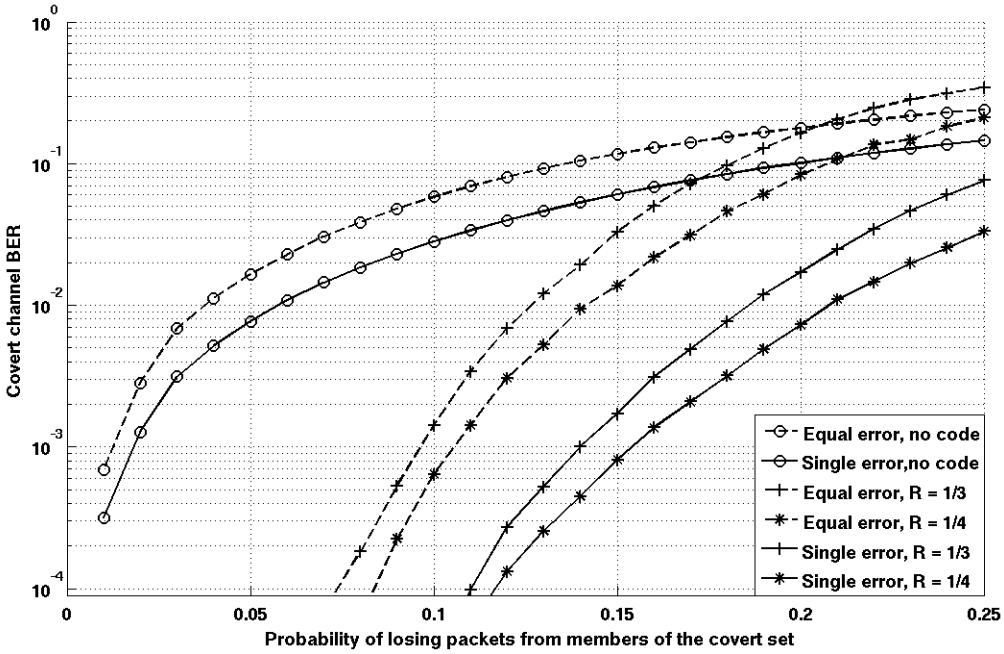


Figure 4.5: Covert channel BER versus the packet loss ratio for the packets from members of the covert set. The plots are according to the first scenario in Table 4.2. Packets from the covert transmitter are assumed to be always detected at the covert receiver.

covert communication scheme depicts a noticeable resilience against losing packets from members of the covert set, in particular at low packet loss ratios (e.g., less than 5%). This is due to the application of the acknowledgment messages in detecting packets from members of the covert set. The figure also highlights the effect of channel coding on boosting the robustness of the proposed scheme. For instance, in the case when both the covert transmitter and the covert receiver experience 10% packet loss from members of the covert set (i.e., equal error case), channel coding reduces the BER of the covert channel from 0.06 (i.e., the uncoded scenario), down to 1.5×10^{-3} for the rate 1/3 channel code, and even lower to 6×10^{-4} if a stronger channel code is used (i.e., the rate 1/4 code). In addition, one can observe that the effect of channel coding is much more pronounced in the single error simulation setup in which only one of the covert transmitter or the covert receiver experiences packet loss from members of the covert set.

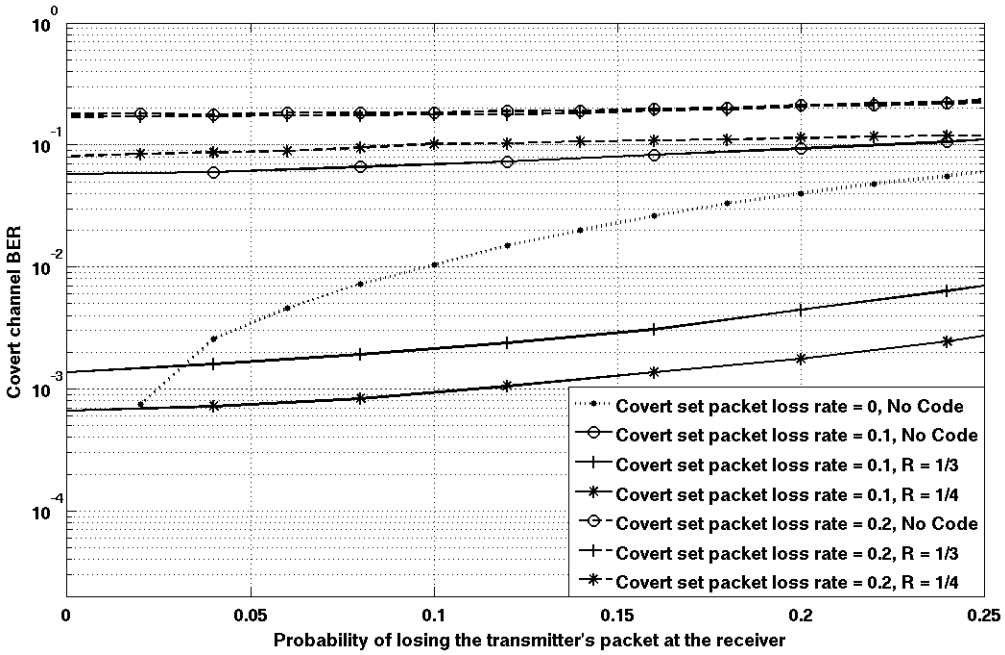


Figure 4.6: Covert channel bit error rate due to the loss of the covert transmitter's packets at the covert receiver. Packets from members of the covert set are subject to error with packet loss rate equal to 0 for dotted lines, 0.1 for solid lines, and 0.2 for dashed lines. The plots are according to the first scenario in Table 4.2.

Figure 4.6 depicts the BER at the covert receiver when the covert transmitter's traffic is also subject to error. The plots in Figure 4.6 are grouped based on the packet loss probability of the traffic from members of the covert set. Each group consists of three plots, one for the uncoded scenario, and the other two for scenarios with channel coding in place. It is also assumed that the covert transmitter and the covert receiver experience the same packet loss ratio for packets that are transmitted by members of the covert set (equal error case). Remarkably, the numerical analysis shows that if the covert transmitter and the covert receiver enjoy loss-less channels from members of the covert set, the effect of losing packets from the covert transmitter at the covert receiver is completely compensated by the extra redundancy that is provided by the implemented channel codes. Hence, those plots are not reported in the graph. Meanwhile, as the clock tick events of the covert clock

become noisy, failure in detecting the transmitter's packet at the covert receiver becomes noticeable in degrading the performance of the FRCC scheme. In fact, losing track of any packet from the covert transmitter accounts for losing at least one covert symbol at the covert receiver. However, the covert receiver benefits from the acknowledgment messages to reduce the chance of losing the covert transmitter's packet. Moreover, the covert receiver can learn about the exact position and the number of lost transmitter's packets, using the sequence number field of the packets it receives correctly. This information is used to improve the decoding performance at the covert receiver.

By comparing the plots in Figure 4.5 and Figure 4.6, it can be observed that the effect of losing packets from members of the covert set is much more prominent as compared to the effect of error events due to failure in detecting the covert transmitter's packets at the covert receiver. The flat plots of Figure 4.6 confirms this observation proving the fact that the events that form the covert clock (i.e., packets from members of the covert set) are far more decisive in dictating the robustness of the proposed covert communication scheme as compared to the signaling events of the proposed covert channel (i.e., packets from the covert transmitter).

4.5.3 Covert Rate Analysis

One salient feature of the proposed covert communication scheme is that it allows the covert transmitter and the covert receiver to keep their overt communication channel. It is noted that the communication rate of a transmitter is one of the basic parameters that are constantly monitored by system observers. Hence, in order to achieve stealthiness, the covert transmitter has to control its overt communication rate such that it does not deviate from the behavior of an ordinary user of the channel. In fact, this observation is the basis for the *transmission rate design criterion* for the proposed covert communication scheme. In other words, the proposed covert channel is designed with a covert transmitter that mimics the transmission rate of a normal node in the network, yet its overt traffic contains a covert message for the covert receiver.

Figure 4.7 compares the overt communication rate of the covert transmitter as compared

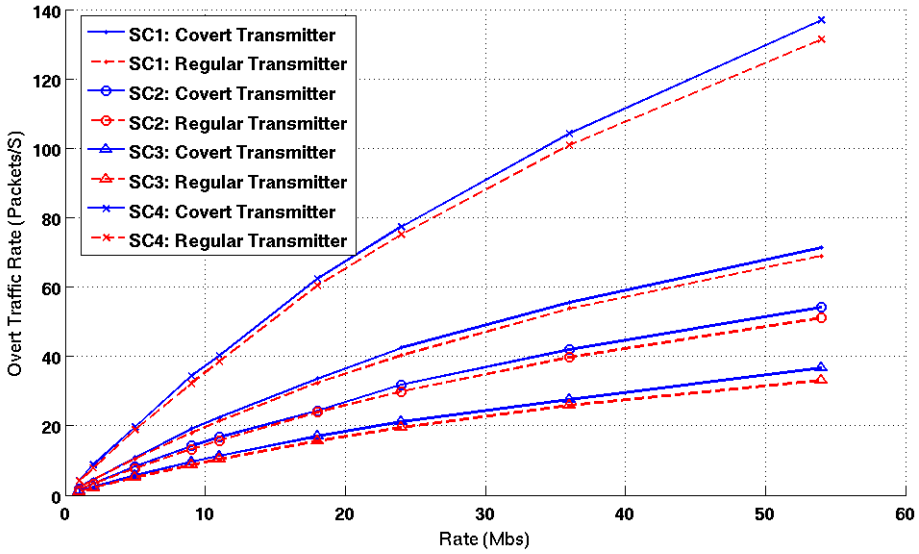


Figure 4.7: Overt communication rate of the covert transmitter and normal network nodes.

to regular nodes in the system. The result in Figure 4.7 confirms the validity of the transmission rate design criterion that guides the covert transmitter to mimic the same transmission behavior as normal transmitters in the channel. According to the graph, in all four scenarios, the overt rate of the covert transmitter is extremely close to the transmission rate of normal users of the network. Such a close similarity in overt communication rate is fundamentally important in order to secure the identity of the covert transmitter against traffic analysis attacks.

Finally, the achievable covert rate of the proposed covert communication scheme is depicted in Figure 4.8. The graph also illustrates how the covert rate of the proposed covert communication scheme scales linearly with the overt communication rate of the network. In fact, the covert channel is capable of achieving relatively high covert rates without compromising the covertness property of the channel. The contribution of using ARCC scheme in boosting the achievable covert rate of the channel is also highlighted in the graph. The dashed lines in Figure 4.8 represent the difference in the achievable covert rate of the ARCC scheme as compared to the covert rate that can be achieved using fixed rate covert communication scheme (i.e., FRCC). In fact, using the ARCC scheme

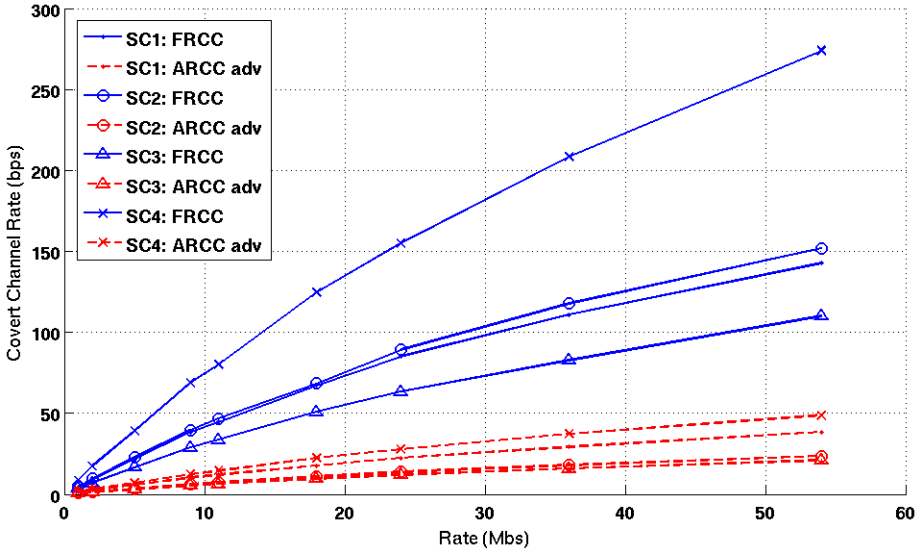


Figure 4.8: Achievable covert rate of the proposed covert channel. Dashed lines depict the additional capacity (advantage) that can be achieved using the ARCC scheme.

one can increase the covert rate of the channel by as much as 20% which is a significant improvement in the achievable covert rate of the proposed covert communication scheme.

4.6 Target Applications

Wireless communication systems have attracted a great deal of attention in communication industry due to their unique characteristics such as high flexibility and scalability. Today, mobile devices are incorporated in virtually every aspect of modern societies. In line with such rapid growth of wireless technology, wireless covert channels offer unique perspectives on how to provide security, and privacy in such systems.

Sensor networks are among the most promising target environments for deploying covert channels. The decentralized structure of such networks combined with high degrees of randomness make sensor networks an appealing setting for covert communication. Covert channels can be utilized in transmitting sensitive information (e.g., authentication infor-

mation, broadcast commands, and critical updates) in order to increase system security and prevent adversaries from conducting selective drop attacks on critical information flows in the network. Covert channels can also be used as a mechanism to safeguard the operation of sensor networks. To this end, a covert message is embedded in the behavioral fingerprints of the elements of the network such that a particular node or a group of nodes can be recognized by the embedded covert message. By comparing the expected covert message with the one that is decoded from the behavioral characteristics of each node, the monitoring entity can detect abnormal behaviors and trace them back their sources in the system.

As an example, in [64] a covert communication scheme based on routing protocols in ad-hoc networks was proposed. In this approach, the covert transmitter generates route request messages (i.e., RREQ) according to the covert messages it intends to transmit. The content of the RREQ messages (e.g., destination ID, request lifetime, *etc*) or the frequency of generating the messages can be used as the medium for covert communication. Now, consider a system in which the routing protocol of network elements is adopted to embed a covert message according to the aforementioned covert communication scheme. The system observer plays the role of the covert receiver, and monitors the transmitted covert messages by other nodes in the network. Thus, when a node behaves differently than expected, due to node compromise or malfunction, the monitoring entity can detect the anomaly and identify the defective node. A similar functionality can be assigned to the proposed covert communication scheme in order to monitor the packet transmission behavior of elements of a WLAN network. In such a configuration, the monitoring entity (i.e., the covert receiver) expects a specific covert message to be embedded in the channel access process of each legitimate node of the network. Thus, any node with abnormal behavior that violates such restriction can be detected and removed from the system. This approach is effective in particular against rogue nodes that are designed to remain hidden against statistical analysis tests (e.g., another covert channel).

4.7 Chapter Summary

In this chapter a new covert channel is proposed which is based on the structural behavior of CSMA/CA and the binary backoff algorithms. To this end, a covert transmitter is designed by modifying the CSMA/CA algorithm such that the transmitter's contention window is controlled by a virtual clock called the covert clock. The covert clock is linked to the channel activities of a subset of regular nodes in the system using the broadcast nature of the wireless channel. These activities are observed by the covert transmitter and the covert receiver in order to synchronize their covert clocks and communicate through the covert channel.

The proposed covert communication scheme is an example of using behavioral mimicry in designing stealth and robust covert channels in wireless networks. In fact, the performance analysis of the proposed covert channel verifies that the channel can evade detection by well known statistical tests that are designed to detect covert channels in communication networks. In more detail, since the packet transmission mechanism of the covert transmitter is influenced by the activities of other users in the system (i.e., the covert clock), the covert transmitter mimics not only the long term characteristics of a legitimate node but also reacts to the temporal changes in the system similar to a typical network transmitter. Thus, the covert channel remains hidden due to the indistinguishable packet transmission pattern of the covert transmitter as compared to any other node in the network. Using properties of the original communication protocol (e.g., acknowledgment messages) the covert channel can achieve high levels of reliability and operate in the error-prone wireless environment. Another important feature of the proposed scheme is the relatively high covert rate of the covert channel which scales linearly with the overt communication rate of the network.

Chapter 5

Turbo Covert Channel

Similar to the wireless environment, public communication networks (e.g., Internet) can be considered as suitable frameworks for covert communication. In fact, in a majority of such network structures, one can identify the three requirements of existing a covert channel according to the Kemmerer principle. A public network is accessible by many users providing a shared environment, each user can transmit and receive information by generating data packets and forwarding them towards the destination address. Hence, the network can be modified by introducing new traffic flows or modifying the pattern of the packets in each flow. Finally, each packet contains multiple header fields (e.g., TCP header, or IP header) each of which are used in one of the protocol layers of the communication network. These header fields carry enough information to achieve synchronization between the transmitter and the receiver. For instance, the sequence number field of TCP packets over the Internet can be considered as a very useful tool to achieve synchronization in covert communication schemes.

An interesting feature of public communication networks is the range of the network which is virtually unbounded and can go beyond physical and geographical barriers in the world. Thus, covert channels in public communication networks can reach much farther as compared to covert channels over wireless networks which are limited by the range of the wireless signal. The existence of an enormous number of users, diverse communication

protocols, and huge volume of information that is exchanged over such communication networks gives a valuable advantage to the covert transmitter and the covert receiver to blend among the crowd and remain hidden effectively. In fact, given the scale of the network, it is extremely difficult for a system observer to monitor all traffic flows and process them in order to distinguish a possible covert channel in the system. However, despite the promising properties of public communication networks for covert communication, designing and implementing a covert channel between two distant nodes has proved to be a challenging task.

As the size of a communication network increases, the network evolves into a very complex system that exhibits unpredictable behaviors. The unpredictable nature of communication network necessitates the existence of robust and reliable covert channel designs that are accustomed to handle extremely noisy environments and powerful adversarial disruptions. For instance, the packet arrival times in an Internet flow are subject to a random noise (i.e., network jitter) due to ever-changing network conditions (e.g., network congestion). Such a variation in the packet arrival times translates directly into noise for TCP/IP timing covert channels that use the Inter-Packet Delay (i.e., IPD) of the network traffic as the carrier for the covert message. Furthermore, an active adversarial entity can easily access the overt channel and maliciously change the channel behavior (e.g., change the IPDs of the covert traffic), in order to limit the capacity of the covert channel.

Inspired by the challenges of designing robust, undetectable, and high rate covert channels over public communication networks, in this chapter a new design methodology for provably undetectable timing covert channels over the Internet is introduced. The proposed framework is based on modeling the covert channel as a differential communication channel, and using the properties of the communication model in order to derive the formulation for modulation/demodulation processes. Then, a trellis structure is proposed that is used by the covert transmitter and the covert receiver in order to modulate and demodulate the covert message, respectively. The trellis structure also enables the covert receiver to perform iterative demodulation/decoding of the covert message and form a turbo covert channel that significantly improves the reliability of the system. In addition, an adaptive modulation strategy for the covert transmitter is designed that improves the robustness of

the channel significantly. To this end, the covert transmitter uses only the error-resistant portion of the IPD spectrum when it modulates the covert message. However, in order to keep the stealthiness property of the covert channel, the modulation scheme is designed such that it does not force the pattern of IPDs of the covert traffic to deviate from the inter-packet delay pattern of the legitimate traffic of the channel.

The aforementioned modulation strategy combined with the proposed trellis structure gives the covert channel considerable flexibility to achieve a wide range of covert rates with extremely low error probability at the covert receiver. Such a robust and reliable covert channel design can be used in watermarking the information flows that go inbound or outbound of a communication network infrastructure. In this way, one can track down possible malicious users even if privacy preserving applications (e.g., Tor) are used in order to hide the origin of the attack (e.g., distributed denial of service attacks).

5.1 TCP/IP Timing Covert Channels

In general, a timing channel is a communication channel in which the message is modulated by the timing of an specific sequence of events. In computer networks, timing covert channels are usually designed based on packet transmission events. The design may involve modulating the covert message by controlling the existence or absence of a packet transmission event in a known period of time, or by using the inter-packet delays (i.e., IPD) of a sequence of packets in a traffic flow. The latter approach provides the covert channel with a higher covert rate that is much harder to detect if designed properly.

Figure 5.1 depicts a basic example of a timing covert channel over the Internet. The objective for the covert transmitter is to modulate the covert message over inter-packet delays of the covert traffic. To this end, the covert transmitter designates two constellation points (i.e., delay values) to represents covert bit-0 (i.e., $d = 50ms$), and covert bit-1 (i.e., $d = 100ms$). The delay values are used in order to modulate the elements of the covert message (i.e., $m = \{m_0, m_1, m_2, ...\}$) over IPDs of the covert traffic. In this way, the covert transmitter delays the next packet in its covert traffic flow by $100ms$, if it intends

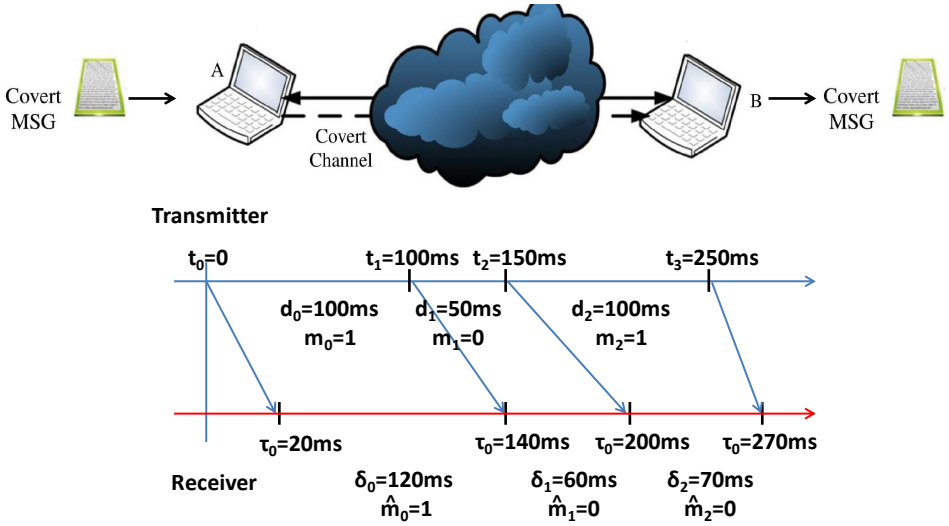


Figure 5.1: A simple timing covert channel. The covert information are modulated over IPDs of the covert traffic ($50ms$ delay for covert bit-0, and $100ms$ delay for covert bit-1).

to modulate the covert bit-1 (e.g., $m_0 = 1 \Rightarrow d_0 = 100ms \Rightarrow t_1 = t_0 + d_0 = 100ms$). On the other hand, in order to transmit the covert bit-0, the covert transmitter generates a packet that is delayed by $50ms$, with respect to the previous transmitted packet in the covert traffic (e.g., $m_1 = 0 \Rightarrow d_1 = 50ms \Rightarrow t_2 = t_1 + d_1 = 150ms$). On the other end of the channel, the covert receiver observes the arrival times of the received packets (i.e., $\tau_0, \tau_1, ...,$), and calculates the inter-packet delays of the covert traffic (i.e., $\delta_0, \delta_1, ...$). Then, the covert message is decoded by finding the corresponding covert bits to the IPDs of the received covert traffic. The aforementioned timing covert channel can achieve an average covert rate of $r = 10^3/(\frac{50+100}{2}) = 13.33$ bps, given the assumption that covert bit-0 and covert bit-1 are equiprobable.

Despite the clear objective of designing timing covert channels over communication networks, these channels are known to be extremely sensitive to the network noise. In particular, the variation in the packet arrival times of the covert traffic (i.e., network jitter) is a major barrier that limits the capacity of timing covert channels over the Internet. As an example, in the setup of Figure 5.1, the transmitter intends to transmit the covert message

$m = \{101\}$. Thus, the elements of the covert message are mapped to the corresponding IPDs according to the designated values for each element of m (i.e., $m = \{1, 0, 1\} \Rightarrow d = \{100\ ms, 50\ ms, 100\ ms\}$). The set of IPDs are used to define the delays of the packets in the covert traffic. However, due to the network noise (e.g., network jitter) the inter-packet delays that are observed at the covert receiver are different from the original IPDs (i.e., $\delta = \{\delta_0 = 120ms, \delta_1 = 60ms, \delta_2 = 70ms\}$). Hence, the covert receiver decodes the covert message in error as some of the observed IPDs have changed dramatically during the course of transmission (i.e., $\tau_2 = 70ms \Rightarrow \hat{m}_2 = 1 \neq 0 = m_2$).

The timing covert channel of Figure 5.1 suffers from another design flaw that is more of a concern than the error sensitive nature of the channel. In fact, the covert traffic of the covert transmitter exhibits distinctive characteristics that do not exist in normal traffic flows of the network. The covert traffic consists of a series of packets that are apart from each other either by $50ms$, or $100ms$ intervals. Such a regulated behavior is hardly observed in public communication networks. In other word, modulating the covert message over IPDs of the covert traffic has changed key characteristic properties of the covert traffic as compared to a normal traffic flow of the network. Thus, the covert channel is vulnerable to pattern analysis and statistical detection methods. Such attacks could reveal the existence of the covert channel, violate the privacy of the covert transmitter or the covert receiver, and even expose the covert message.

The aforementioned example highlights some of the difficulties in designing robust, undetectable, and high rate covert channels over communication networks. In what follows we present *turbo covert channel* (i.e., TCC), an iterative covert communication framework that is designed according to our design methodology, and it is capable of achieving *provable polynomial undetectability*. This means that the covert channel is undetectable against any polynomial-time statistical test that analyzes samples of the covert traffic and the legitimate traffic of the system. The TCC scheme also provides means to handle network noise and adversarial disruptions in order to improve the robustness of the covert channel.

5.2 System Model

A timing covert channel over the Internet consists of a covert transmitter that intends to send a covert message to at least one covert receiver, where the covert message (i.e., $\mathbf{m} = (m_1, m_2, ..., m_k)$) is a sequence of binary elements (e.g., $m_i \in \{0, 1\}$). For this purpose, the covert message is modulated into a sequence of IPDs (i.e., $\mathbf{d} = (d_1, d_2, ..., d_n)$) that forms the covert traffic of the covert transmitter. The number of packets that is required to modulate the covert message (i.e., n) is a function of the message length (i.e., $|\mathbf{m}| = k$), the rate of the channel code, and the modulation scheme that is used by the covert transmitter. At the other end of the channel, the covert receiver detects a noisy version of the IPDs of the covert traffic (i.e., $\delta = (\delta_1, \delta_2, ..., \delta_n)$), as the arrival time of each packet is subject to an independent random delay. Finally, the received IPDs are processed in order to decode the covert message (i.e., $\hat{\mathbf{m}}$) at the covert receiver.

The legitimate traffic of the channel is assumed to have independent and identically distributed (i.e., (*i.i.d.*)) inter-packet delays. The (*i.i.d.*) traffic model is used in analyzing communication networks as it provides essential properties that makes model-based system designs and channel analysis tractable [94, 95]. Meanwhile, it is shown that such a model can be extended to more general traffic scenarios with minor analytical adjustments. For instance, in [96] it is suggested that a *batch renewal process*, with (*i.i.d.*) batch sizes and (*i.i.d.*) inter-batch delays, can be used in order to model a traffic source with correlated inter-packet delays.

The distribution of the inter-packet delays is also a source of vigorous research in communication networks. The early belief in modeling network traffic was to use the Poisson process to model the packet arrival and connection request events of a network traffic flow [94]. However, recent discoveries have suggested that for a majority of traffic sources, specifically those with self-similar behaviors (e.g., shell connections), Pareto distribution models provide a much more accurate estimation of the real traffic [95]. Hence, the channel model in this dissertation is modeled according to the Pareto distribution. However, to achieve a more realistic system model, the legitimate traffic of the channel is derived based on real world traffic samples according to CAIDA network traces [97].

5.3 Turbo Covert Channel

In principle, a timing covert channel is a differential communication channel that encodes the covert message into the difference of the arrival times of the packets of the covert traffic. In model-based covert channels, the covert transmitter incorporates its knowledge of the statistical properties of the legitimate traffic in order to generate a stream of packets (i.e., the covert traffic) with similar statistical properties as compared to the legitimate traffic, yet a covert message is embedded in the inter-packet delays of the covert traffic. In this section, a detailed description of the modulation and the demodulation algorithms of the proposed covert communication scheme is presented. Then, the decoding of the covert message is discussed followed by an iterative demodulation/decoding approach that significantly improves the robustness and achievable rate of the covert channel. It is worth noting that in all design scenarios, the undetectability of the covert channel is non-negotiable meaning that the channel should always achieve provable polynomial undetectability according to Definition 3.2.

5.3.1 Covert Transmitter

Figure 5.2 depicts a high level description of the proposed covert communication scheme. The covert message (i.e., $\mathbf{m}$) is encoded by a channel encoder to form the covert codeword of length n (i.e., $\mathbf{u} = (u_1, u_2, ..., u_n)$). Then, the covert codeword is sent to the modulator block that maps the covert codeword into the inter-packet delays of the covert traffic. In order to achieve provable polynomial undetectability, the elements of the covert codeword (i.e., $u_i,\ i \in [n]$) should form a sequence of $(i.i.d.)$ random bits with equal probability for covert bit-0 and covert bit-1. To achieve such a property, a random interleaver is used to break correlation between the elements of the covert codeword. The covert transmitter uses the generated IPDs in order to form the covert traffic that is sent to the covert receiver.

Encoding: The encoder block at the covert transmitter may consist of a single channel code (e.g., a convolutional code) or a concatenated channel code configuration (e.g., a Reed-Solomon code as the inner code and a convolutional code as the outer code) [51]. A

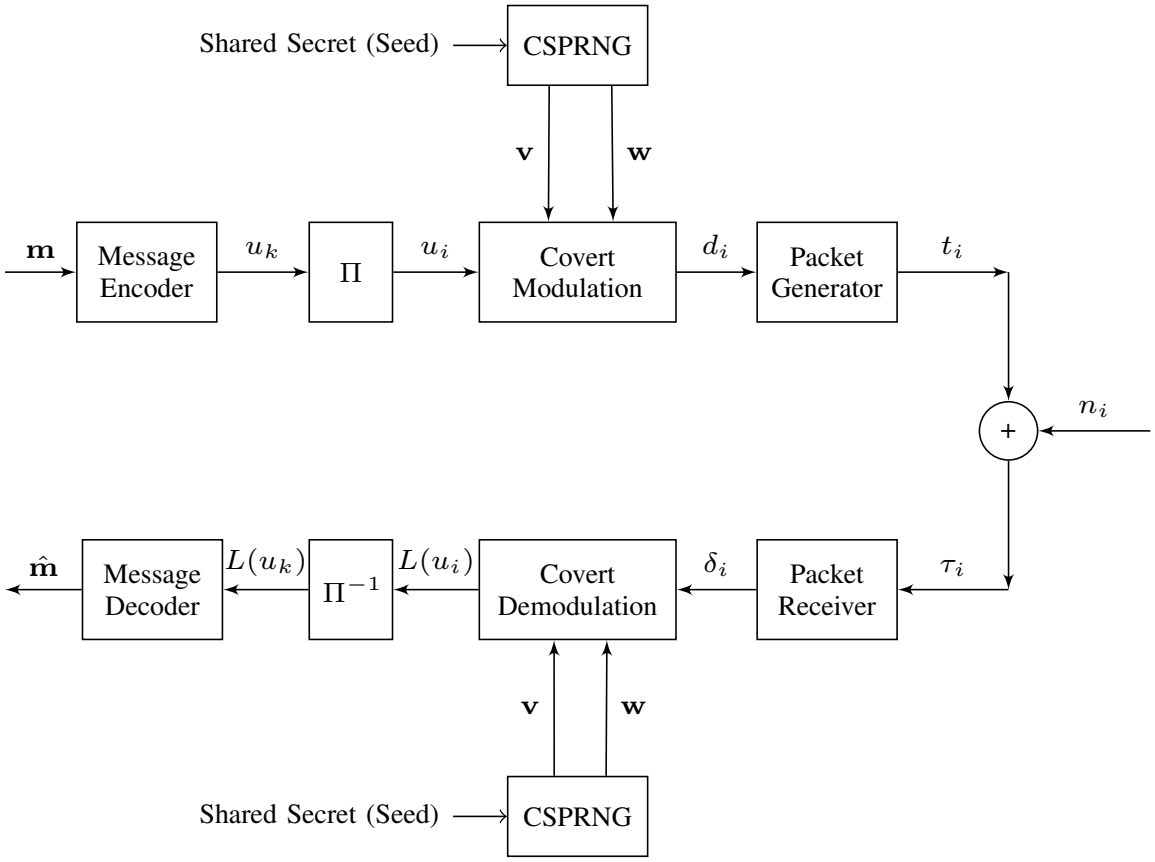


Figure 5.2: High level system model of the turbo covert channel.

rate $r = \frac{k}{n}, k \leq n$ channel code accepts an input of size k, and generates a codeword of length n by adding extra redundancy information to the original message. In this way, the receiver can tolerate some few errors in the received codeword and still recover the message. Although using the concatenated codes would significantly improve the reliability of the system, it is decided to keep the system complexity at minimum with a single convoloutional code as the encoding block. It is noted that the rate of the channel code plays a major role in robustness and the achievable covert rate of the covert channel. In other words, increasing the code rate improves the covert rate at the expense of higher bit error rate at the covert receiver and vice versa.

Covert Message Modulation: The modulation block is designed to map the covert codeword into the IPDs that form the covert traffic. Let $\mathcal{F}(.)$ be the cumulative distribution function (i.e., CDF) of the IPDs of the legitimate traffic of the channel. The key in model-based covert channels is to design a modulation scheme such that the covert traffic mimics the same statistical characteristics as the ones of the legitimate traffic of the channel. To this end, one can apply the application of *Monte Carlo sampling* and *quantile function* in generating a sequence of random numbers that are distributed according to the desired distribution function (e.g., $\mathcal{F}(.)$) [98].

The construction of Monte Carlo samples for a given distribution function is facilitated using the quantile function of the distribution. In brief, the quantile function of a probability distribution is the inverse of the cumulative distribution function (i.e., $\mathcal{F}^{-1}(.)$). Having the knowledge of the quantile function, one can generate a sequence of random numbers according to the desired distribution by driving the quantile function with a sequence of uniformly distributed random numbers. For instance, the sequence of random delays $\mathbf{d} = \{d_1, d_2, ..., d_n\}$ will have the same distribution as the IPDs of the legitimate traffic of the channel if the elements of $\mathbf{d}$ are generated as follows:

$$d_i = \mathcal{F}^{-1}(a_i),\ i \in [n] \tag{5.1}$$

Where, $\mathcal{F}(.)$ is the CDF of the IPDs of the legitimate traffic of the channel, and $\mathbf{a} = \{a_1, a_2, ..., a_n\}$ is a sequence of uniformly distributed random numbers (i.e., $\mathbf{a} \sim \mathcal{U}(0, 1)$).

The aforementioned method in generating sequences of random numbers can be used in order to modulate the covert message over IPDs of the covert traffic. To this end, the covert message is masked with a sequence of uniformly distributed random numbers (i.e., $\mathbf{v} \sim \mathcal{U}(0, 1)$). The masking operation is performed using the addition modulo 1 operation. Hence, we have:

$$a_i \equiv v_i + u_i \pmod 1,\ i \in [n] \tag{5.2}$$

It is noted that given the assumption that the elements of $\mathbf{u}$ are (*i.i.d.*) equiprobable random bits, the result of the masking operation (i.e., $\mathbf{a} = \{a_1, a_2, ..., a_n\}$) is in fact a

sequence of uniformly distributed random numbers (i.e., $\mathbf{a} \sim \mathcal{U}(0,1)$) [43]. Thus, one can pass the elements of the masked parameter (i.e., $a_i,\ i \in [n]$) to the quantile function of the IPDs of the legitimate traffic (i.e., $\mathcal{F}^{-1}(.)$) in order to generate a sequence of inter-packet delays that has the same distribution function as the IPDs of the legitimate traffic.

Alternatively, It is possible to map the elements of the covert codeword (i.e., $\mathbf{u} = (u_1, u_2, ..., u_n)$) directly into a sequence of uniformly distributed random numbers (i.e., $\mathbf{a} = (a_1, a_2, ..., a_n)$) as follows:

$$a_i = \frac{u_i + v_i}{2}, \qquad \forall i \in [n]. \tag{5.3}$$

Where, $v_i \sim \mathcal{U}(0,1)$. It is easy to verify that if the elements of the covert codeword (i.e., u_i) are (*i.i.d.*) and equiprobable random bits, the result of Equation (5.3) (i.e., $\mathbf{a} = \{a_1, a_2, ..., a_n\}$) forms a set of uniformly distributed random numbers. Therefore, one can pass the generated sequence of random numbers as input to the quantile function in order to form the IPDs of the covert traffic (i.e., $d_i = \mathcal{F}^{-1}(a_i)$).

Figure 5.3 depicts an example of the covert modulation process. According to the figure, the domain of the input to the quantile function (i.e., the Y-axis in the plot) is divided in two sections with the border at $\mathcal{F}(d) = \frac{1}{2}$. The projection of these two sections over the CDF of the legitimate traffic (i.e., $\mathcal{F}(.)$), defines the range of inter-packet delays that are used to modulate covert bit-0, and covert bit-1. Clearly, the delay ranges are not equal in size, however they are equiprobable considering the distribution function of the legitimate traffic. The objective of the uniformly distributed random numbers (i.e., $v_i \in \mathbf{v}$) is to sweep the domain of the quantile function with equal probability while the elements of the covert codeword define the range of the IPDs that are allowed to be selected. The result is a sequence of random delays that are distributed according to the distribution of the legitimate traffic, yet they carry the embedded covert message as well.

The aforementioned covert modulation strategy treats each element of the covert codeword independently. However, using a trellis structure in modulating the covert codeword, it is possible to design a covert modulation scheme that embeds information about not only an individual element of the covert codeword, but also defines the complete structure

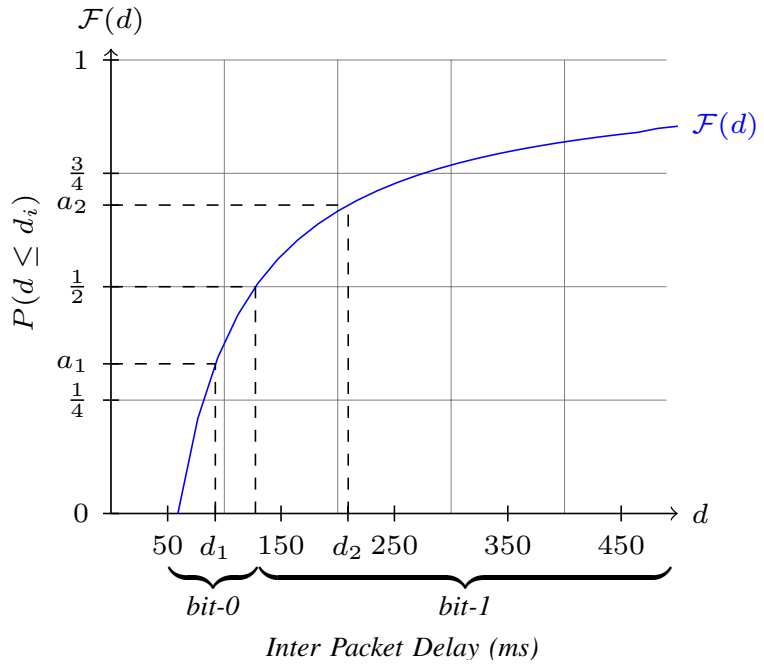


Figure 5.3: Modulating the covert message $\mathbf{u} = (0, 1)$ given the random sequence vector $\mathbf{v} = (0.33, 0.18)$.

of the transmitted message at the covert receiver. In this way, correct detection of some of the inter-packet delays at the covert receiver would be beneficial in detecting other samples of the covert traffic that are affected by channel noise or adversarial disruptions.

Consider two independent sequences of $(i.i.d.)$ random numbers $\mathbf{v} = (v_1, v_2, ..., v_n)$, and $\mathbf{w} = (w_1, w_2, ..., w_n)$, where the elements of each sequence are drawn according to the uniform distribution (i.e., $\mathbf{v}, \mathbf{w} \sim \mathcal{U}(0, 1)$). Given the covert codeword $\mathbf{u}$, for each element of the covert codeword (i.e., u_i) define s_i as follows:

$$s_i = u_i \oplus s_{i-1}, \quad i \in [n], \ s_0 = 0. \tag{5.4}$$

Where, $\oplus$ denotes the bit-wise XOR operation. It is noted that since the elements of the covert codeword (i.e., $u_i, \ i \in [n]$) are $(i.i.d.)$ random bits with equal probability for the covert bit-0 and the covert bit-1, the result of Equation (5.4) (i.e., $\mathbf{s} = (s_1, s_2, ..., s_n)$) is also a sequence of $(i.i.d.)$ equiprobable random bits (i.e., $P(s_i = 1) = \frac{1}{2} = P(s_i = 0), \ i \in [n]$).

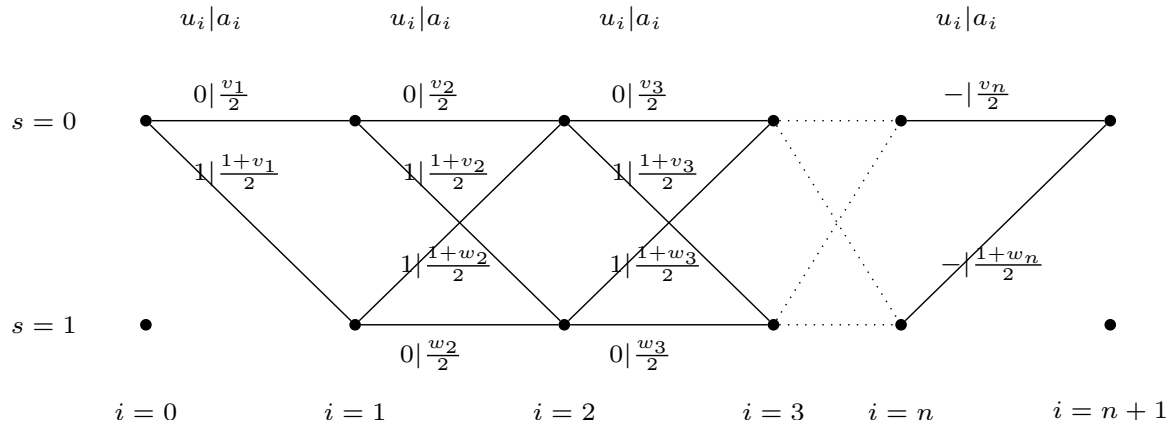


Figure 5.4: Modulation trellis with the branch indicator parameter for each branch.

Thus, one can define the branch indicator parameter (i.e., $a_i \in [0,1]$) that maps the i^{th} element of the covert codeword (i.e., u_i) to a unique branch of the modulation trellis as follows:

$$a_i = \begin{cases} \frac{u_i+v_i}{2} & s_{i-1}=0 \\ \frac{u_i+w_i}{2} & s_{i-1}=1 \end{cases}, \quad i=1,2,3,...,n \tag{5.5}$$

Figure 5.4 depicts the modulation trellis and the corresponding branch indicator parameter (i.e., a_i) in each branch of the trellis. It is noted that in order to uniquely identify each branch of the trellis, two independent sequences of random numbers (i.e., $\mathbf{v}$, $\mathbf{w}$) are required as the number of the trellis branches are twice the number of the states of the trellis. Given the binary elements of $\mathbf{u}$, and the uniformly distributed random sequences $\mathbf{v}$ and $\mathbf{w}$, it can be observed that $a_i \in [0,1],\ \forall\, i \in [n]$. The trellis starts at state 0 (i.e., $s_0 = 0$) and has $n+2$ stages, one for the initial stage (i.e., the initial packet at the covert traffic), one for the final transition back to state 0 (i.e., $s_{n+1} = 0$), and the rest for the modulation of the elements of the covert codeword.

Using the branch indicator parameter and the quantile function of the legitimate traffic (i.e., $\mathcal{F}^{-1}(.)$), the modulation scheme transforms the elements of the covert codeword into their corresponding IPD values (i.e., d_i) as follows:

$$d_i = \mathcal{F}^{-1}(a_i), \quad i=1,2,...,n \tag{5.6}$$

Thus, the transmission time of the packets of the covert traffic can be written as follows:

$$t_j = \sum_{i=1}^{j} d_i + t_0, \quad j = 1, 2, ..., n \tag{5.7}$$

Where, t_0 marks the transmission instance of the initial packet in the flow.

5.3.2 Covert Receiver

The arrival time of the packets of the covert traffic (i.e., τ) is the only channel signal observable to the covert receiver. However, the packet arrival time is subject to the channel noise (e.g., network jitter or adversarial jamming). Hence,

$$\tau_i = t_i + n_i, \quad \forall\, i \in [n] \tag{5.8}$$

Where, n_i is the noise affecting the arrival time of the i^{th} packet of the covert traffic. In order to decode the covert message, the receiver is required to derive the inter-packet delays of the received covert traffic (i.e., $\delta_i,\ i \in [n]$). Hence,

$$\begin{aligned} \delta_i &= \tau_i - \tau_{i-1} \\ &= t_i + n_i - (t_{i-1} + n_{i-1}) \\ &= d_i + n_i - n_{i-1} = d_i + z_i \end{aligned} \tag{5.9}$$

Where, $\tau_0 = 0$, and $z_i = n_i - n_{i-1}$ is the equivalent of the channel noise on the i^{th} element of the sequence of received IPDs (i.e., $\delta = (\delta_1, \delta_2, ...\delta_n)$).

Demodulation: Given the sequence of the received IPDs, the demodulation objective is to find the most probable covert codeword that is likely to be transmitted by the covert transmitter. In other word, the demodulation task is to compute $P(u_i|\delta)$ or equivalently the log likelihood ratio (i.e., LLR) of the probability function, which is defined as follows :

$$L(u_i) = \log \frac{P(u_i = 1|\delta)}{P(u_i = 0|\delta)}, \quad \forall\, i \in [n] \tag{5.10}$$

To this end, using a similar trellis structure as the one used by the covert transmitter, the receiver performs the *maximum a-posteriori probability (MAP)* demodulation which is based on a recursive algorithm proposed by Bahl *et al.* [99].

During the modulation process, the transition from each stage of the encoding trellis to the next stage is selected according to the branch indicator parameter (i.e., a_i) that depends on the elements of the covert codeword (i.e., u_i). Thus, stage transition probabilities are considered as intermediate quantities in order to compute the likelihood of the elements of the covert codeword, given the detected IPDs of the covert traffic at the covert receiver (i.e., $P(u_i|\delta)$). Let S_i denote the state of the modulation trellis at stage i, and define $\sigma_i(s', s)$ as follows:

$$\sigma_i(s', s) = P(S_{i-1} = s', S_i = s, \delta) \tag{5.11}$$

In fact, $\sigma_i(s', s)$ represents the joint probability of two individual events. The first event denotes the transition from state s' in stage $i-1$, to state s in the next stage of the modulation trellis (i.e., $S_{i-1} = s$), when the covert transmitter is calculating the i^{th} inter-packet delay for the covert traffic. Meanwhile, the second event indicates the detection of $\delta = \{\delta_1, \delta_2, ..., \delta_n\}$ as the sequence of received IPDs at the covert receiver. Using the definition of $\sigma_i(s', s)$, Equation (5.10) can be rewritten as follows:

$$\begin{aligned}
L(u_i) &= \log \frac{P(u_i = 1|\delta)}{P(u_i = 0|\delta)} \\
&= \log \frac{P(u_i = 1, \delta)P(\delta)}{P(u_i = 0, \delta)P(\delta)} \\
&= \log \frac{\sum_{(s',s):u_i=1} P(S_{i-1} = s', S_i = s, \delta)}{\sum_{s',s:u_i=0} P(S_{i-1} = s', S_i = s, \delta)} \\
&= \log \frac{\sum_{(s',s):u_i=1)} \sigma_i(s', s)}{\sum_{(s',s):u_i=0} \sigma_i(s', s)}
\end{aligned} \tag{5.12}$$

In other words, instead of calculating $P(u_i|\delta)$, one can evaluate $\sigma_i(s', s)$ for all branch transitions that are associated to $u_i = 1$ (i.e., $(s', s) : u_i = 1$), and those transitions that are

taken if $u_i = 0$ (i.e., $(s', s) : u_i = 0$), then calculate the LLR value (i.e., $L(u_i)$) according to Equation (5.12). Furthermore, using the chain rule for conditional probability [98], Equation (5.11) is rewritten as follows:

$$\sigma_i(s', s) = \underbrace{P(S_{i-1} = s', \delta_1^{i-1})}_{\alpha_{i-1}(s')} \times \underbrace{P(S_i = s, \delta_i | S_{i-1} = s', \delta_1^{i-1})}_{\gamma_i(s', s)} \times \underbrace{P(\delta_{i+1}^{n} | S_i = s, S_{i-1} = s', \delta_1^{i})}_{\beta_i(s)}$$

$$= \alpha_{i-1}(s')\ \gamma_i(s', s)\ \beta_i(s) \tag{5.13}$$

Where, $\alpha_i(s)$ is derived recursively as follows:

$$\begin{aligned}
\alpha_i(s) &= P(S_i = s, \delta_1^i) \\
&= \sum_{s'} P(S_{i-1} = s', S_i = s, \delta_1^i) \\
&= \sum_{s'} P(S_{i-1} = s', \delta_1^{i-1}) P(S_i = s, \delta_i | S_{i-1} = s', \delta_1^{i-1}) \\
&= \sum_{s'} \alpha_{i-1}(s') \gamma_i(s', s), \qquad \alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases}
\end{aligned} \tag{5.14}$$

Similarly, for $\beta_i(s)$ we have:

$$\beta_i(s) = \sum_{s'} \beta_{i+1}(s') \gamma_{i+1}(s, s'), \qquad \beta_{n+1}(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases} \tag{5.15}$$

The initialization of the forward recursion (i.e., $\alpha_0(s)$) is straightforward as the trellis is assumed to start at state $s = 0$. Meanwhile, the backward recursion (i.e., $\beta_{n+1}(s)$) is initialized using the starting state of the next block. Figure 5.5 depicts the trellis structure at the covert receiver where each branch is marked with its corresponding branch transition weight (i.e., $\gamma_i(s', s)$).

Given the recursive formulas for $\alpha_i(s)$ and $\beta_i(s)$, it all comes to calculate the branch transition weights (i.e., $\gamma_i(s', s),\ i \in [n]$) in order to derive the LLR of the elements of the covert codeword. We have,

$$\gamma_i(s', s) = P(S_i = s, \delta_i | S_{i-1} = s', \delta_1^{i-1})$$

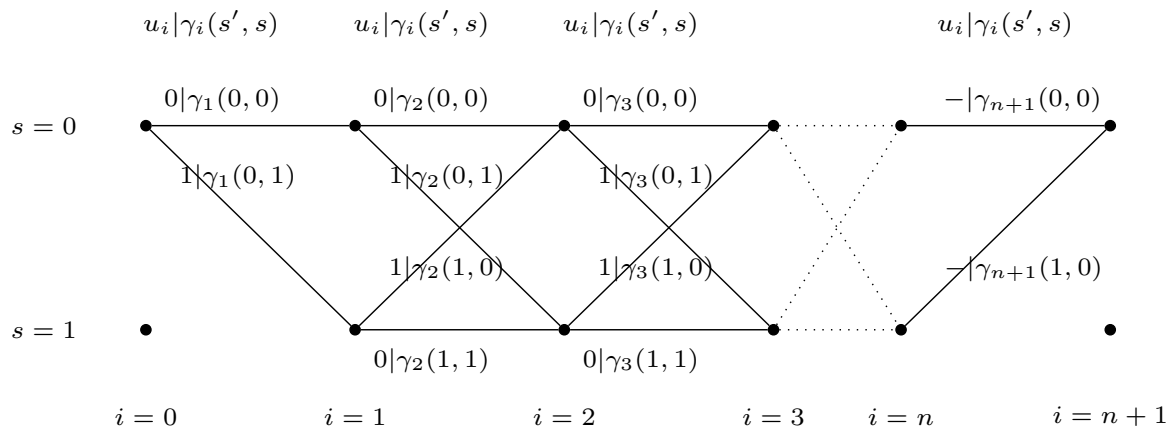


Figure 5.5: Demodulation trellis with branch transition weights.

$$= \sum_j P(S_i = s, \delta_i, u_i = j | S_{i-1} = s', \delta_1^{i-1})$$

$$= \sum_j P(u_i = j | S_{i-1} = s', S_i = s, \delta_i, \delta_1^{i-1}) \times P(S_i = s | S_{i-1} = s', \delta_1^{i-1})$$

$$\times P(\delta_i | S_{i-1} = s', S_i = s, \delta_1^{i-1}) \tag{5.16}$$

The first term in Equation (5.16) is either 1 or 0 depending on whether the transition from state $S_{i-1} = s'$ to state $S_i = s$ is associated to $u_i = j$ or not. The second term is the initial *a priori* information about the covert codeword at the covert receiver. That is, if the covert receiver knows that the input bit causing the transition from s' to s happens with certain probability. It is noted that since the elements of the covert message are equiprobable and mutually independent, this term is constant regardless of the stage or the state of the trellis. Thus, it is canceled out in the LLR formulation.

The last term of Equation (5.16) represents the probability of receiving δ_i, assuming the transition from s' to s took place at the modulation trellis. Let $a_i(s', s)$ denote the branch indicator parameter that is associated to the state transition (s', s), according to Equation (5.5). Hence, one can interpret the last term of Equation (5.16) as the probability of the event in which the i^{th} IPD was the hypothesis $\hat{d}_i = \mathcal{F}^{-1}(a_i(s', s))$, yet the receiver observes

δ_i due to the channel noise. Thus, the final term of Equation (5.16) can be written as follows:

$$P(\delta_i|S_i = s, S_{i-1} = s', \delta_1^{i-1}) = P(\delta_i|\hat{d}_i)$$
$$= P(z_i = \delta_i - \hat{d}_i) \tag{5.17}$$

Where, $z_i = n_i - n_{i-1}$ is the equivalent channel noise on the i^{th} IPD. Let $f_n(n)$ be the probability distribution function of the channel noise (i.e., n) that affects the packet arrival times of the covert traffic. Using the convolution formula for the distribution of sum of two random variables [84] we have:

$$f_z(z) = \int_n f_n(z+n) f_n(n)\, dn \tag{5.18}$$

For instance, if n is uniformly distributed in the range $[0, \Delta]$ (i.e., $n \sim \mathcal{U}(0, \Delta)$), the distribution of the IPD equivalent noise (i.e., $f_z(z)$) can be derived as follows:

$$f_z(z) = \begin{cases} \frac{\Delta - |z|}{\Delta^2} & -\Delta \leq z \leq \Delta \\ 0 & otherwise \end{cases} \tag{5.19}$$

Similarly, if the channel noise has a zero mean normal distribution with variance σ^2 (i.e., $n \sim \mathcal{N}(0, \sigma^2)$), the equivalent channel noise on the inter-packet delay of the received traffic has a normal distribution with zero mean and variance of $2\sigma^2$ (i.e., $z \sim \mathcal{N}(0, 2\sigma^2)$). Combining Equations (5.12),(5.13),(5.16), and (5.17) the log-likelihood ratio of the i^{th} element of the covert codeword at the covert receiver can be written as follows:

$$L(u_i) = \log \frac{\sum_{(s',s):u_i=1} \alpha_{i-1}(s')\ f_z(\delta_i - \hat{d}_i)\ \beta_i(s)}{\sum_{(s',s):u_i=0} \alpha_{i-1}(s')\ f_z(\delta_i - \hat{d}_i)\ \beta_i(s)}$$

$$= \log \frac{\sum_{s',s:u_i=1} \alpha_{i-1}(s') f_z\big(\delta_i - \mathcal{F}^{-1}(a_i(s',s))\big)\beta_i(s)}{\sum_{s',s:u_i=0} \alpha_{i-1}(s') f_z\big(\delta_i - \mathcal{F}^{-1}(a_i(s',s))\big)\beta_i(s)} \tag{5.20}$$

Where, $a_i(s', s)$ is the branch indicator parameter that is associated to the transition (s', s) (i.e., Figure 5.4). It is easy to verify that all the elements of Equation (5.20) are

available, or can be derived recursively at the covert receiver. At the end of the demodulation process, the computed LLR values are fed into the de-interleaver block and the result is sent to the decoder to decode the covert message.

Decoding: The decoding module is a simple Soft-Input Soft-Output (SISO) decoder [56]. In more detail, the decoding module accepts the LLR values of the elements of the covert codeword (i.e., the output of the demodulation block) as input, and returns the decoder's estimation of the covert message (i.e., $\hat{\mathbf{m}}$). The decoder block works based on a similar principle as the one that is described for the demodulation block. The only difference lies in the fact that the decoder uses a different trellis structure. In fact, the trellis of the decoder is selected according to the channel code that is used at the covert transmitter (e.g., the rate 1/2 Voyager code).

5.3.3 Iterative Demodulation/Decoding

In a non-iterative demodulation/decoding configuration (i.e., Figure5.2), each block of the covert receiver, that is involved in decoding the covert message, benefits from the information that is gleaned about the covert codeword at the previous blocks (e.g., the decoder uses LLRs from the demodulator). However, the information flow is unidirectional and does not run in the reverse direction. This lack of cooperation limits the decoding performance at the covert receiver and reduces the robustness of the covert channel. One way to tackle this problem is to use iterative demodulation/decoding. In this way, the belief of the decoding block on the correct covert codeword elements is propagated back to the demodulation module for another round of demodulation/decoding. Thus, the decision of each block is enhanced by extra information that is received from all other blocks involved in the demodulation/decoding process at the covert receiver.

Figure 5.6 depicts the covert receiver block diagram that is modified to perform the iterative demodulation/decoding. In this setup, the combination of demodulation, de-interleaver, and decoder blocks is envisioned as a serially concatenated coding system in which the outer code is the convolutional code while the demodulation block acts as the inner code [100]. In fact, since the proposed demodulation block and the decoding block

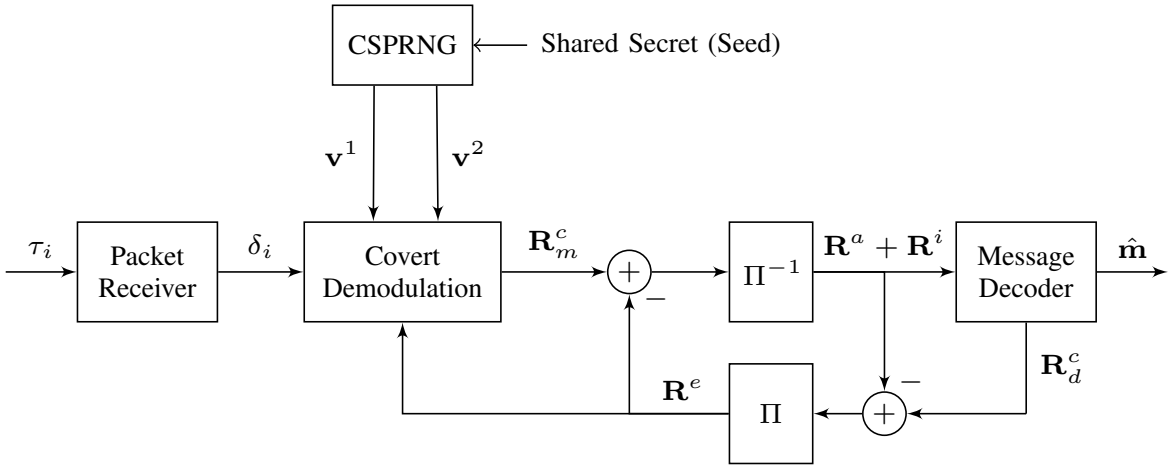


Figure 5.6: Covert receiver block diagram with iterative demodulation/decoding.

produce soft-decision outputs (i.e., the log-likelihood ratios of the elements of the covert codeword), they are very well suited for an iterative demodulation/decoding configuration.

Let's begin with the demodulation block. The output of the demodulator is the log-likelihood ratio of the elements of the covert codeword. To this end, the demodulator uses the received IPDs (i.e, δ), the initial *a priori* information on the elements of the covert codeword (i.e., R_i^a), and its knowledge about the channel to calculate the *a posteriori* probability of the elements of the covert codeword (i.e., $L(u_i)$). Thus, the output of the demodulator can be written as follows:

$$\begin{aligned} L(u_i) &= \log \frac{P(u_i = 1|\delta)}{P(u_i = 0|\delta)} \\ &= \log \frac{P(\delta|u_i = 1)}{P(\delta|u_i = 0)} + \log \frac{P(u_i = 1)}{P(u_i = 0)} \\ &= R_i^i + R_i^a \end{aligned} \tag{5.21}$$

The first term in Equation (5.21) is called the *intrinsic* information (i.e., R_i^i). The intrinsic information denotes the information that is gleaned about the i^{th} element of

the covert codeword (i.e., u_i) from the input signal (i.e., δ), and perhaps the *a priori* information on other elements of the covert codeword at the demodulation block (i.e., $u_j,\ j \neq i$) [101]. The second term is the log-likelihood ratio of the *a priori* information on u_i. It is noted that the elements of the covert codeword are considered to be (*i.i.d.*) equiprobable random bits due to the use of a random permutation block at the covert transmitter. Thus, there would be no initial *a priori* information about the elements of the covert codeword to the modulation block. For clarity of description, lets rewrite Equation (5.21) in vector format for all elements of the covert codeword. Hence,

$$\mathbf{R}_m^c = \mathbf{R}^i + \mathbf{R}^a \tag{5.22}$$

Where, $\mathbf{R}_m^c$ is the composite log-likelihood ratio at the output of the demodulator block. The result of the demodulation block is passed through a de-interleaver and then sent to the decoder block as the *a priori* information about the received covert codeword. Hence, the input to the decoder is the combination of the *a priori* information (i.e., $\mathbf{R}^a$) and the *intrinsic* information of the demodulation block (i.e., $\mathbf{R}^i$). The decoder uses this information (i) to produce an estimation of the covert message (i.e., $\hat{\mathbf{m}}$), and (ii) to compute the decoder's version of the log-likelihood ratio of the elements of the covert codeword (i.e., $\mathbf{R}_d^c$). Thus, we have:

$$\mathbf{R}_d^c = \mathbf{R}^i + \mathbf{R}^a + \mathbf{R}^e \tag{5.23}$$

Where, $\mathbf{R}_d^c$ is the decoder's composite log-likelihood ratio at the end of the first iteration round. $\mathbf{R}^e$ is called the *extrinsic* information [101], and represents the extra information added by the decoding block about the elements of the covert codeword. It is worth noting that Equations (5.23) is valid only if inputs to the decoder are independent. Otherwise, the log likelihood ratios can not be represented as separate parameters. This independence is achieved by the interleaver/de-interleaver blocks between the demodulator and the decoding block.

The result of the first iteration is returned to the modulation block as the *a priori* information for the second round. However, in order for the iterative processing to converge

to the correct answer, *the information used as the a priori information for each block should not contain the information that has been generated by that block in the previous round* [102]. For instance, the *a priori* information to the demodulation block for the second iteration round consists of the original *a priori* information (i.e., $\mathbf{R}^a$) and the extrinsic information from the decoding block (i.e., $\mathbf{R}^e$), yet it does not include the *intrinsic* information which was generated by the modulation block in the previous iteration (i.e., $\mathbf{R}^i$). To this end, the output of the decoder has to be subtracted from the results of the demodulation block before it can be used as the input to the demodulation block for the next iteration round (i.e., Figure 5.6). The same argument holds for the decoding block. Therefore, the input to the decoder should be the combination of the original *a priori* information and the *intrinsic* information from the modulation block. However, the *extrinsic* information of the previous round should be removed from the decoder's input.

5.3.4 Guard Band

The iterative demodulation/decoding technique significantly enhances the receiver's capability to correctly recover the covert message. However, in-depth analysis of the proposed scheme reveals that it is possible to improve the reliability of the covert channel even further by means that are provided by the flexible design of the modulation scheme. Timing covert channels in communication networks are usually designed based on embedding the covert message in the IPDs of the covert traffic. Hence, the error rate at the covert receiver is significantly reduced if the inter-packet delays that represent bit-0 and bit-1 are easily distinguishable.

Consider the model-based covert modulation of Figure 5.7, where small delays represent bit-0 and large delays represent bit-1. In order to preserve the stealthiness property of the covert channel, the covert transmitter has to select all possible IPDs according to the distribution pattern of the legitimate traffic. However, for those IPDs that are close to the domain deliminator line (i.e., $d = \mathcal{F}^{-1}(\frac{1}{2})$), the probability of detecting the received IPD at the wrong side of the threshold line (due to the channel noise) increases significantly. Such an event would dramatically reduce the decoding performance at the covert receiver.

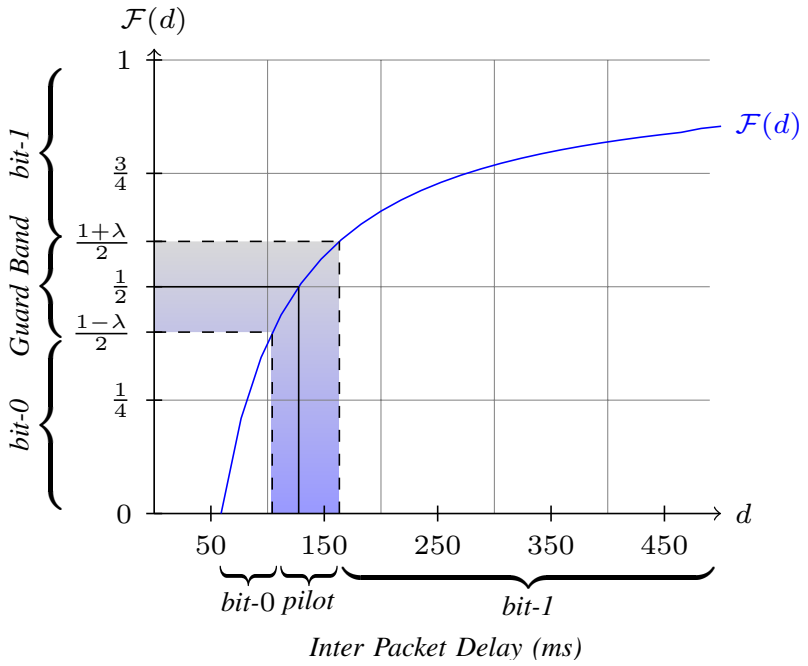


Figure 5.7: Model-based covert modulation with guard band.

To solve this problem it is suggested to consider a guard band around the domain deliminator line (i.e., the shaded region in Figure 5.7), and let the transmitter ignore the guard band during the modulation process [47]. Unfortunately, this approach forces the pattern of the inter-packet delays of the covert traffic to deviate from the pattern of the legitimate traffic which can be detected by a system observer. In fact, it can be shown that the negative effect of increasing the size of the guard band (i.e., λ) on the stealthiness property of the channel is much more severe than its contribution to the robustness of the covert channel.

In order to prevent the negative effects of using the guard band on the covertness of the channel, we introduce pilot symbols in the design of the turbo covert channel. In this way, every time the modulation block needs to modulate a message with an IPD in the guard band region, it uses a *pilot symbol* instead of an information symbol that carries covert information. The pilot symbol is known at the covert transmitter and the covert receiver, hence decoding the pilot symbol is guaranteed to be error-free at the covert receiver. To

this end, the covert transmitter splits the domain of input to the quantile function (i.e., the *Y-axis* in Figure 5.7) into three distinct regions. One region is for the pilot symbols (i.e., the guard band) and the other two are associated to modulate covert bit-0 and covert bit-1. Therefore, the modulation process has to be modified by redefining the branch indicator parameter (i.e., a_i) as follows:

$$a_i = \begin{cases} \frac{u_{k_i}+v_{k_i}+\lambda(u_{k_i}-v_{k_i})}{2} & s_{i-1}=0,\ g_i \geq \lambda \\ \frac{u_{k_i}+w_{k_i}+\lambda(u_{k_i}-w_{k_i})}{2} & s_{i-1}=1,\ g_i \geq \lambda \\ \frac{1-\lambda}{2}+g_i & g_i < \lambda \end{cases} \tag{5.24}$$

Where, g_i is a uniformly distributed random number (i.e., $g_i \sim \mathcal{U}(0,1)$) that is generated by a CSPRNG. The value of g_i controls the transmitter's decision on whether to include a pilot symbol as the i^{th} symbol of the covert traffic or modulate an information symbol instead. $k_i \in [n]$, is the index of the next information bit that is expected to be transmitted by the covert transmitter and it is calculated as follows:

$$k_i = \begin{cases} k_{i-1}+1 & g_i \geq \lambda \\ k_{i-1} & g_i < \lambda \end{cases}, \quad i \in [n],\ k_0 = 0 \tag{5.25}$$

The state parameter (i.e., s_i) is also redefined as follows:

$$s_i = \begin{cases} u_{k_i} \oplus s_{i-1} & g_i \geq \lambda \\ 1 & \frac{\lambda}{2} \leq g_i < \lambda \\ 0 & 0 \leq g_i < \frac{\lambda}{2} \end{cases}, \quad s_0 = 0 \tag{5.26}$$

It is noted that when $g_i < \lambda$ (i.e., pilot symbol modulation), a_i and s_i depend only on g_i. Therefore, similar to the discussion on Equation (5.4), $\mathbf{s} = (s_1, s_2, ...)$ forms a sequence of (*i.i.d.*) equiprobable random bits. To be more precise, when $g_i < \lambda$, the value of the corresponding state parameter (i.e., s_i) would be 0 or 1 if g_i falls in the lower half (i.e., $0 \leq g_i < \frac{\lambda}{2}$) or the higher half of the guard band (i.e., $\frac{\lambda}{2} \leq g_i < \lambda$), respectively. Therefore, one can realize the probability of the state parameter in this case as follows:

$$P(s_i = 0 | g_i < \lambda) = P(s_i = 1 | g_i < \lambda) = \frac{1}{2} \tag{5.27}$$

On the other hand, when the covert transmitter is modulating the covert codeword (i.e., $g_i \geq \lambda$), the state parameter (i.e., s_i) is defined by combining the corresponding covert codeword (i.e., u_i) and the previous state of the encoder trellis (i.e., s_{i-1}). However, since the elements of the covert codeword are (*i.i.d.*) equiprobable random bits, the value of the state parameter (i.e., s_i) can be either 0 or 1 with equal probability as well. Therefore,

$$P(s_i = 0|g_i \geq \lambda) = P(s_i = 1|g_i \geq \lambda) = \frac{1}{2} \tag{5.28}$$

Combining Equations (5.27) and (5.28) for any $j \in \{0, 1\}$, we have:

$$\begin{aligned} P(s_i = j) &= \sum_{g_i} P(s_i = j|g_i)P(g_i) \\ &= \lambda P(s_i = j|g_i < \lambda) + (1 - \lambda)P(s_i = j|g_i \geq \lambda) \\ &= \frac{1}{2} \end{aligned} \tag{5.29}$$

By sharing $\mathbf{g} = (g_1, g_2, ...)$ between the covert transmitter and the covert receiver, the receiver gains perfect knowledge on the location, and the value of the pilot symbols, regardless of the channel condition and the covert message. It is worth noting that sharing the random vector $\mathbf{g}$ does not imply any additional constraint on the system design. In fact, the covert transmitter and the covert receiver already have a shared secret that can be used to generate sequences of independent random numbers.

In addition to increase the minimum distance between IPDs that reflect covert bit-0 with those that represent covert bit-1, pilot symbols can be used to improve the effectiveness of the demodulation/decoding process at the covert receiver. In more detail, having perfect knowledge on pilot symbols, the receiver uses this information in order to refine its estimation of the elements of the covert codeword. In addition, the covert receiver can use the pilot symbols to update its estimation of the statistical characteristics of the channel (i.e., $f_z(z)$) which directly affects the decoding performance at the covert receiver.

5.4 Performance Analysis

The three main performance metrics of a covert communication scheme i.e., undetectability, reliability, and covert rate are the major players of the performance analysis study of the turbo covert channel. In this section, we test the ability of the TCC scheme to evade detection, provide a robust covert communication channel, and achieve high covert rate.

5.4.1 Undetectability Analysis

Hiding the communication channel without being detected is one of the main design criteria of a network covert channel. In other words, the stealthiness property of the turbo covert channel is of a great importance in analyzing the effectiveness of the proposed covert communication framework. To this end, it is shown that the TCC scheme achieves *provable polynomial undetectability* according to Definition 3.2. This means that the turbo covert channel is undetectable against any polynomial-time statistical test that analyzes the samples of the covert traffic and the legitimate traffic. A salient feature of the TCC scheme is that its covertness property is always assumed to be non-negotiable. In other words, the channel should always achieve polynomial undetectability regardless of the channel conditions (e.g., error rate) or the configuration parameters of the system (e.g., code rate, guard band size).

To show the undetectability of the TCC scheme, we first confirm that the set of branch indicator parameters (i.e., $\mathbf{a} = \{a_1, a_2, ...\}$), form a sequence of (*i.i.d.*) uniformly distributed random numbers. Then, a reduction is demonstrated that transforms the problem of detecting the covert channel into the problem of distinguishing the output of a CSPRNG from a true random sequence. This leads to a contradiction according to Definition 3.3.

Preposition 5.1. *Let* $v, w, g, \lambda \sim \mathcal{U}(0, 1)$ *be four independent uniformly distributed random variables generated by a CSPRNG according to Definition 3.3. Define* u *and* s *to be two independent binary random bits that take values* 0 *or* 1 *with equal probability. Let the*

branch indicator parameter (i.e., a) to be defined as follows:

$$a = \begin{cases} \frac{u+v+\lambda(u-v)}{2} & s = 0,\ g \geq \lambda \\ \frac{u+w+\lambda(u-w)}{2} & s = 1,\ g \geq \lambda \\ \frac{1-\lambda}{2} + g & g < \lambda \end{cases} \tag{5.30}$$

Then, a is uniformly distributed in the range $[0, 1]$*. In other words,*

$$f_a(x) = \frac{d\mathcal{F}_a(x)}{dx} = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & otherwise \end{cases} \tag{5.31}$$

Where, $f_a(.)$ *denotes the probability density function (i.e., PDF), and* $\mathcal{F}_a(.)$ *is the cumulative distribution function (i.e., CDF) of the random variable a.*

Proof. The proof comes directly from the definition of a, and the properties of u, s, v, w, g, and λ. We have,

$$\begin{aligned}
\mathcal{F}_a(x) &= P(a \leq x) \\
&= \sum_{u=0}^{1}\sum_{s=0}^{1}\big[P(a \leq x|u,s,g \geq \lambda)\ P(u,s|g \geq \lambda)\ P(g \geq \lambda) \\
&\qquad + P(a \leq x|u,s,g < \lambda)\ P(u,s|g < \lambda)\ P(g < \lambda)\big] \\
&\overset{(a)}{=} \sum_{u=0}^{1}\sum_{s=0}^{1}\big[(1-\lambda)P(u,s)P(a \leq x|u,s,g \geq \lambda) + \lambda P(u,s)\ P(a \leq x|u,s,g < \lambda)\big] \\
&\overset{(b)}{=} \sum_{u=0}^{1}\sum_{s=0}^{1}\big[\frac{(1-\lambda)}{4}P(a \leq x|u,s,g \geq \lambda) + \frac{\lambda}{4}P(a \leq x|g < \lambda)\big] \\
&= \frac{1-\lambda}{4}\sum_{u=0}^{1}\sum_{s=0}^{1}\big[P(a \leq x|u,s,g \geq \lambda)\big] + \lambda P(a \leq x|g < \lambda) \\
&\overset{(c)}{=} \frac{1-\lambda}{4}\sum_{u=0}^{1}\big[P(a \leq x|u,s=0,g \geq \lambda) + P(a \leq x|u,s=1,g \geq \lambda)\big]
\end{aligned}$$

$$
\begin{aligned}
&+ \lambda P(\frac{1-\lambda}{2} + g \leq x) \\
&= \frac{1-\lambda}{4} \sum_{u=0}^{1} \big[P(\frac{u+v+\lambda(u-v)}{2} \leq x|u) + P(\frac{u+w+\lambda(u-w)}{2} \leq x|u) \big] \\
&+ \underbrace{\lambda P(g \leq x - \frac{1-\lambda}{2})}_{P_g(x)} \\
&= \frac{1-\lambda}{4} \sum_{u=0}^{1} P(v \leq \frac{2x - u(1-\lambda)}{1-\lambda}|u) + P(w \leq \frac{2x - w(1+\lambda)}{1-\lambda}|u) + P_g(x) \\
&= \frac{1-\lambda}{4} \big(P(v \leq \frac{2x}{1-\lambda}) + P(w \leq \frac{2x}{1-\lambda}) \big) \\
&+ \frac{1-\lambda}{4} \big(P(v \leq \frac{2x-(1+\lambda)}{1-\lambda}) + P(w \leq \frac{2x-(1+\lambda)}{1-\lambda}) \big) + P_g(x) \\
&\overset{(d)}{=} \underbrace{\frac{1-\lambda}{2} P(v \leq \frac{2x}{1-\lambda})}_{P_0(x)} + \underbrace{\frac{1-\lambda}{2} P(v \leq \frac{2x-(1+\lambda)}{1-\lambda})}_{P_1(x)} + P_g(x) \\
&= P_0(x) + P_1(x) + P_g(x)
\end{aligned} \tag{5.32}
$$

Where, (a) is according to the fact that u, s, and g are independent random variables. The equality in (b) comes from the fact that u and s are $(i.i.d.)$ equiprobable random bits. From the definition of the random variable a (i.e., Equation (5.30)), one can realize that given $g < \lambda$, the random variable a depends only on g and λ. This observation forms the basis for the equality in (c). Finally, the transition in (d) is based on the fact that v and w are independent and uniformly distributed random variables (i.e., $v, w \sim \mathcal{U}(0,1)$). Hence, one can express the terms P_0, P_1, and P_g according to the following formulas respectively,

$$
P_0(x) = \begin{cases} \frac{1-\lambda}{2} & x > \frac{1-\lambda}{2} \\ x & 0 < x < \frac{1-\lambda}{2} \\ 0 & x \leq 0 \end{cases} \tag{5.33}
$$

$$P_1(x) = \begin{cases} \frac{1-\lambda}{2} & x > 1 \\ x - \frac{1+\lambda}{2} & \frac{1+\lambda}{2} < x \leq 1 \\ 0 & x \leq \frac{1+\lambda}{2} \end{cases} \tag{5.34}$$

$$P_g(x) = \begin{cases} \lambda & x > \frac{1+\lambda}{2} \\ x - \frac{1-\lambda}{2} & \frac{1-\lambda}{2} < x \leq \frac{1+\lambda}{2} \\ 0 & x \leq \frac{1-\lambda}{2} \end{cases} \tag{5.35}$$

Combining Equations (5.32), (5.33), (5.34), and (5.35) one can write $\mathcal{F}_a(x)$ as follows:

$$\mathcal{F}_a(x) = P_0(x) + P_1(x) + P_g(x) = \begin{cases} 1 & x > 1 \\ x & 0 < x < 1 \\ 0 & x \leq 0 \end{cases} \tag{5.36}$$

Hence,

$$f_a(x) = \frac{d\mathcal{F}_a(x)}{dx} = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & otherwise \end{cases} \tag{5.37}$$

□

Using the result of Preposition 5.1, we state the undetectability of the turbo covert communication scheme according to the following theorem:

Theorem 5.1. *Let $\mathcal{F}(.)$ be the CDF of inter-packet delays of a sequence of samples from legitimate traffic of a channel (i.e., $\tilde{\mathbf{d}} = \{\tilde{d}_1, \tilde{d}_2, ...\}$). Consider $\mathbf{d} = \mathcal{F}^{-1}(\mathbf{a})$ to be the sequence of IPDs of the covert traffic of the same channel that is generated according to the turbo covert communication scheme. Then, there exist no polynomial time statistical test that can distinguish $\mathbf{d}$ from any same length samples of the legitimate traffic of the network (i.e., $\tilde{\mathbf{d}}$). In other words, for any probabilistic polynomial time statistical test $\mathcal{T}(.)$, and any positive number κ, there exists a negligible function $\nu(\kappa)$ such that:*

$$|\mathcal{T}(\mathbf{d}) - \mathcal{T}(\tilde{\mathbf{d}})| = \nu(\kappa) \tag{5.38}$$

Proof. Since all elements of the random sequences that are used by the covert transmitter of the turbo covert channel (i.e., $\mathbf{v}, \mathbf{w}, \mathbf{g}$) are generated by a CSPRNG, they are mutually independent from each other. In addition, the elements of the covert codeword (i.e., $\mathbf{u}$), and the state variables (i.e., s_i) are designed to be equiprobable independent random bits. Hence, from Equation (5.24) and Preposition 5.1, it can be observed that the set of branch indicator parameters (i.e., $\mathbf{a} = (a_1, a_2, ...)$) is a sequence of (*i.i.d.*) random variables drawn according to a uniform distribution. This is analogous to the output of a CSPRNG according to Definition 3.3.

Now assume there exists a polynomial-time statistical test (i.e., $\mathcal{T}$) that is capable of distinguishing the sequence of IPDs of the covert traffic (i.e., $\mathbf{d}$) from the samples of the legitimate traffic of the network (i.e., $\tilde{\mathbf{d}}$). Hence, one can define another polynomial-time statistical test (i.e., $\mathcal{T}^* = \mathcal{T} \circ \mathcal{F}^{-1}$), that can distinguish $\mathbf{a} = \mathcal{F}(\mathbf{d})$ from a true sequence of random numbers. However, this contradicts with the fact that $\mathbf{a}$ has the properties of the output of a CSPRNG. □

In other words, the turbo covert channel achieves provable undetectability against any polynomial-time statistical test that runs on the samples of inter-packet delays of the covert traffic and the legitimate traffic of the same network.

5.4.2 Covert Rate Analysis

The covert rate of the TCC scheme is defined as the ratio of transmitted covert information bits to the number of packets in the covert traffic. Hence, the covert rate of the channel is denoted in bits/packets (i.e., bpp). From the structure of the covert transmitter (i.e., Figure 5.2) it can be observed that the covert rate of the TCC scheme is directly affected by the channel code rate (i.e., r). In fact, the higher the code rate the covert codeword contains more covert information and less redundancy which directly translates into higher covert rate for the channel. However, increasing the rate of the channel code means that the received codeword at the covert receiver contains less redundancy information to deal with possible channel noise.

The size of the guard band is another compelling factor in the achievable covert rate of the TCC scheme. In fact, the covert rate of the turbo covert channel changes proportionally to the size of the guard band. It is noted that when the guard band is in place, the covert transmitter modulates pilot symbols instead of covert information bits when it generates packets with IPDs within the guard band region. Thus, with a guard band of size λ, the covert transmitter modulates a pilot symbol with probability λ. Hence, one can write the expected achievable covert rate of the proposed turbo covert channel (i.e., ρ) as follows:

$$\rho = r(1 - \lambda) \tag{5.39}$$

Where, r is the rate of the channel code, and λ is the guard band size. It is noted that by using a more reliable modulation scheme (i.e., larger guard band) the transmitter achieves the flexibility to increase its channel code rate in order to compensate for some of the rate loss due to the transmission of pilot symbols. This is analogous to the concept of adaptive coding and modulation in digital communication [103]. In this way, the covert transmitter can use the channel code rate, and the size of the guard band as two control parameters to achieve a desired balance between the covert rate and the robustness without compromising the stealthiness property of the channel.

5.4.3 Robustness and Channel Reliability

The robustness of the turbo covert channel relies on the channel code that is selected at the covert transmitter (i.e., code rate), the modulation strategy (i.e., guard band size), and also the effort of the covert receiver on decoding the covert message (i.e., number of decoding iterations). In this section the robustness of the TCC scheme is studied in terms of decoding bit error rate (i.e., BER) at the covert receiver under different system configurations. In fact, using a variety of channel codes and guard band sizes, one can demonstrate the flexibility of the TCC scheme in adapting to channel conditions and system requirements. On the one hand, there is the setup with no guard band in the modulation block, where the reliability of the channel depends solely on the channel code and the iterative demodulation/decoding process at the receiver. On the other hand,

Table 5.1: Performance analysis system configuration parameters

System parameter	Selected value
Channel code rate	$r = \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{6}{7}$
Guard-band size	$\lambda = 0.15, 0.25$
Cumulative distribution function	$\mathcal{F}(d) = 1 - (\frac{\alpha}{d})^{\beta}$
Quantile function	$\mathcal{F}^{-1}(a) = \alpha(\frac{1}{1-a})^{\frac{1}{\beta}}$
Pareto distribution parameters	$\alpha = 49ms, \beta = 0.95$
Jitter standard deviation	$\sigma = 10ms$
Jamming noise distribution	$n \sim \mathcal{U}(0, \Delta)$
Maximum jamming noise level	$\Delta \in [0,\ 200ms]$

the transmitter may choose to use a guard band in the modulation process, and use a higher channel code rate to compensate for the rate loss due to the pilot symbols in the covert traffic. It is noted that the existence of a channel code is necessary for the iterative demodulation and decoding process at the covert receiver. Table 5.1 summarizes the parameters of the system configurations that are used for the performance analysis of the proposed turbo covert communication scheme.

The channel code in use is a rate 1/2 convolutional code with the generator matrix [133; 171]. Given this mother code, one can puncture the original covert codeword and generate a wide variety of channel codes with different code rates [51]. It is noted that combining a second stage channel code (e.g., a Reed-Solomon code [104]) with the first stage convoloutional encoder would significantly improve the decoding performance of the covert receiver, yet it is decided to leave the system complexity at minimum with a single encoder block.

To model the legitimate traffic of the channel, samples of about 10^6 SSH packets are extracted from CAIDA traffic traces [97]. These samples represent the traffic pattern of shell sessions that are used as the legitimate traffic of the channel. In this way, the CDF of the legitimate traffic is derived according to Pareto distribution as $\mathcal{F}(d) = 1 - (\frac{\alpha}{d})^{\beta}$, where $\alpha = 49ms$ is the scale parameter and $\beta = 0.95$ is the shift parameter of the distribution.

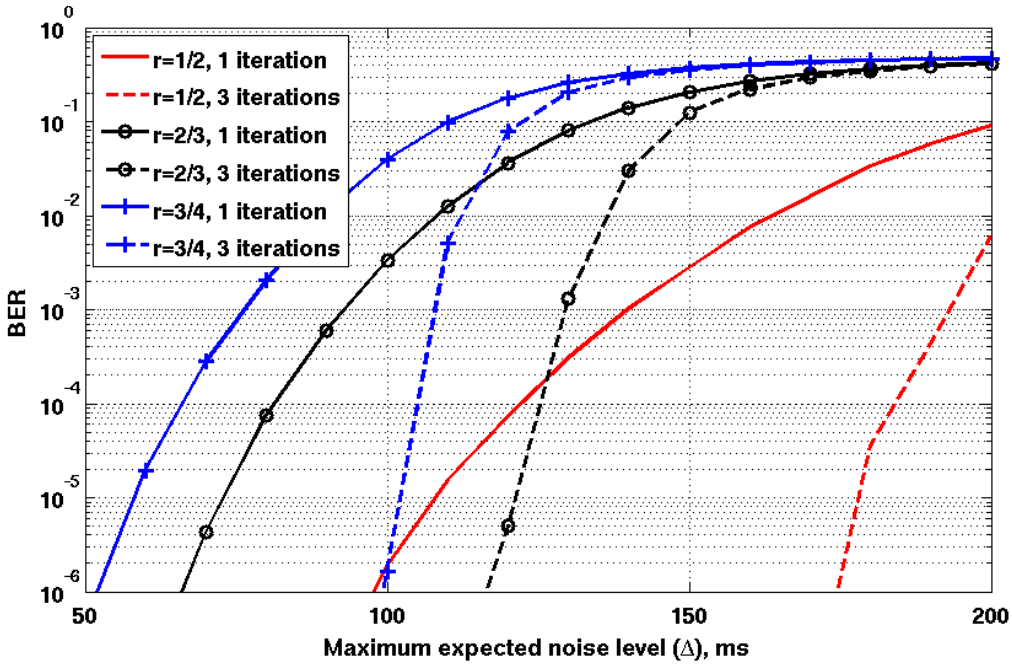


Figure 5.8: BER performance of the TCC scheme with no guard band.

The channel noise consists of two main components. The first element represents the inherent network noise due to the channel jitter or packet loss. This noise component is modeled using samples of about 10^6 packets that are collected over the Internet, and has an average standard deviation of $10ms$. The second component is the jamming noise that is added to packet arrival times by an active adversarial entity. The jamming noise may have any arbitrary distribution and usually is much more notable as compared to the network jitter. In this section, the jamming noise is assumed to be uniformly distributed in the range $[0, \Delta]$.

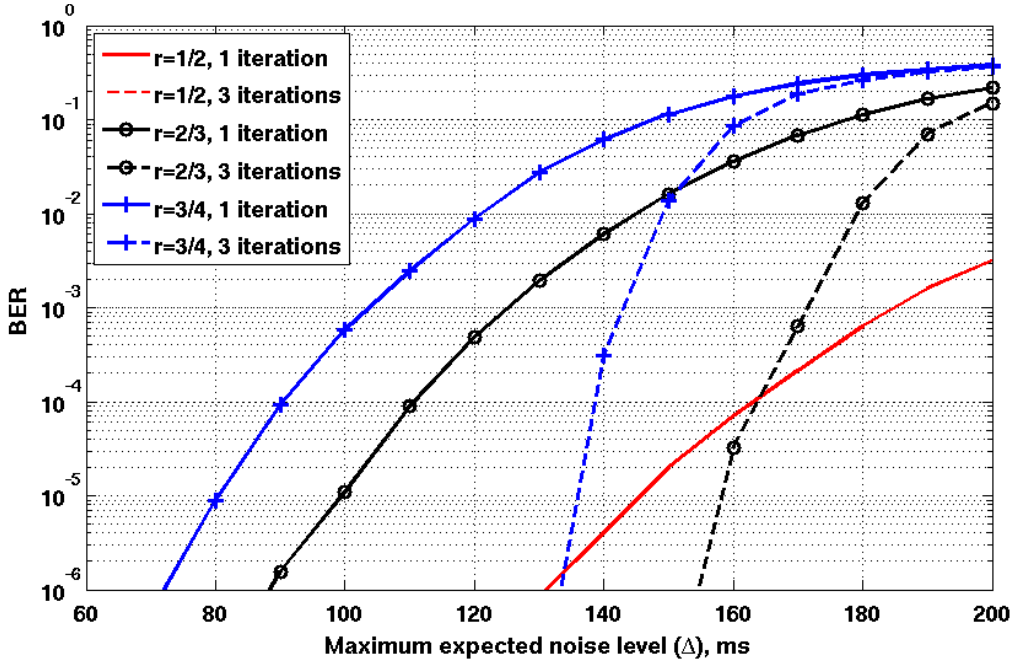
Figure 5.8 illustrates the bit error rate at the covert receiver, given different channel code rates with no modulation guard band. For each individual channel code, the performance of a single decoding round and three rounds of iterative decoding are reported. The graph is based on the maximum expected jamming noise (i.e., Δ) for a uniformly distributed jamming noise. According to the graph, the covert receiver can withstand high levels of adversarial disruption and yet decode the covert message with extremely low error rates.

The graph also reveals the importance of iterative decoding on boosting the decoding performance of the covert receiver. In fact, performing only three rounds of iterative demodulation/decoding can reduce the error rate by more than 4 orders of magnitude (e.g., BER is reduced by a factor of 10^{-4} for $r = 3/4$ at $\Delta = 100ms$). It is worth noting that such a significant improvement is achieved with no extra bandwidth overhead. The computational complexity of decoding at the covert receiver is also minimal due to existence of efficient implementations of the *MAP decoding* algorithm [105].

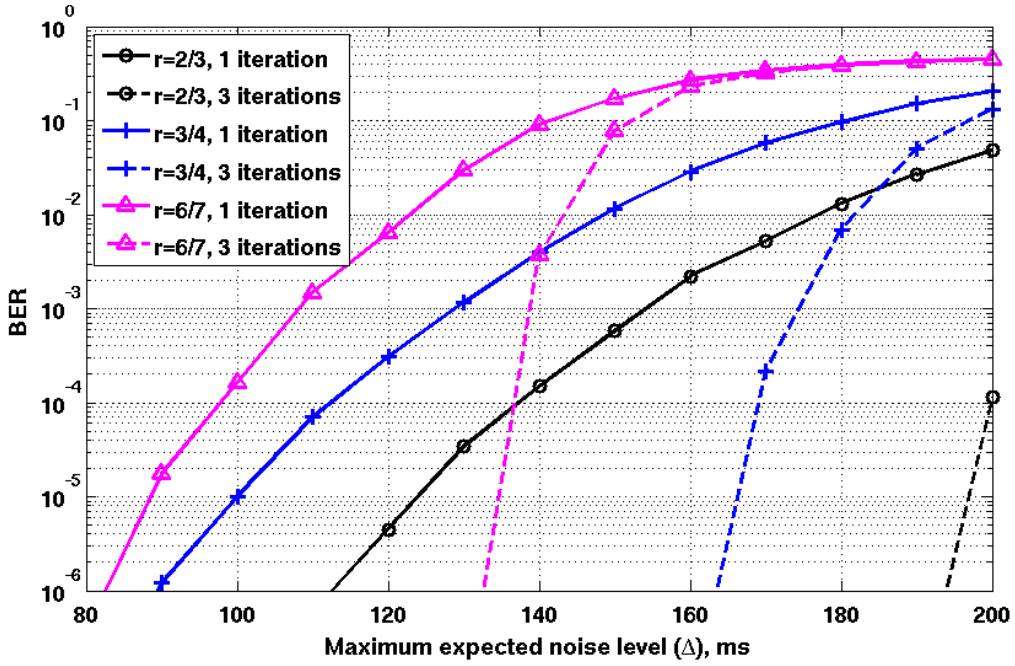
In addition to the channel code, the modulation scheme provides some degrees of freedom that can be exploited using the proposed guard band scheme. The effect of using guard band on the performance of the TCC scheme is depicted in Figure 5.9. In fact, the impact of the modulation guard band on BER of the covert receiver becomes much more pronounced with the iterative decoding in place. For instance, given the maximum jamming delay of $\Delta = 160ms$, a rate 2/3 channel code barely makes a difference when no guard band is used (i.e., Figure 5.8). In contrast, using even small guard band sizes, the error rate decreases to 3×10^{-5} for $\lambda = 0.15$, and even further to less than 10^{-6} for $\lambda = 0.25$.

The graphs in Figure 5.9 also illustrate the effect of increasing the size of the guard band on the decoding performance of the covert receiver. In fact, expanding the guard band separates the IPDs that represent covert bit-0 and covert bit-1 even further. This makes it much easier to remove the channel noise and decode the covert message correctly at the covert receiver. Thus, the covert channel becomes extremely robust and can withstand enormous adversarial disruption and channel noise levels. For instance, it can be observed that increasing the size of the guard band from $\lambda = 0.15$ to $\lambda = 0.25$ enables the covert receiver to withstand adversarial disruptions with relatively high jamming noise levels (e.g., $\Delta = 200ms$), and yet recover the covert message with very low bit error ratio (e.g., 10^{-4}). However, this performance improvement comes with the bandwidth overhead that is needed for the pilot symbols in the covert traffic.

The application of the modulation guard band can be combined with the proper selection of the channel code at the covert transmitter in order to achieve optimum tradeoff



(a) $\lambda = 0.15$

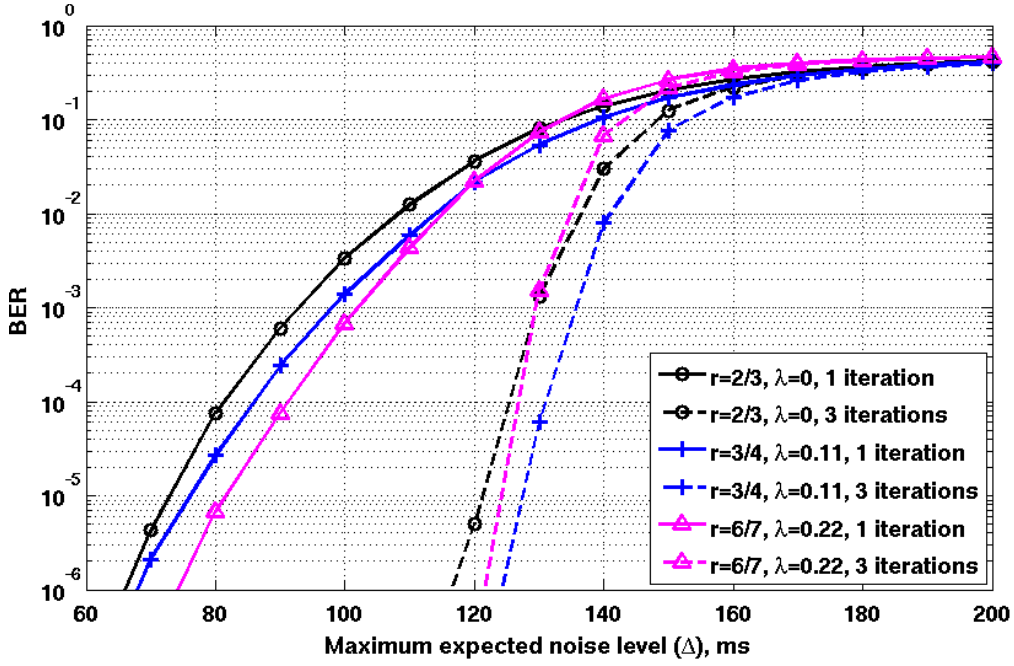


(b) $\lambda = 0.25$

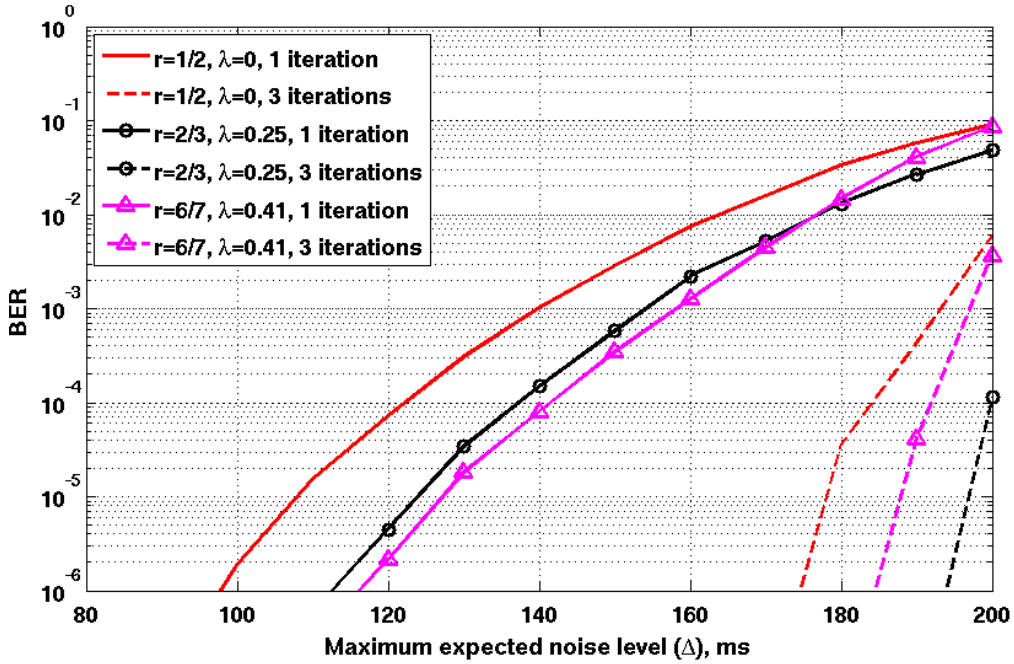
Figure 5.9: BER performance of the TCC scheme with guard band in place.

between the robustness and the covert rate of the channel. Since using the guard band improves the channel reliability, the covert transmitter may increase the channel code rate in order to compensate for the bandwidth reduction due to the pilot symbols. Figure 5.10 compares the performance of different combinations of guard band sizes and channel code rates while the covert rate of the channel is kept constant in each scenario. According to the graph, by combining the guard band and the iterative demodulation/decoding technique, one can push the performance boundaries of the TCC scheme even further and achieve higher covert rates with lower decoding error rates simultaneously. For instance, The graph illustrates that with $\Delta = 130ms$ and after three rounds of iterative decoding, the decoding performance of the combination of the guard band of size $\lambda = 0.11$ and a rate 3/4 convoloutional code (i.e., Figure 5.10a) is 10 times (i.e., one order of magnitude) superior than the BER performance of a rate 2/3 convolutional code with no guard band (i.e., $\lambda = 0$). It is worth noting that in both cases the achievable covert rate of the channel is similar (i.e., $\rho = 2/3bpp$). Moreover, the decoding complexity of the covert receiver in both cases are the same as the receiver performs three rounds of iterative demodulation and decoding in either configuration.

The plots in Figure 5.10 reveal another important design criterion of the TCC scheme. According to the graph, the trade-off between the selected code rate and the guard band size is not linear, and varies depending on the covert rate and noise level in the channel. It is noted that, in order to fix the covert rate of the channel, enlarging the guard band should be combined with increasing the code rate of the channel code. However, increasing the code rate reduces the strength of the channel code up to a point that the decoding block at the covert receiver can not efficiently contribute to the iterative demodulation/decoding process, specifically in noisy channel conditions. The result is a poor decoding performance for covert channel configurations with high rate channel codes. As an example, for the maximum expected noise level of $\Delta = 130ms$ and the covert rate of $\rho = 2/3bpp$, the graphs in Figure 5.10a illustrates similar decoding performance (i.e., BER $\approx 10^{-3}$) for the covert channel configuration that uses no modulation guard band (i.e., $r = 2/3$, $\lambda = 0$), and the setup that is formed by combining the guard band of size $\lambda = 0.22$ and a rate 6/7 convoloutional code. Meanwhile, the combination of a rate 3/4 convoloutional code



(a) Covert rate= 2/3 *bpp*



(b) Covert rate= 1/2 *bpp*

Figure 5.10: BER performance of the TCC scheme given a fixed covert rate.

with the guard band of size $\lambda = 0.11$ hits a remarkably low error rate at 7×10^{-5}. This is a considerable improvement in the decoding performance of the covert receiver which is achieved with no bandwidth overhead or computational complexity at the covert receiver. Therefore, one has to pay special attention in balancing the size of the guard band and the channel code rate, when designing a turbo covert channel.

5.5 Target Applications

Covert channels over public communication networks have attracted a lot of attention due to their application in secure communication, hidden coordination, side channel authentication, and their effectiveness in watermarking packet streams that flow in/out of a communication network. Typically, these applications enforce the requirement of having a stealthy and robust covert communication channel that resists adversarial disruption without being detected.

Providing a stealth communication channel to transmit authentication information is another promising line of application for covert channels. In order to improve system security, each node has to be authenticated before it receives service in the system. Using cryptographic primitives for authentication ensures the correctness of the authentication process (i.e., only legitimate users receive service), however it can not protect the channel that is used to exchange the authentication data. Thus, an adversarial entity may target such communication channels in order to disrupt the authentication process. Therefore, it is suggested to use covert communication schemes to send authentication information through a hidden communication channel in the system [64, 106].

The impressive ability of public communication networks (e.g., Internet) in reaching beyond geographical and political borders has triggered intense censorship activities in order to restrict data communication over computer networks. Thwarting censorship barriers is another major application of network covert channels. In particular, there are several proposals on using covert channels to bypass filtering systems that aim to limit the free flow of information over the Internet [107,108]. It is noted that using encryption or anonymization

techniques can only protect the contents of the communication channel or the identity of parties that exchange information over the network. However these methods are hardly effective in dealing with restrictions that are usually imposed by censorship authorities.

Covert channels have been extensively used in de-anonymization and flow watermarking roles. It is noted that the anonymization services in communication networks can be easily exploited by attackers or compromised users in order to lunch untraceable attacks. Using covert channels the downstream network nodes can glean more information about the upstream path of a packet stream. In this way, it is possible to correlate network flows through different network nodes (i.e., stepping stones) [109], or trace back the source of a network flow (i.e., potential attacker) that may have come through an anonymization system in order to hide the true identity of the transmitter. [110].

The application domain of covert communication schemes over computer networks is rapidly expanding, and every day a new approach that incorporates covert channels for secure communication is introduced in the literature. However, the reliability of a covert channel is considered to be a decisive factor when it comes to deploying covert channels in real life scenarios. In fact, the sensitivity of covert channels to the network noise and adversarial disruptions often proves to be a major obstacle that limits the functionality of covert channels in practice. The proposed turbo covert communication scheme provides a promising solution to this challenge as it achieves a high level of undetectability combined with remarkable tolerance against channel noise and adversarial disruptions.

5.6 Chapter Summary

In this chapter a new framework for robust, undetectable and high rate timing covert channels over public communication networks is introduced. Based on the proposed framework, a model-based covert channel is designed that combines a trellis structure and an adaptive modulation scheme to map the covert message into inter-packet delays of the covert traffic. The design enables the covert receiver to perform iterative demodulation/decoding of the covert message that is proven to achieve extremely high level of robustness even in the pres-

ence of active adversarial entities in the channel. A salient feature of the proposed covert communication scheme is that it achieves provable undetectability regardless of the channel condition, desired robustness, and the covert rate of the channel. Performance analysis of the proposed scheme reveals that the turbo covert channel can withstand extremely high levels of channel noise and adversarial disruption while it maintains an excellent undetectability level and high covert rate.

Chapter 6

Conclusion and Future Work

Covert channels have attracted a great deal of attention as a security threat in computer systems. Traditionally, covert channels are viewed as communication pathways that are neither designed nor intended to exchange information. However, these paths can be exploited by unorthodox methods to leak sensitive information beyond the permitted boundaries that are defined by access control authorities. The secrecy of theses channels mostly relies on inherited secrecy from the obscurity of the communication method and the resource that is used for communication. However, as the shared resource is exposed, the covert traffic can be detected due to the effects of modulating a covert message over normal characteristics of the shared resource.

The evolution of computer networks and wireless technology led the communication community to revisit the concept of covert communication not only as a security threat but also as an alternative way of providing security and privacy to communication networks. However, network covert channels often rely on resources in target communication network that are likely to be monitored by system observers and traffic analysis entities. Therefore, new design strategies are required to construct stealth, reliable and high rate covert channels that are suited for such a noisy and tightly monitored environment.

In this dissertation we explored a novel design methodology for network covert channels. Undetectability, robustness and the achievable covert rate of the channel are the major ob-

jectives in our design methodology. The design methodology is based on the concept of behavioral mimicry in which the covert transmitter aims on mimicking the statistical behaviors of a normal node in the system. In this way, the generated covert traffic inherits the same statistical properties of a normal traffic flow of the system. Thus, it would be extremely difficult for a system observer to track down the covert channel based on the statistical characteristics of the covert traffic or the behavioral fingerprints of the covert transmitter. In order to identify the proper medium for covert communication we propose to search for available sources of randomness in communication protocols and network environments. It is noted that randomness is used in design of communication protocols in order to achieve fairness, stability, and scalability. Such sources of random behavior can be exploited for covert communication in order to facilitate modulating a covert message over behavioral characteristics of the covert transmitter. In fact, in this thesis we have identified several sources of randomness in communication protocols such as random behaviors in access control protocols (e.g., CSMA/CA) and cryptographic algorithms (e.g., digital signature schemes). By combining the covert modulation process with the inherent random behavior of the communication protocol, the covert transmitter achieves more degrees of freedom and releases itself from constraints that are imposed by the existence of a fixed covert message in its behavioral fingerprints. Thus, the covert transmitter acts more like a normal transmitter of the system and the covert traffic resembles the properties of a normal traffic flow in the network. A salient feature of such design methodology is the ability to design covert channels with high levels of stealthiness that is a vital requirement for covert communication over tightly monitored environments like communication networks.

Robustness and achievable rate of covert communication schemes are also studied in this dissertation. In fact, without a reliable communication channel, the covert receiver can not decode the covert message and the channel becomes practically ineffective. More importantly, the channel design needs to be resistant against active adversarial attacks in which an active adversary deliberately introduces noise to the covert traffic in order to disrupt the normal flow of covert information. To elaborate more on our design mythology, in the second part of this dissertation we have introduced two practical covert communication schemes according to the behavioral mimicry covert communication approach. To

this end, wireless LAN and public computer networks (i.e., Internet) are selected as target environments for covert communication. The selection is due to the diverse applications, incredibly widespread usage, enormous number of devices, and the huge volume of information that are exchanged over these networks. For each covert communication scheme a three step design approach is followed. First of all, we identified a proper source of randomness in the target communication network. This leads to the selection of the medium that is used for covert communication. Then, an approach to exploit the shared resource and modulate the covert message is proposed. The modulation approach is designed specifically based on the concept of behavioral mimicry. Thus, it enables the covert transmitter to mimic the behavioral characteristics of a normal transmitter in the network. Finally, proper modifications are proposed in order to increase the robustness and the achievable rate of the covert channel.

The proposed wireless covert communication scheme is based on the structural behaviors of the CSMA/CA and the binary back off algorithms. The covert message in this approach is modulated over the inter-packet delays of the covert transmitter's traffic, however the delays are measured based on a uniquely designed clock called the *covert clock*. The covert clock is a virtual clock that uses the channel activities of a subset of regular nodes in the system as its *clock ticks*. The packet transmission process of the covert transmitter is controlled by the covert clock as well. Consequently, the packet transmission activities of the covert transmitter are automatically linked with the normal transmission pattern of the elements of the network. This adaptive behavior proves to be crucial in making an undetectable and stealth covert communication scheme. Using the broadcast nature of the wireless environment, the covert receiver is capable of observing the same channel activities as those that are observed by the covert transmitter. Thus, the covert receiver can synchronize its covert clock with the one at the covert transmitter. In order to match the behavioral characteristics of the covert transmitter to the ones of a normal node in the channel, three design criteria for the covert transmitter are proposed (i.e., transmission rate, transmission window, expansion postpone design criteria). The adaptive design of the covert transmitter enables it to mimic not only the long-term statistical characteristics of a normal transmitter in the channel, but also it can exhibit the temporal

short-term behaviors of an ordinary network node. This means that the covert transmitter can react to unpredictable network events (e.g., packet loss) very similar to an ordinary transmitter in the channel. Thus, it is difficult to detect the covert channel even using second order statistical test like the regularity score. By using the properties of the original communication protocol (e.g., acknowledgment messages), the covert receiver increases the likelihood of decoding the covert message and improve the channel robustness in order to operate in the error-prone wireless environment. Another important feature of the proposed scheme is that the covert rate of the channel linearly increases with the overt rate of the communication channel. We have backed up our analysis of the proposed scheme with a series of numerical analysis tests. The numerical results confirm that the proposed wireless covert communication scheme achieves relatively high covert communication rates with outstanding security and reliability scores.

We also introduced turbo covert channels which are a family of model-based timing covert channels over public communication networks (e.g., Internet). Turbo covert communication schemes are designed based on sound engineering foundations of communication theory, and they are intended to be extremely reliable against network noise and adversarial disruptions. To this end, we have modeled the covert channel as a differential communication channel over a noisy environment. The covert modulation process is formed by combining an efficient trellis structure with an adaptive modulation scheme that maps a covert message into inter-packet delays of the covert traffic. The adaptive modulation design gives the covert transmitter enough degrees of freedom to use only the error resistance portion of the available inter-packet delay spectrum without compromising the stealthiness property of the channel. In fact, assuming that the inter-packet delays of the overt traffic of the channel are $(i.i.d.)$ random variables, the proposed covert communication scheme achieves provable polynomial undetectability, regardless of the network condition, desired robustness, and the expected covert rate of the channel. This means the covert channel is undetectable against any polynomial-time statistical test that analyzes samples of the covert traffic and the legitimate traffic of the system. It is noted that this assumption does not limit the practically of the turbo covert channel design as $(i.i.d.)$ random processes have been widely used in modeling the traffic of communication networks. The design

of the covert modulation process enables the covert receiver to perform iterative demodulation/decoding of the covert message that is proven to achieve extremely high level of robustness even in the presence of active adversarial entities in the channel. We argue that the proposed covert communication scheme is a solid candidate for a diverse range of applications that require robust and undetectable covert channels in communication network. We have discussed a few of such applications and provided insight on how the proposed scheme can be adapted for those services in practical scenarios.

In brief, the contribution of this dissertation can be summarized as follows:

- Based on the concept of behavioral mimicry, we proposed a design methodology that addresses the challenge of designing robust, undetectable, and high rate covert channels over communication networks.

- We have recognized the inherent randomness in communication protocols and network environments as the key in finding a proper medium for covert communication. We have identified several sources of randomness and provided examples on how these random behaviors lead to the discovery of suitable shared resources for covert channels.

- We introduced a novel covert communication scheme for wireless networks. The design of the new covert communication scheme begins with identifying the source of randomness in the legacy system which leads to the selection of the medium for the covert communication scheme. Next a step-by-step design approach was presented in order to adopt the original communication protocol for the purpose of covert communication. The result is a set of design criteria that defines the operational parameters of the covert transmitter.

- We introduced turbo covert channels (TCC) which are a family of model-based network timing covert channels. The TCC family is capable of achieving incredibly high levels of robustness due to the implementation of an adaptive modulation scheme at the covert transmitter and the iterative demodulation/decoding at the covert receiver. The covert channel also proved to be polynomially undetectable, under correct

assumptions, against any polynomial-time statistical test that runs on the samples of the covert traffic and the legitimate traffic of the channel.

In addition, we have supported our proposed design methodology and covert communication schemes with extensive numerical analysis and tests. In fact, the proposed schemes are studied from undetectability, reliability, and achievable covert rate perspectives. The dissertation also includes discussions on prospective applications and the feasibility of deploying the proposed covert communication schemes in practical scenarios.

6.1 Future Directions

We have identified two main fronts that one can consider as future directions on extending the contributions of this dissertation. The first front is focused on the concept of behavioral mimicry covert communication and the design methodology that is presented in this dissertation. In this thesis, the randomness in communication protocols and computer networks is studied as the gateway to select proper resources for covert channels. We also presented few examples of such random sources in communication protocols. However, the diversity of network topologies, devices, and communication protocols implies the existence of countless possible sources of randomness that are yet to be discovered and studied from covert communication perspective. As an example, we discussed the application of behavioral mimicry covert communication in sensor networks. Design of such covert communication schemes demands extensive research and requires further study of the design methodology that is presented in this dissertation.

The second angle on extending the contributions of this thesis is on improving the performance of the proposed covert communication schemes. The proposed turbo covert communication scheme is mainly designed with the mindset of optimizing the robustness of the covert channel under the constraint that the channel should achieve provable polynomial undetectability. The other perspective of such design is to consider the achievable covert rate of the channel as the primary objective. In more detail, the presented version of the turbo covert transmitter modulates only one bit of the covert codeword in each packet

transmission. By splitting the domain of available inter-packet delays to more than two sections, one can increase the number of possible symbols per packet transmission in order to increase the achievable covert rate of the covert channel. However, this requires some modifications in the modulation scheme, guard-band configuration, and more importantly the design of the trellis structure that is used for encoding and decoding the covert message. It is noted that increasing the covert rate of the channel inevitably impacts the reliability of the covert communication scheme which has to be studied as well.

As mentioned in Chapter 5, preventing illegal exploitation of privacy preserving services is one of the most promising applications of robust and undetectable network covert channels including the turbo covert channel. However, implementing such covert communication scheme over an overlay of privacy preserving nodes would expose the channel to re-packetization, and extensive packet reordering noise. The effect of such environment may significantly reduce the reliability of the covert channel, and limits its efficiency for watermarking network flows over public communication networks. In [50] it is suggested to use a family of trellis codes called *toroidal codes* for covert communication. These channel codes are specifically designed to control the effect of symbol injection error events on the robustness of network covert channels. We conjecture that the trellis structure that is presented in [50] would be an exceptional addition to the proposed turbo covert communication scheme, in particular against inherent network noise due to the privacy preserving services and complex adversarial attacks. The same research plan can be applied to the proposed wireless covet communication scheme. In fact, the proposed wireless covert communication scheme relies on traditional implementations of convoloutional codes to handle possible error events in the channel. Although the covert receiver is supposed to be capable of tracking lost and retransmitted packets using sequence number information of the received packets, we believe the combination of the proposed covert communication scheme and the *toroidal codes* would increase the resilience of the covert channel design, in particular against byzantine attacks in wireless networks. This approach can also be an efficient addition for covert communication schemes in sensor networks.

References

[1] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 3, pp. 44–57, 2007.

[2] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[3] B. Lampson, "A note on the confinement problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[4] D. Bell and L. La Padula, "Secure computer system: Unified exposition and multics interpretation," tech. rep., DTIC Document, 1976.

[5] B. Lampson, "Protection," *ACM SIGOPS Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.

[6] S. Lipner, "A comment on the confinement problem," *ACM SIGOPS Operating Systems Review*, vol. 9, no. 5, pp. 192–196, 1975.

[7] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," vol. 37, no. 5, pp. 164–177, 2003.

[8] S. Foley, "A model for secure information flow," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 248–258, IEEE, 1989.

[9] J. Gray III, "Countermeasures and tradeoffs for a class of covert timing channel," tech. rep., Hong Kong University of Science and Technology, 1994.

[10] S. Katzenbeisser and F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA, USA: Artech House, Inc., 2000.

[11] S. Brand, "Dod 5200.28-std department of defense trusted computer system evaluation criteria (orange book)," *National Computer Security Center*, 1985.

[12] G. Simmons, "The subliminal channel and digital signatures," *Advances in Cryptology*, pp. 364–378, 1985.

[13] T. Handel and M. Sandford, "Hiding data in the OSI network model," in *Information Hiding*, pp. 23–38, Springer, 1996.

[14] R. Kemmerer, "Shared resource matrix methodology: An approach to identifying storage and timing channels," *ACM Transactions on Computer Systems*, vol. 1, no. 3, p. 277, 1983.

[15] C. Girling, "Covert channels in LANs," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 292–296, 1987.

[16] J. Patrik and P. Gallagher, *A guide to understanding covert channel analysis of trusted systems*. National Computer Security Centre NCSCTG030, 1993.

[17] D. Kundur and K. Ahsan, "Practical internet steganography: data hiding in IP," in *Proceedings of Texas Workshop on Security of Information Systems*, vol. 2, 2003.

[18] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in internet traffic with active wardens," in *Proceedings of the Information Hiding Conference*, pp. 18–35, Springer, 2003.

[19] K. Ahsan and D. Kundur, "Practical data hiding in TCP/IP," in *Proceedings of the ACM Workshop on Multimedia Security*, vol. 2002, 2002.

[20] J. Rutkowska, "The implementation of passive covert channels in the Linux kernel," in *Chaos Communication Congress*, 2004.

[21] S. Murdoch and S. Lewis, "Embedding covert channels into TCP/IP," in *Proceedings of the Information Hiding Conference*, pp. 247–261, Springer, 2005.

[22] J. Wray, "An analysis of covert timing channels," *Journal of Computer Security*, vol. 1, no. 3, pp. 219–232, 1992.

[23] V. Berk, A. Giani, G. Cybenko, and N. Hanover, "Detection of covert channel encoding in network packet delays," tech. rep., Department of Computer Science, Dartmouth College, Technical Report TR2005536, 2005.

[24] J. Giles and B. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2455–2477, 2002.

[25] W. Hu, "Reducing timing channels with fuzzy time," *Journal of Computer Security*, vol. 1, no. 3, pp. 233–254, 1992.

[26] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert messaging through TCP timestamps," in *Privacy Enhancing Technologies*, pp. 189–193, Springer, 2003.

[27] J. Millen, "20 years of covert channel modeling and analysis," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 113–114, IEEE, 1999.

[28] K. Ahsan, "Covert channel analysis and data hiding in TCP/IP," Master's thesis, University of Toronto, 2002.

[29] C. Rowland, "Covert channels in the TCP/IP protocol suite," *First Monday Peer Reviewed Journal on the Internet*, vol. 2, no. 5, 1997.

[30] C. Abad, "IP checksum covert channels and selected hash collision," *Available at: http://www.gray-world.net/papers/ipccc.pdf*, 2001.

[31] R. Rivest *et al.*, "Chaffing and winnowing: Confidentiality without encryption," *CryptoBytes (RSA laboratories)*, vol. 4, no. 1, pp. 12–17, 1998.

[32] M. Bellare and P. Rogaway, "Collision-resistant hashing: Towards making UOWHFs practical," in *Proceedings of the 17th Annual International Cryptology Conference*, p. 470, Springer, 1997.

[33] M. Padlipsky, "Limitations of end-to-end encryption in secure computer networks," tech. rep., DTIC Document, 1978.

[34] S. Cabuk, C. Adviser-Brodley, and E. Adviser-Spafford, *Network covert channels: design, analysis, detection, and elimination*. PhD thesis, Purdue University, 2006.

[35] S. Cabuk, C. Brodley, and C. Shields, "IP covert timing channels: design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 178–187, ACM, 2004.

[36] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in *Proceedings of the 15th conference on USENIX Security Symposium*, vol. 15, p. 5, USENIX Association, 2006.

[37] S. Gianvecchio and H. Wang, "Detecting covert timing channels: an entropy-based approach," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 307–316, ACM, 2007.

[38] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert channel detection," *ACM Transactions on Information Systems and Security*, vol. 12, pp. 22:1–22:29, Apr. 2009.

[39] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-based covert timing channels: Automated modeling and evasion," in *Recent Advances in Intrusion Detection*, pp. 211–230, Springer, 2008.

[40] Y. Liu, D. Ghosal, F. Armknecht, A. Sadeghi, S. Schulz, and S. Katzenbeisser, "Hide and Seek in Time: Robust Covert Timing Channels," in *Proceedings of the 14th European Symposium of Research in Computer Security (ESORICS)*, pp. 120–135, Springer, 2010.

[41] R. Walls, K. Kothari, and M. Wright, "Liquid: A detection-resistant covert timing channel based on ipd shaping," *Computer Networks*, vol. 55, no. 6, pp. 1217–1228, 2011.

[42] K. Kothari, "Mimic: An active covert channel that evades regularity-based detection," Master's thesis, Department of Computer Science and Engineering, University of Texas at Arlington, 2010.

[43] S. Sellke, C. Wang, S. Bagchi, and N. Shroff, "TCP/IP timing channels: Theory to implementation," in *Proceedings of the 28th IEEE International Conference on Computer Communications*, pp. 2204–2212, IEEE, 2009.

[44] S. Sellke, C. Wang, S. Bagchi, and N. Shroff, "Camouflaging timing channels in web traffic," tech. rep., School of Elecetrical and Computer Engineering, Purdo University, 2009.

[45] R. W. Smith and G. Scott Knight, "Predictable design of network-based covert communication systems," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 311–321, IEEE, 2008.

[46] A. Houmansadr and N. Borisov, "CoCo: coding-based covert timing channels for network flows," in *Proceedings of the Information Hiding Conference*, pp. 314–328, Springer, 2011.

[47] "Covert Fountain," tech. rep., Department of Computer Science, University of California Davis,Technical Report CSE-2011-6, 2011.

[48] M. Rodrigues, "Timing channel codes for covert communication and network flow tracing," Master's thesis, University of Illinois at Urbana-Champaign, 2012.

[49] Y. Liu, D. Ghosal, F. Armknecht, A. Sadeghi, S. Schulz, and S. Katzenbeisser, "Robust and undetectable steganographic timing channels for iid traffic," in *Proceedings of the Information Hiding Conference*, pp. 193–207, Springer, 2010.

[50] R. W. Smith and G. S. Knight, "Predictable three-parameter design of network covert communication systems," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 1–13, 2011.

[51] S. Lin and D. Costello, *Error control coding*. Prentice-Hall Englewood Cliffs, NJ, 1983.

[52] K. Szczypiorski, "HICCUPS: Hidden communication system for corrupted networks," in *Proceedings of the International Multi-Conference on Advanced Computer Systems*, pp. 31–40, 2003.

[53] L. Butti and F. VEYSSET, "Wi-Fi Advanced Stealth," in *Proceedings of the Black Hat US*, 2006.

[54] S. Li and A. Ephremides, "A Network Layer Covert Channel in Ad-hoc Wireless Networks," in *Proceedings of the 1st IEEE Conference on Sensor and Ad-Hoc Communications and Networks (SECON)*, pp. 88–96, 2004.

[55] T. Carroll, "Chaotic communications that are difficult to detect," *Physical Review E*, vol. 67, no. 2, p. 026207, 2003.

[56] J. Proakis *et al.*, *Digital Communication*. Osborne-McGraw-Hill, 2001.

[57] W. Ditto and T. Munakata, "Principles and applications of chaotic systems," *Communications of the ACM*, vol. 38, no. 11, pp. 96–102, 1995.

[58] H. Schuster and W. Just, *Deterministic chaos*. Wiley-VCH, 2006.

[59] E. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[60] T. Carroll, "Noise-robust synchronized chaotic communications," *IEEE Transactions on Circuits and Systems*, vol. 48, no. 12, pp. 1519–1523, 2001.

[61] L. Pecora and T. Carroll, "Synchronization in chaotic systems," *Physical review letters*, vol. 64, no. 8, pp. 821–824, 1990.

[62] E. Schöll and H. Schuster, *Handbook of chaos control*. Wiley-Vch, 2008.

[63] L. Frikha, Z. Trabelsi, and W. El-Hajj, "Implementation of a covert channel in the 802.11 header," in *Proceedings of the International Wireless Communications and Mobile Computing Conference*, pp. 594–599, IEEE, 2008.

[64] S. Li and A. Ephremides, "Covert channels in ad-hoc wireless networks," *Ad Hoc Networks*, vol. 8, no. 2, pp. 135–147, 2010.

[65] T. Doğu and A. Ephremides, "Covert information transmission through the use of standard collision resolution algorithms," in *Proceedings of the Information Hiding Conference*, pp. 419–433, Springer, 2000.

[66] S. Li and A. Ephremides, "A covert channel in MAC protocols based on splitting algorithms," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1168–1173, 2005.

[67] Z. Wang, J. Deng, R. Lee, and P. Princeton, "Mutual anonymous communications: a new covert channel based on splitting tree MAC," in *Proceedings of the 26th IEEE International Conference on Computer Communications*, pp. 2531–2535, 2007.

[68] T. Calhoun, R. Newman, and R. Beyah, "Authentication in 802.11 LANs using a covert side channel," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, IEEE, 2009.

[69] C. Krätzer, J. Dittmann, A. Lang, and T. Kühne, "WLAN steganography: a first practical review," in *Proceedings of the 8th Workshop on Multimedia and Security*, pp. 17–22, ACM, 2006.

[70] R. Holloway, "Covert DCF - a DCF based covert timing channel in 802.11 networks," Master's thesis, Georgia State University, Atlanta, Georgia, 2010.

[71] S. Bhadra, S. Bodas, S. Shakkottai, and S. Vishwanath, "Communication through jamming over a slotted ALOHA channel," *IEEE Transactions on Information Theory*, vol. 54, no. 11, p. 5257, 2008.

[72] N. Kothari, R. Mahajan, T. Millstein, R. Govindan, and M. Musuvathi, "Finding protocol manipulation attacks," in *Proceedings of the ACM SIGCOMM 2011 Conference*, vol. 41, p. 26, 2011.

[73] D. Denning, "A lattice model of secure information flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[74] P. Porras and R. Kemmerer, "Covert flow trees: A technique for identifying and analyzing covert storage channels," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 36–51, IEEE, 1991.

[75] R. Kemmerer, "A practical approach to identifying storage and timing channels: Twenty years later," in *Proceedings of the 18th Annual Computer Security Applications Conference*, pp. 109–118, IEEE, 2002.

[76] A. Donaldson, J. McHugh, and K. Nyberg, "Covert channels in trusted LANs," in *Proceedings of the 11th National Computer Security Conference*, pp. 17–20, 1988.

[77] A. Jeng and M. Abrams, "On network covert channel analysis," in *Proceedings of the 3rd Aerospace Computer Security Conference*, 1987.

[78] I. Moskowitz and M. Kang, "Covert channels here to stay," in *Proceedings of the 9th Annual Conference on Computer Assurance, Safety, Reliability, Fault Tolerance, Concurrency and Real Time Security*, pp. 235–243, IEEE, 1994.

[79] J. Millen, "Covert channel capacity," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 60–66, 1987.

[80] I. Moskowitz and A. Miller, "Simple timing channels," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 56–64, IEEE, 1994.

[81] M. Kang and I. Moskowitz, "A pump for rapid, reliable, secure communication," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 119–129, ACM, 1993.

[82] M. Kang, I. Moskowitz, and D. Lee, "A network pump," *IEEE Transactions on Software Engineering*, vol. 22, no. 5, pp. 329–338, 1996.

[83] M. Kang, I. Moskowitz, and S. Chincheck, "The pump: A decade of covert fun," in *Proceedings of the 21st Annual Computer Security Applications Conference*, pp. 7–pp, IEEE, 2005.

[84] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, random variables, and stochastic processes*. McGraw-Hill New York, 2002.

[85] P. Peng, P. Ning, and D. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 15–pp, IEEE, 2006.

[86] S. Gianvecchio and H. Wang, "An entropy-based approach to detecting covert timing channels," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 785–797, 2011.

[87] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[88] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[89] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton: CRC Press, 1997.

[90] A. Tanenbaum and M. Van Steen, *Distributed systems*, vol. 2. Prentice Hall, 2002.

[91] I. . W. Group, "IEEE 802.11 WG. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, 2007.

[92] J. Proakis and M. Salehi, *Digital communications*. McGraw-Hill New York, 2001.

[93] Q. Ni, T. Li, T. Turletti, and Y. Xiao, "Saturation throughput analysis of error-prone 802.11 wireless networks," *Wireless Communications and Mobile Computing*, vol. 5, no. 8, pp. 945–956, 2005.

[94] L. Kleinrock, *Queueing Systems*. Wiley-interscience, 1975.

[95] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking (ToN)*, vol. 3, no. 3, pp. 226–244, 1995.

[96] W. Li, R. Fretwell, and D. Kouvatsos, "Analysis of correlated traffic by batch renewal process," in *Proceedings of the International Conference on E-Business and Information System Security*, pp. 1–5, IEEE, 2009.

[97] "The CAIDA UCSD Anonymized Internet Traces 2011," *http://www.caida.org/data/passive/passive_2011_dataset.xml*.

[98] S. Ross, *A First Course in Probability*. Prentice Hall, NJ, 2009.

[99] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[100] P. Hoeher and J. Lodge, "Turbo DPSK: Iterative differential PSK demodulation and channel decoding," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 837–843, 1999.

[101] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 1064–1070, IEEE, 1993.

[102] M. Gertsman and J. Lodge, "Symbol-by-symbol MAP demodulation of CPM and PSK signals on Rayleigh flat-fading channels," *IEEE Transactions on Communications*, vol. 45, no. 7, pp. 788–799, 1997.

[103] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[104] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[105] M. Valenti, "A guided tour of CML, the coded modulation library," *Morganown, W. Va.: Iterative Solutions*, 2008.

[106] R. Degraaf, J. Aycock, and M. Jacobson, "Improved port knocking with strong authentication," in *Proceedings of the 21st Annual Computer Security Applications Conference*, pp. 10–pp, IEEE, 2005.

[107] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, "Infranet: Circumventing web censorship and surveillance," in *Proceedings of the 11th USENIX Security Symposium*, pp. 247–262, USENIX Association, 2002.

[108] A. Houmansadr, G. Nguyen, M. Caesar, and N. Borisov, "Cirripede: circumvention infrastructure using router redirection with plausible deniability," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 187–200, ACM, 2011.

[109] X. Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 20–29, ACM, 2003.

[110] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 116–130, IEEE, 2007.