Modeling, Analysis, and Detection of Information Leakage via Protocol-Based Covert Channels

MODELING, ANALYSIS, AND DETECTION OF INFORMATION LEAKAGE VIA PROTOCOL-BASED COVERT CHANNELS

BY

JASON JASKOLKA, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applicd Science (2010)
(Software Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE:	Modeling, Analysis, and Detection of Information Leakage via Protocol-Based Covert Channels
AUTHOR:	Jason Jaskolka B.Eng. (Software Engineering) McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Dr. Ridha Khedri
NUMBER OF PAGES:	xiii, 132

To my family

Abstract

With the emergence of computers in every day activities and with the ever-growing complexity of networks and network protocols, covert channels are becoming an eminent threat to confidentiality of information. With increasing sensitivity of data in many computer application domains, the leakage of confidential information can have severe repercussions on the institution from which the information was leaked.

In light of this eminent threat, we propose a technique to detect confidential information leakage via covert channels. We limit our focus to instances where the users of covert channels modulate the information that is being sent; either by encryption, or some other form of encoding.

In the literature, the difference between classes of covert channels under the current classification is unclear. This lack of clarity results in the development of incomplete techniques for modeling, detecting and preventing covert channels. In this thesis, we propose a new classification for covert channels which organizes covert channels into two types: protocol-based covert channels and environment-based covert channels. We also develop a novel, comprehensive model for protocol-based covert channel communication. Using the developed model, we explore the relationship between covert channel communication, steganography and watermarking. The intent is to provide a better understanding of covert channel communication in an attempt to develop investigative support for confidentiality. Finally, we propose a technique for detecting confidential information leakages via covert channels. The technique is based on relation algebra and offers tests for verifying the existence of an abstraction relation which relates the confidential information to the information that is observed to be sent on the communication channel. It focuses on protocol-based covert channels.

With a better understanding of covert channel communication, we are able to develop more effective and efficient mechanisms for detecting and preventing the use of covert channels to leak confidential information in computer systems.

Acknowledgements

I would first and foremost like to extend my sincere appreciation and thanks to my supervisor, Dr. Ridha Khedri, for his guidance, motivation, encouragement, and support throughout the preparation of this thesis.

I would like to thank Dr. William Farmer and Dr. Ryan Leduc for their constructive, thoughtful, and useful comments. I would like to thank Khair Eddin Sabri for his valuable discussions and suggestions.

The preparation of this thesis was made possible due to the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

A very special thank you goes to my parents, my sister, and my brother, for all their love, encouragement, and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we introduce covert channels in the context of computer and information security, providing motivation for the need to detect and eliminate covert channels in computer systems. In Section 1.1, we give an introduction to computer and information security and explain its ever-growing importance in today's society. In Section 1.2, we introduce covert channels and indicate their use in computer systems. In Section 1.3, we provide a review of the literature and discuss some existing techniques for preventing and detecting covert channels in computer systems while indicating how the existing techniques are not sufficient to practically uncover covert channels. In Section 1.4, we give the motivation for a new technique to detect covert channels and help maintain the confidentiality of information. In Section 1.5, we state the problem subject of our work. In Section 1.6, we summarize our contributions. Finally, in Section 1.7, we give the structure of the remainder of the thesis.

1.1 General Context

With the ever-growing popularity and sophistication of computer systems, computer and information security is becoming more important than ever. Computers are being used in virtually every workplace in some form or another. Hence, due to the widespread use of computers and the variety of application domains, security concerns have varying implications and priority from one domain to another.

Computer and information security has three facets which have a strong relationship to covert channel communication: confidentiality, integrity, and availability [Bis02]. Confidentiality refers to the concealment of information or resources. The demand to keep information concealed arises from the use of computers in government, medical, and industry domains. For example, military institutions in the government restrict access to information to those individuals or groups who have a need for that information. Often times, confidentiality can be established by implementing a "need to know" principle which states that access to information shall be granted only to those who require the information. This principle applies to many application domains one of which is industrial organizations, which keep their proprietary designs and information secure such that their competitors are unable to steal trade secrets. Integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change. Integrity includes both data integrity, the content of the information, and origin integrity, the source of the data. Origin integrity is often called authentication. The accuracy and credibility of information relies heavily on the integrity of information and is central to the proper functioning of a system. Availability refers to the ability to use the information or resource desired. Availability is directly related to the reliability of a system since a system that is unavailable is at least as bad as no system at all. In terms of computer and information security, availability has implications that extend to the ability of an agent to deliberately deny access to data or a service by making it unavailable, thus rendering the system unusable. Confidentiality, integrity, and availability are strongly related to covert channels. Covert channels have an enormous impact on confidentiality since they allow for confidential information to be leaked to agents for which the information is not intended. Integrity can be compromised by covert channels since covert channeling techniques enable tampering with data stores in a manner that is unknown to the system. Lastly, covert channels hinder availability since they are able to use system resources to such an extent that it degrades the system's performance and jeopardizes its availability. So, while developing mechanisms to aid in the elimination of covert channels in computer and information systems, we must remember the three facets of computer and information security: confidentiality, integrity, and availability.

In order to discuss computer and information security, we must have a means of specifying what is, and what is not, a violation of security. Hence, we require a *security policy* to state what is, and what is not, allowed. According to [SKJ09], a security policy is a predicate on the knowledge of a set of agents that establishes what each agent should know and communicate. In [SKJ09], an agent's knowledge is captured by a mathematical structure called an *Information Algebra*. Consider a data store of student records which contains information classified as *name*, *id*, and *courses*. A policy for this example may be that no agent should know both the *name* and *id* of a student. Confidential information is defined as the information that is protected by the security policy.

Once we have a specification of what is, and what is not, allowed in the form of a security policy, we can set forth the goals of computer and information security as outlined in [Bis02]. The goals of computer and information security are to prevent an attack, detect an attack, or recover from an attack. Prevention refers to the failure of an attempted attack on a system. For example, if an agent was attempting to gain knowledge of both the *name* and *id* of a student from a data store of student records, and they are unable to gain access to the data store in one way or another, then the attack has been prevented. Prevention techniques are often cumbersome and hinder the performance of a system. However, there exists some simple techniques, such as passwords which prevent (to a certain limit) unauthorized agents from accessing the system, that have been widely accepted and implemented. Detection is often used when an attack cannot be prevented. Detection techniques accept that an attack will occur and attempts to determine if an attack is underway or has occurred. For example, if a monitor were to be watching the access to the student records data store, it would be able to record and report any attempt to access both the *name* and *id* of an individual student and thus the attack would be detected. Typical detection techniques monitor various aspects of the system, looking for actions or information indicating that an attack is underway or has occurred. Lastly, recovery refers to the ability to stop an attack and to assess and repair any damage caused by that attack. For example, if an agent were to delete a student record from the data store, one recovery technique would be to restore the student record from backup. Recovery is often difficult to implement and often times results in impaired performance of the system.

Computer and information security depend on many aspects of a computer system. All aspects of computer security begin with the nature of the threat being dealt with and countering that threat with security techniques whether they be prevention, detection, or recovery techniques in order to maintain the three facets of computer and information security: confidentiality, integrity, and availability. Throughout the rest of this thesis, we will show how a focus on detection techniques for uncovering covert channels in computer systems will aid in the sustainment of confidentiality of information.

1.2 Specific Context

In the area of computer and information security, there are a number of concerns, whether it be the leak of confidential information, the unauthorized manipulation of sensitive data, or the denial of a required service. This thesis focusses particularly on the leak of confidential information via covert channel communication.

The concept of covert channel communication was first introduced by Lampson in 1973 [Lam73]. He defined a covert channel as a communication channel that is neither designed nor intended to transfer information at all [Lam73]. Several definitions for covert channels have since been proposed. According to [Gli93], a covert channel is a parasitic communication channel that draws bandwidth from another channel in order to transmit information without the authorization or knowledge of the latter channel's designer, owner, or operator. According to [Kem83], a covert channel is a channel that uses entities not normally viewed as data objects to transfer information from one subject to another. Although these are valid definitions, we will adopt the definition given in [DoD85] which defines a covert channel as any communication means that can be exploited to transfer information in a manner that violates the system's security policy. This means that channels that may be hidden from the view of others are allowed to exist in a computer system provided they do not violate the security policy employed by the system.

Our analysis of covert channel communication leads us to classify covert channels into two types: *protocol-based covert channels* and *environment-based covert channels*. Similar to the definition given in [HZD05], a protocol-based covert channel is a channel that uses the communication protocol to convey messages that violate a security policy. An example of such a channel is one where agents are able to send either long messages or short messages. A possible encoding for such a protocol can be derived as a mapping where the receipt of a long message maps to 1 and the receipt of a short message maps to 0. By communicating in accordance with the communication protocol, a sender is able to choose to send long and short messages in such a way as to encode a hidden message to be decoded by the receiver. An environment-based covert channel is a communication means that uses environmental resources, functionalities, or features to convey messages that violate a security policy. An example of such a channel is one where two people who are communicating openly in the same room arrange the pens on a desk in a particular way so as to encode a message that is not detected by an observer. An example of an environment-based covert channel in the context of computer systems is one where a sending process is able to modulate the timing of events in such a way that it can be detected by a receiving process whereby the timing of events can be mapped to a particular encoding of information. A particular feature of environment-based covert channels is that the communication of information occurs in an open system which belongs to the public domain. Each user of the system has access to the information being presented, however, due to its encoding, an observer may not be able to detect or decode the information to gain a knowledge that the communication of information exists at all. This leads to an inherent relationship between covert channels and steganography where messages are hidden in such a way that an observer is prevented from obtaining a knowledge of the existence of secret data. An important observation that must be made is that in both protocol-based covert channels and environment-based covert channels, it is not the communication between agents that is in violation of a security policy, but rather, the information that is being communicated.

Covert channels pose a threat to system security for a number of reasons. The first reason is, of course, a confidentiality concern, as covert channels can be used to pass critical information secretly. This is a particular concern to large organizations that wish to maintain confidentiality regarding company secrets as covert channel communication allows for this secret information to flow into, or out of, the organization. This is the case with the more recent idea of cloud computing. As organizations are storing huge amounts of data in the cloud, they must ensure that the cloud is secure. From a confidentiality perspective, organizations must use prevention and detection mechanisms to protect their data and secrets from any sort of attack or data leakage. A second reason is an economical concern as covert channels provide a means of transmitting information using an existing system without paying for the service provided. This is often the case when a system is infected by a Trojan Horse. Moreover, covert channels are often based on an obscure use of system resources and functionalities which ultimately reduces the performance of the system. For these reasons, among others, covert channel analysis has become part of the evaluation criteria for the classification of secure systems by the United States Department of Defense (DoD) and the National Computer Security Center (NCSC) as outlined in [DoD85] and [NCSC93].

In this thesis, we focus on protocol-based covert channels as they are often much easier to reason about. There is an abundance of communication protocols that can be exploited to create covert communication channels, many of which are used each and every day. In Chapter 3, we will provide examples of practical protocol-based covert channels to emphasize the eminent need for a method of detecting and preventing the communication of confidential information through communication protocols.

1.3 Literature Survey of Covert Channel Detection and Prevention Techniques

When it comes to eliminating the use of covert channels in computer systems, a variety of approaches have been proposed. Some approaches look at detecting the use of covert channels and some approaches look at preventing the use of covert channels, while there are very few approaches which aim to recover from the effects of covert channel use. Each approach has its own strengths and weaknesses and some are more applicable in real world scenarios than others. In this section, we discuss and assess the strengths and weaknesses of existing approaches to covert channel elimination, and provide insight as to why there is a need for a new technique to detect the use of covert channels.

In [NW06], Nagatou and Watanabe present a technique for detecting the use of covert channels at run time. Nagatou and Watanabe use monitors which watch the state transitions of a system along with an emulator to emulate the behaviour of the system by running a subsequence of states that is restricted from a sequence of states observed by the monitor. Nagatou and Watanabe enforce a security policy through flow control and access control mechanisms. The way in which their technique works is that the flow control mechanism compares the result of each system call into a system resource and the result of the emulator. If the results are different then it is considered that a covert channel occurred in the system and the monitor terminates the process that invoked the infracting system call. This technique is only able to enforce non-interference and non-inference policies. With non-interference and non-inference, computer systems are modelled as machines with inputs and outputs, each classified as either low-level or high-level. A computer system has the *non-interference* property if and only if any sequence of low-level inputs will produce the same low outputs, regardless of what the high-level inputs are [GM82]. A computer system has the *non-inference* property if and only if an adversary cannot infer the value of a high-level output from low-level inputs [RGI06]. The authors also admit that the monitor which they propose does not scale well since it would need to have emulators that have equal security levels and exploit many system resources. This technique also runs the risk of false positives, whereby the result of a system call into a system resource and the result of the emulator are different for reasons other than covert channels such as emulator failure. These noted weaknesses of this technique dramatically reduce the technique's ability to be used in many real world applications. However, the idea of monitoring the communication among agents in the system is a good way to maintain a knowledge of the information flow of the system and we will see, in Section 4.3.1, that this idea will play a large role in the development of the proposed technique in this thesis.

In [Kem83], Kemmerer describes a technique for detecting the use of covert channels in computer systems based on shared resources. His technique is appropriately called the *Shared Resource Matrix* (SRM).The motivation for the SRM technique lies within the knowledge that the use of covert channels requires the collusion between an agent with the authorization to signal or leak information to an unauthorized agent and that the authorization is granted on system objects which may include file locks, device busy flags, the passing of time, etc. The SRM technique is performed in two steps. The first step involves the enumeration of all shared resources that can be referenced or modified by an agent in the system. The second step involves the careful examination of each resource to determine whether it can be used to transfer information from one agent to another in a covert manner. A matrix is constructed where the attributes of all shared resources are indicated in the row headings and the operation primitives, (i.e., `Write File`, `Read File`, `Lock File`, etc.), are indicated in the column headings. After all of the row and column headings are determined, one must determine for each attribute (each row) whether the primitive indicated by the column heading modifies and/or references that attribute. This is done by carefully reviewing the description for each of the primitives, whether it is stated in natural language, formal specification, or implementation code. The generated matrix is then used to determine whether any covert channels exist. Kemmerer provides the following minimum criteria which must be satisfied in order to have a covert channel:

(i) The sending and receiving agents must have access to the same attribute of a shared resource.

(ii) There must be some means by which the sending agent can force the shared attribute to change.

(iii) There must be some means by which the receiving agent can detect the attribute change.

(iv) There must be some mechanism for initiating the communication between the sending and receiving processes and for sequencing the events correctly.

If each of these criteria are satisfied, then a covert channel exists. The advantages of the SRM technique include the ability to quickly discard attributes that do not meet the preliminary criteria of being modified or referenced by an agent and the ability to provide a graphical design for developers in all stages of software design. However, the SRM technique is quite tedious and a little bit ad hoc in that the analyst must decipher scenarios in which the criteria might be satisfied. It is quite difficult to automate the technique which proves to be one of its downfalls.

Another technique for detecting covert channels in computer systems is *Covert Flow Trees* (CFTs). Presented by Kemmerer and Porras in [KP91] and [PK91], CFTs are techniques for detecting covert channels with the goal of identifying operation sequences that support either the direct or indirect ability of an agent to detect when an attribute has been modified. This means that CFTs aid in recognizing when system attributes have been changed in some way by a sequence of operations. Covert Flow Trees can automatically be constructed by providing the algorithm described in [KP91] with information regarding the operations of the system. The operations need to be divided into three lists:

(i) operations which reference the values of attributes during execution,

(ii) operations which modify the values of attributes during execution, and

(iii) operations which return attribute values after execution.

Once the CFT is constructed, the tree can be traversed to develop all possible operation sequences of the system. These operation sequences can then be analyzed by developing hypothetical agents and system states that could use the operation sequences for covert communication. The analyst may assume that the sender and receiver share some mechanism whereby they can synchronize communication. Additionally, a worst-case scenario is assumed, i.e., the analyst may assume that the attributes being used in the scenario are not subject to alteration by other agents. Under these assumptions, the analyst attempts to identify an encoding scheme and system state that would support the flow of information in a manner that violates the system's security policy. The benefits of using CFTs include the ability to generate a comprehensive list of scenarios that could potentially support covert communication and that CFTs graphically illustrate the routes that information travels as it is relayed from attribute to attribute, and eventually detected by the receiver. The downfall of CFTs lies in the size of the CFTs that are generated and the scalability of the approach. Also, more work needs to be done in the area of reducing and expanding the operation sequences produced by the CFT since currently it is an ad hoc process. Due to the process being ad hoc, there is a risk of false positives since a complex hypothetical scenario consisting of numerous agents and systems states can be generated to show a covert channel which is not be possible in the system.

Hélouët, et al. in [HZJ03] and [HZD05], propose a method for detecting potential covert channels using scenarios. The use of scenarios has several advantages in that scenarios are often the first information one can obtain about a system's behaviour since they are used to describe system requirements and that several recommendations [DoD85, NCSC93] ask to document the use of covert channels with such models. The idea is that from a scenario description of a system, a covert channel is modelled as a game where a pair of corrupted users, sender and receiver, (S, R), try to send information while the rest of the protocol is attempting to prevent the information from being communicated. If (S, R) can find a winning strategy in the game and if R can decode the message using a transducer built from the winning strategy, then a potential covert channel exists. There are a number of assumptions made by this approach. The first assumption is that S and R agree on a protocol for sending and receiving covert information. The second assumption is that covert messages can be of arbitrary length, and wc suppose that the same functionality of the diverted protocol is used an arbitrary number of times. Although this approach can immediately offer scenarios for using a potential covert channel and a decoder for covert messages, given as a transducer, it is often the case that model-based studies can miss some covert channels or exhibit unrealistic scenarios. This scenario based approach has the same drawback in that it only reveals "potential covert channels", the existence of which needs to be tested on a real implementation of the protocol.

Another approach to covert channel suppression is to use the concept of separability. According to [Bro94], a system is separable (i.e., multilevel secure) if and only if it is behaviourally equivalent to a collection of single level systems that do not interact. In [Bro94], Browne presents an approach called *Mode Security* as a means of suppressing the use of covert channels. The general idea behind Mode Security is to organize the state transitions of a multilevel state machine into distinct sets called modes (which can be seen as equivalence classes). The aim is to create a separable system. In essence, each machine mode is considered totally secure when considered in isolation of all other modes. This means that covert channels can only occur when the machine makes a transition from one mode to another. Therefore, by reducing the number of mode transitions in the system, one can reduce the number of potential covert channels in the system. Similarly, in [Jac90], Jacob proposes a method for detecting covert channels based on separability as well. The idea is to begin by making a list of all channels in a system. From this list, a new system is produced by "cutting" known channels from the system. This new system is checked for separability. If the new system is separable, then there are no covert channels, otherwise, at least one covert channel exists. The downfall of Jacob's method is that it does not detect covert channels completely dependent on known channels. The major concern with approaches for mitigating covert channels based on separability is that these approaches are not universally applicable to all systems.

In [AR80], Andrews and Reitman take an axiomatic approach to information flow in programs. Andrews and Reitman provide an axiomatic definition for information flow in sequential programs, with particular emphasis on proof rules for programs containing assignment, alternation, iteration, composition, and procedure calls. The definition provided closely resembles Hoare's deductive system for functional correctness found in [Hoa69]. The axiomatic approach of Andrews and Reitman analyzes a program looking for information flows which violate the security policy of the system. A similar approach was taken by Sabri et al. in [SKJ09], where an amended version of Hoare logic was used to verify the satisfiability of security policies in communication protocols. These information flows may be largely a result of the use of a covert channel being used in the program. The downfall to this approach is that it is unclear whether the flow semantics of the language is preserved during execution.

Since the confinement notion introduced by Lampson in [Lam73], more and more approaches to detect illegal information flows have been proposed. A short while after Lampson, in [GM82], Goguen and Meseguer defined the existence of covert channels through non-interference properties. According to Goguen and Meseguer, one group of users, using a certain set of commands, is non-interfering with another group of users if what the first group does with those commands has no effect on what the second group of users can see. By this approach, security verification consists of showing that a given policy is satisfied by a given model. Numerous approaches to non-interference have been proposed. For example, in [VS97], Volpano and Smith describe non-interference through typing where a system contains interference if it cannot be correctly typed and in [Low02], Lowe describes non-interference using process algebra. The notion of non-interference is questioned in [RMMG01] since the transfer of a single bit of information causes a non-interference violation. According to [HZD05], it is often the case that non-interference approaches attempt to classify data and processes of a system according to two security levels: high and low. However, it may not always be the case that there are only two security levels which leads to a fundamental restriction of the use of non-interference properties to define the existence of covert channels in a system.

A wide variety of prevention schemes for the use of covert channels in computer systems have been proposed. The aim of each scheme is to simply reduce the usefulness of covert channels in the system. One such approach is through bandwidth analysis. In [Low02] and [SC99], mechanisms for computing the bandwidth of covert channels in computer systems are presented. The idea behind bandwidth analysis is that if an analyst can reduce the bandwidth of a covert channel to a reasonably small rate, then the channel is rendered unusable as a means of effectively transferring information. The guidelines outlined in [DoD85] and [NCSC93], state that covert channels with bandwidths of less than one bit per second are usually considered acceptable; while a bandwidth of more than 100 bits per second is considered unacceptable. One such method, developed by the United States Naval Research Laboratory, is called the

Pump. Described in [KM93] and [LMST$^+$04], the Pump lets information pass from a low security level system to one at a higher level. The motivation for the Pump comes from the idea that acknowledgements are required for reliable communication and that if a higher level system passed acknowledgements directly to a lower level system, then the higher level system could pass high information by altering acknowledgement delays. In order to minimize such a covert channel, the Pump decouples the acknowledgement stream by inserting random delays. With consideration on overall performance of the system in mind, the Pump uses statistical averages to compute the delay time which it inserts into the communication stream. It is admitted in [LMST$^+$04] that this method cannot handle a large state space which proves to be its major flaw. A number of additional prevention schemes take probabilistic approaches to covert channel mitigation. For example, in [GKT05], Grusho, et al. assume that covert channels will exploit, for secure transmission, a manipulation of the probability distribution parameters of the sent message sequence. The authors suggests that with enough statistical theory, covert channels based on statistics can be detected and prevented using probability models and methods.

Although many techniques already exist which aid in the fight again covert channels, there seems to be no single technique which can handle any type of covert channel in any type of system. In this thesis, we aim to develop investigative support for confidentiality by proposing a new technique to detect the leak of confidential information via covert channels.

1.4 Motivation

It has been a general assumption that it is impossible to completely eliminate covert channels from open systems. Any given open system typically contains several covert communication channels [Gra00]. Many covert channels in computer systems arise from resource sharing. In order to completely eliminate them, one would need to remove all contention for that resource which leads to an inefficient utilization of system resources and an unacceptable reduction in system performance. The detection and prevention of covert channels has been labelled as difficult since the objects that are being used to hold the information being transferred are not normally seen as data objects i.e., buffer size, device flags, the passing of time, etc. A comprehensive and systematic way to model, detect, and prevent the use of covert channels without reducing the performance of the system to an unacceptable level is required.

In addressing covert channel threats, two challenges are distinguished: detection of covert channels and prevention of covert channels. In detecting covert channels, we ought to strive to develop techniques to identify covert channels in a systematic and comprehensive way. We must uncover the use of covert channels efficiently with a minimum number of false positives. We would like to provide some measure of assurance in the detection techniques being used. In covert channel prevention, we should determine ways to remove covert channels or at least find ways to restrict the use of covert channels without degrading the performance of the system to an unacceptable level. We ought to balance the tradeoff between system security and system performance which may not always being the most trivial of decisions.

In this thesis, we are developing investigative support for confidentiality. This involves looking at covert channel communication from a digital forensics perspective. By its very nature, digital forensics is analysis after-the-fact [Sri06]. Hence, the primary focus of a digital forensics investigation is placed on detection, that is, to prove that some form of violation of the security policy has taken place. Therefore, since we are dealing with analysis after-the-fact, performance is not really an issue when developing detection mechanisms for covert channel communication.

There are many challenges that need to be addressed in detecting the use of these covert channels. It leads us to ask a number of questions regarding the abilities of the developers of the covert channels. Consider a protocol-based covert channel. What if the developers of the covert channel encrypt their message with an encryption technique that is very challenging, if not impossible to break before concealing it? What if the developers mask their message with noise by sending a number of random bits of messages before and after the covert message? What if the developers use abstract coding techniques such as *Gray codes*[1] or so-called *baseball codes*[2] for their covert message? What if the developers of the covert channels modulate the order of the information that is being sent, i.e., sending permutations of data?

[1] A Gray code is an encoding of numbers such that adjacent numbers have a single digit differing by 1. Suppose that we wish to represent the angular position (in multiples of 45°) of a continuously rotating shaft. We may encode the angular position by 3-bit binary words as follows: $[0°, 45°) \mapsto 000$, $[45°,90°) \mapsto 001$, $[90°,135°) \mapsto 011$, $[135°,180°) \mapsto 010$, $[180°,225°) \mapsto 110$, $[225°,270°) \mapsto 111$, $[270°,315°) \mapsto 101$, $[315°,360°) \mapsto 100$ [WW90].

[2] Baseball codes refer to the encoding systems that are employed by baseball teams. Often one player on the team may give a sequence of hand gestures to an observing player in order to relay a message to the observing player. In many cases, in order to avoid the opposing team from learning the hand gestures, the signalling player may begin with a series of random gestures followed by a *marking* gesture which indicates that the real signal begins after that gesture.

All of the challenges given above are concerns that must be addressed in order to effectively detect the use of covert channels. We aim to propose mathematical formulations for covert channels and to develop a formal theory for the detection of the leak of confidential information in protocol-based covert channels using algebraic techniques. We do not rule out the possibility that someone may develop heuristics to discover the use of covert channels of various types. However, we are looking to provide a mathematical method which gives a more formal and rigorous way to uncover the use of covert channels. A mathematical method for detecting the use of covert channels gives us the power to uncover the use of covert channels with more flexibility than that which could be done with heuristics. A mathematical method for detecting covert channels also gives us a significant advantage in that we are able to mechanize and automate the computations needed to discover the use of covert channels and to build and configure monitors which are able to supervise a system on which we strive for confidentiality.

As far as I know, after a thorough investigation of the literature, a formal method such as the one we propose is non-existent. This leads to new and fertile grounds for developing a theory of covert channels and provides us with new and innovative ways to approach the problem of covert channels being a threat to the confidentiality of information. Because our society relies so heavily on computers every day, privacy and confidentiality become increasingly important and we wish to aid in the elimination of covert channels which threaten the privacy and confidentiality of information.

1.5 Problem Statement

With the emergence of computers in every day activities and with the ever-growing complexity of networks and network protocols, covert channels are becoming an eminent threat to confidentiality of information. With increasing sensitivity of data in many computer application domains, the leak of confidential information can have severe repercussions on the institution from which the information was leaked.

In light of this eminent threat, we develop a mathematical framework to detect the leak of confidential information via covert channels. We limit our focus to instances where the users of covert channels modulate the information that is being sent, i.e., by encryption, or some other form of encoding.

1.6 Main Contributions

The main contributions to the area of covert channels include:

(i) A new classification for covert channels; the new classification classifies covert channels as either protocol-based covert channels or environment-based covert channels.

(ii) A novel, comprehensive model for protocol-based covert channel communication, including special cases such as steganography and watermarking.

(iii) A technique, based on relation algebra, for detecting the leak of confidential information via covert channels; the technique offers tests for verifying the existence of an abstraction relation which relates the confidential information to the information that is observed to be sent on the communication channel.

1.7 Structure of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 introduces the required mathematical background including set theory and relation algebra.

Chapter 3 provides a number of examples of ways in which confidential information can be leaked via covert channels while discussing the connection between protocol-based covert channels and steganography. A model for protocol-based covert channels is proposed.

Chapter 4 describes the process by which we formulate a new technique to detect the leak of confidential information in protocol-based covert channels.

Chapter 5 gives a number of illustrative examples demonstrating the application of the proposed detection technique in various covert channel scenarios such as the modulation of confidential information.

Chapter 6 discusses the impact of our approach in helping to remedy the problem of covert channels. It discusses the possible applications of the proposed detection technique as a tool to aid in computer forensics investigations. This chapter also provides an assessment of the strengths and weaknesses of our contributions.

Chapter 7 draws conclusions and suggests future work.

Chapter 2

Mathematical Background

In this chapter, we introduce the necessary mathematical concepts required for the understanding of the material presented in the thesis. In Section 2.1, we give an introduction to sets and its associated theory. In Section 2.2, we give definitions and examples of relations, properties of relations and operations on relations. Finally, in Section 2.3, we conclude with a summary of the core concepts and describe where they are used throughout the remainder of the thesis.

2.1 Set Theory

We aim to detect the use of covert channels through the communication of agents in a system. We represent the stream of messages sent between agents in a system using relations. Therefore, since relations are defined generally in terms of sets, we first give a general introduction to set theory. The material regarding set theory is extracted from [GS93].

We begin by defining a set.

Definition 2.1.1. *A set is a collection of distinct elements.*

There are two ways to describe a set. The first is called set enumeration, which describes a set by listing its elements. For example, $\{1, 8, 27, 64\}$ denotes the set consisting of the elements 1, 8, 27, and 64. The second is called set comprehension, which describes a set by stating properties shared by its elements. For example, the set comprehension $\{x \in \mathbb{N} \mid \exists(y \mid y \in \mathbb{N} \land 1 \leq y \leq 4 : x = y^3)\}$ denotes the set of all natural numbers x such that $\exists(y \mid y \in \mathbb{N} \land 1 \leq y \leq 4 : x = y^3)$ is satisfied.

We identify two special sets: the universal set and the empty set.

Definition 2.1.2.

(i) The universal set, denoted by U, is the set fixed within the framework of a theory and consisting of all objects considered in this theory.

(ii) The set $\emptyset$ is called the empty set and is defined by

$$\emptyset \stackrel{def}{=} \{x \mid \mathsf{false}\}$$

The following are a selection of useful operations on sets.

Definition 2.1.3. *Let S and T be sets and let U be the universal set.*

(i) Size (Cardinality): $\#S = \sum(x \mid x \in S : 1)$

(ii) Subset: $S \subseteq T \iff \forall(x \mid x \in S : x \in T)$

(iii) Proper Subset: $S \subset T \iff S \subseteq T \wedge S \neq T$

(iv) Superset: $T \supseteq S \iff S \subseteq T$

(v) Proper Superset: $T \supset S \iff S \subset T$

(vi) Complement: $x \in \overline{S} \iff x \in U \wedge x \notin S$

(vii) Union (Join): $x \in S \cup T \iff x \in S \vee x \in T$

(viii) Intersection (Meet): $x \in S \cap T \iff x \in S \wedge x \in T$

(ix) Difference: $x \in S - T \iff x \in S \wedge x \notin T$

(x) Power Set: $x \in \mathcal{P}(S) \iff x \subseteq S$

2.2 Relation Algebra

Definition 2.2.1. *Given two sets, A and B, we define the Cartesian product $A \times B$ as*

$$A \times B = \{(x, y) \mid x \in A \wedge y \in B\}$$

We introduce the notion of relations. In natural language, the word "relation" indicates a connectedness between two or more things [SS93]. We define relations in terms of set theory as given in [SS93].

Definition 2.2.2. *Let A and B be two sets. A relation R on $A \times B$ is a subset of the Cartesian product $A \times B$, that is, $R \subseteq A \times B$.*

When $A = B$ we say that R is a homogenous relation and when $A \neq B$ we say that R is a heterogeneous relation.

We identify three special relations: the identity relation, the universal relation, and the empty relation.

Definition 2.2.3.

(i) For every set A, the relation $\mathbb{I}$ on $A \times A$ is called the identity relation on A and is defined by

$$\mathbb{I} \overset{def}{=} \{(x,x) \mid x \in A\}$$

(ii) For every two sets A and B, the relation $\mathbb{L}$ on $A \times B$ is called the universal relation and is defined by

$$\mathbb{L} \overset{def}{=} \{(x,y) \mid \mathsf{true}\}$$

(iii) For every two sets A and B, the relation $\emptyset$ is called the empty relation and is defined by

$$\emptyset \overset{def}{=} \{(x,y) \mid \mathsf{false}\}$$

We define the domain of a relation and the range of a relation as in [SS93].

Definition 2.2.4. *Let $R \subseteq X \times Y$ be a relation:*

(i) The domain *of the relation R is given by*

$$\mathit{dom}(R) = \{x \mid \exists(y \mid y \in Y : (x,y) \in R)\}$$

(ii) The range *of the relation R is given by*

$$\mathit{ran}(R) = \{y \mid \exists(x \mid x \in X : (x,y) \in R)\}$$

There are three important operations on relations that are needed to continue: composition, converse and complement. Each of the definitions of the composition of relations, the converse of a relation and the complement of a relation are taken from [SS93].

Definition 2.2.5. *Let $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ be relations. Then, their composition $R;S$ is defined by*

$$R;S = \{(x,z) \mid \exists(y \mid y \in Y : (x,y) \in R \wedge (y,z) \in S)\}$$

Definition 2.2.6. *We define the converse of a relation R as follows*

$$R^{\smallsmile} = \{(x,y) \mid (y,x) \in R\}$$

Definition 2.2.7. *We define the complement of a relation R as follows*

$$\overline{R} = \{(x,y) \mid (x,y) \notin R\}$$

The next definition introduces total, univalent, surjective, injective, and mapping relations. A graphical example of each type of relation is given in Figure 2.1. The definition is taken from [SS93].

Definition 2.2.8. *If R is a relation, then we say*

(i) R is total

$\iff \mathbb{L} = R;\mathbb{L} \iff \mathbb{I} \subseteq R;R^{\smile} \iff \overline{R} \subseteq R;\overline{\mathbb{I}}$

$\iff \forall(S \mid S;R = \emptyset : S = \emptyset)$

$\iff \forall(S \mid: R;\overline{S} \supseteq \overline{R;S})$

(ii) R is univalent (deterministic or functional)

$\iff R^{\smile};R \subseteq \mathbb{I} \iff R;\overline{\mathbb{I}} \subseteq \overline{R}$

$\iff \forall(S \mid: R;\overline{S} \subseteq \overline{R;S})$

$\iff \forall(S \mid: \overline{R;S} = R;\overline{S} \cup \overline{R;\mathbb{L}})$

(iii) R is surjective

$\iff R^{\smile}$ *total*

$\iff \mathbb{L} = \mathbb{L};R \iff \mathbb{I} \subseteq R^{\smile};R \iff R;\overline{\mathbb{I}} \subseteq \overline{R}$

$\iff \forall(S \mid R;S = \emptyset : S = \emptyset)$

(iv) R is injective

$\iff R^{\smile}$ *univalent*

$\iff R;R^{\smile} \subseteq \mathbb{I} \iff \overline{\mathbb{I}};R \subseteq \overline{R}$

(v) R is a mapping

$\iff$ *R total and univalent* $\iff R;\overline{\mathbb{I}} = \overline{R}$

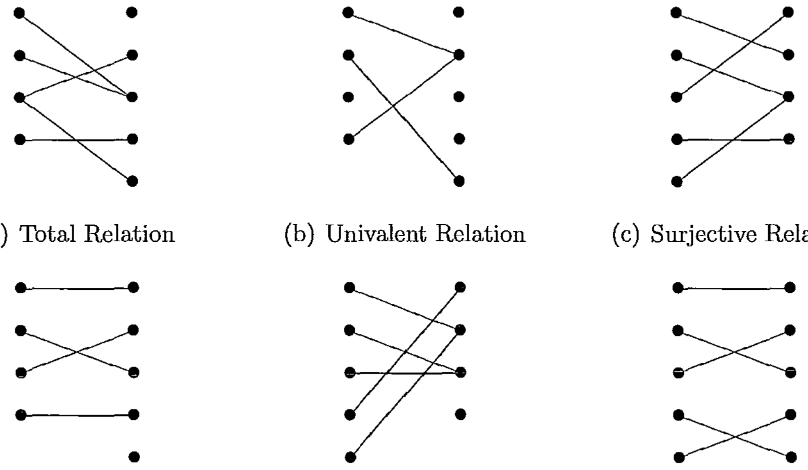
$\iff \forall(S \mid: R;\overline{S} = \overline{R;S})$

If R is univalent, it is also called right-univalent or a partially defined function.

If R is injective, it is also called left-univalent.

If R is surjective and injective, it is also called bijective.

If R is mapping, it is also called a totally defined function.



(a) Total Relation (b) Univalent Relation (c) Surjective Relation

(d) Injective Relation (e) Mapping Relation (f) Bijective Relation

Figure 2.1: Types of Relations.

Some important properties of relations from [SS93], which are used throughout this thesis are given below.

Proposition 2.2.1. *Let P and Q be relations.*

(i) $\overline{\overline{P}} = P$

(ii) $P^{\smile\smile} = P$

(iii) $\overline{(P \cup Q)} = \overline{P} \cap \overline{Q}$

(iv) $\overline{(P \cap Q)} = \overline{P} \cup \overline{Q}$

(v) $(P \cup Q)^{\smile} = P^{\smile} \cup Q^{\smile}$

(vi) $(P \cap Q)^{\smile} = P^{\smile} \cap Q^{\smile}$

(vii) $(P \mathbin{;} Q)^{\smile} = Q^{\smile} \mathbin{;} P^{\smile}$

The interplay between relational composition, converse, and complement with respect to containment is given by the Schröder equivalences. The definition of the Schröder equivalences is taken from [SS93].

Proposition 2.2.2. *Let P, Q and R be relations. Then,*

$$P \mathbin{;} Q \subseteq R \quad \Longleftrightarrow \quad P^{\smile} \mathbin{;} \overline{R} \subseteq \overline{Q} \quad \Longleftrightarrow \quad \overline{R} \mathbin{;} Q^{\smile} \subseteq \overline{P}$$

Proof. The proof can be found in [SS93]. □

We introduce the notion of residue. Residue is a special operation on relations. It helps solve equations of the form $P \mathbin{;} X = Q$ or $X \mathbin{;} P = Q$. Definition 2.2.9 is taken from [SS93].

Definition 2.2.9. *Let P and Q be two relations.*

(i) $P\backslash Q \overset{\text{def}}{=} \overline{P^{\smile} ; \overline{Q}}$ *is said to be the right residue of Q by P;*

(ii) $Q/P \overset{\text{def}}{=} \overline{\overline{Q} ; P^{\smile}}$ *is said to be the left residue of Q by P.*

The left residue and the right residue are also called, in [HH86a, HH86b], *weakest prespecification* and *weakest postspecification*, respectively.

Example 2.2.1. *Let $A = \{a, b\}$, $P \subseteq A \times A$ such that $P = \{(a,a),(b,a)\}$, and $Q \subseteq A \times A$ such that $Q = \{(a,b),(b,b)\}$. In this case the universe of values is $A \times A$.*

We first compute $P\backslash Q$.

$$
\begin{aligned}
& P\backslash Q \\
= & \qquad \langle \textit{ Definition 2.2.9(i) } \rangle \\
& \overline{P^{\smile} ; \overline{Q}} \\
= & \qquad \langle \textit{ Substitution: } P^{\smile} = \{(a,a),(a,b)\} \textit{ and } \overline{Q} = \{(a,a),(b,a)\} \ \rangle \\
& \overline{\{(a,a),(a,b)\} ; \{(a,a),(b,a)\}} \\
= & \qquad \langle \textit{ Definition 2.2.5 } \rangle \\
& \overline{\{(a,a)\}} \\
= & \qquad \langle \textit{ Definition 2.2.7 } \rangle \\
& \{(a,b),(b,a),(b,b)\}
\end{aligned}
$$

Next, we compute Q/P.

$$
\begin{aligned}
& Q/P \\
= \quad & \langle \textit{ Definition 2.2.9(ii) } \rangle \\
& \overline{\overline{Q} ; P^{\smile}} \\
= \quad & \langle \textit{ Substitution: } P^{\smile} = \{(a,a),(a,b)\} \textit{ and } \overline{Q} = \{(a,a),(b,a)\} \rangle \\
& \overline{\{(a,a),(b,a)\} ; \{(a,a),(a,b)\}} \\
= \quad & \langle \textit{ Definition 2.2.5 } \rangle \\
& \overline{\{(a,a),(a,b),(b,a),(b,b)\}} \\
= \quad & \langle \textit{ Definition 2.2.7 } \rangle \\
& \emptyset
\end{aligned}
$$

The left residue constitutes the greatest solution to $X ; P \subseteq Q$ (see Proposition 2.2.3 (i)). A solution to $X ; P \subseteq Q$ is any relation, X, such that the equation is satisfied. If the equation $X ; P = Q$ has a solution (i.e., when $\mathsf{ran}(P) = \mathsf{ran}(Q)$, $X ; P = Q$ has a solution), the left residue is its greatest solution. We illustrate this using Example 2.2.2. P/Q, P, and Q are presented as graphs in Figure 2.2. We can see that $\mathsf{ran}(P) \cap \mathsf{ran}(Q) = \emptyset$ since $\mathsf{ran}(P) = \{a\}$ and $\mathsf{ran}(Q) = \{b\}$. Therefore, we cannot find a relation $X \neq \emptyset$ such that $X ; P = Q$. Only $X = \emptyset$ satisfy $X ; P \subseteq Q$.

Example 2.2.2.

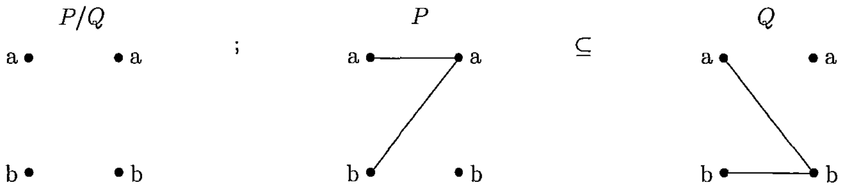


Figure 2.2: $P/Q\,;P \subseteq Q$ with $P/Q = \emptyset$

The right residue constitutes the greatest solution to $P\,;X \subseteq Q$ (see Proposition 2.2.3 (ii)) and also is the greatest solution to $P\,;X = Q$ if this equation is solvable (i.e., when $\mathsf{dom}\,(P) = \mathsf{dom}\,(Q)$, $P\,;X = Q$ has a solution). We illustrate this in Example 2.2.3. P, $P\backslash Q$, and Q are presented as graphs in Figure 2.3. From these graphs, we can see that $P\,;P\backslash Q = Q$. So $P\backslash Q$ is the solution of the equation $P\,;X = Q$, i.e., $X = P\backslash Q$.

Example 2.2.3.

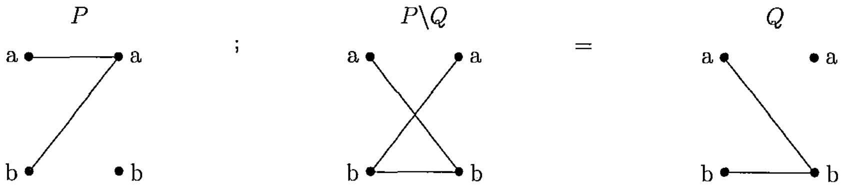


Figure 2.3: $P\,;P\backslash Q = Q$

Proposition 2.2.3. *Let P, Q and X be relations.*

(i) $X \mathbin{;} P \subseteq Q \iff X \subseteq Q/P$,

(ii) $P \mathbin{;} X \subseteq Q \iff X \subseteq P\backslash Q$.

Proof. The proof can be found in [Khe98, SS93]. □

Some important properties of residues which are used throughout this thesis are taken from [FK98] and are given below.

Proposition 2.2.4. *For relations P, Q, and R we have*

(i) $(P/Q)^{\smile} = Q^{\smile}\backslash P^{\smile}$

(ii) $(P\backslash Q)^{\smile} = Q^{\smile}/P^{\smile}$

(iii) $(P \cap Q)/R = P/R \cap Q/R$

(iv) $R\backslash(P \cap Q) = R\backslash P \cap R\backslash Q$

(v) $P \subseteq (P \mathbin{;} Q)/Q$

(vi) $P \subseteq Q\backslash(Q \mathbin{;} P)$

(vii) $(P/Q) \mathbin{;} Q \subseteq P$

(viii) $Q \mathbin{;} (Q\backslash P) \subseteq P$

In some cases, it is required that a relation be a left reside and right residue simultaneously. This notion is called the *symmetric quotient*. The definition is taken from [SS93].

Definition 2.2.10. *If P and Q are relations, we define the* symmetric quotient *as*

$$\mathsf{syq}(P,Q) \overset{def}{=} \overline{P^{\smile} ; \overline{Q}} \cap \overline{\overline{P^{\smile}} ; Q} = (P\backslash Q) \cap (P^{\smile}/Q^{\smile})$$

The symmetric quotient $\mathsf{syq}(P,Q)$ of two relations P and Q is defined as the greatest relation X such that $P ; X \subseteq Q$ and $X ; Q^{\smile} \subseteq P^{\smile}$.

Some important results used in the remainder of this thesis regarding the properties of relations and residues are given in Proposition 2.2.5.

Proposition 2.2.5. *Let P and Q be relations.*

(i) P is a bijection $\wedge$ Q is surjective $\Longrightarrow$ $P\backslash Q$ is surjective

(ii) $P\backslash Q = (Q\backslash P)^{\smile} \Longrightarrow P \subseteq Q ; (Q\backslash P)$ for Q a bijection and P surjective

(iii) $P \subseteq Q ; (Q\backslash P) \;\wedge\; Q \subseteq P ; (P\backslash Q) \Longleftrightarrow P\backslash Q = (Q\backslash P)^{\smile}$ for P and Q bijections

Proof.

(i) $P\backslash Q$ is surjective

$\Longleftrightarrow$ $\langle$ Formalization $\rangle$

$\mathbb{L} = \mathbb{L} ; (P\backslash Q)$

$\iff$ $\langle$ Definition 2.2.9(i) $\rangle$

$\mathbb{L} = \mathbb{L} ; \overline{P^{\smile} ; \overline{Q}}$

$\iff$ $\langle$ P is a bijection $\iff P^{\smile}$ is a mapping $\iff P^{\smile} ; \overline{S} = \overline{P^{\smile} ; S}$ for all S $\rangle$

$\mathbb{L} = \mathbb{L} ; \overline{\overline{P^{\smile} ; Q}}$

$\iff$ $\langle$ Proposition 2.2.1(i) $\rangle$

$\mathbb{L} = \mathbb{L} ; P^{\smile} ; Q$

$\iff$ $\langle$ P is total $\iff$ $P ; \mathbb{L} = \mathbb{L}$ $\iff$ $\mathbb{L} ; P^{\smile} = \mathbb{L}$ $\rangle$

$\mathbb{L} = \mathbb{L} ; Q$

$\iff$ $\langle$ Q is surjective $\iff$ $\mathbb{L} ; Q = \mathbb{L}$ $\rangle$

$\mathbb{L} = \mathbb{L}$

$\iff$ $\langle$ Identity of $=$ $\rangle$

true

(ii) $P \backslash Q = (Q \backslash P)^{\smile}$

$\iff$ $\langle$ $(Q \backslash P)^{\smile} = P^{\smile} / Q^{\smile}$ $\rangle$

$P \backslash Q = P^{\smile} / Q^{\smile}$

$\iff$ $\langle$ Definition 2.2.9(i) & Definition 2.2.9(ii) $\rangle$

$\overline{P^{\smile} ; \overline{Q}} = \overline{\overline{P^{\smile}} ; Q}$

$\iff$ $\langle$ Complement both sides & Proposition 2.2.1(i) $\rangle$

$P^{\smile} ; \overline{Q} = \overline{P^{\smile}} ; Q$

$\iff$ $\langle$ Antisymmetry $\rangle$

$$P^{\smile};\overline{Q} \subseteq \overline{P^{\smile}};Q \;\wedge\; \overline{P^{\smile}};Q \subseteq P^{\smile};\overline{Q}$$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.3(i) $\rangle$

$$P^{\smile} \subseteq (\overline{P^{\smile}};Q)/\overline{Q} \;\wedge\; \overline{P^{\smile}} \subseteq (P^{\smile};\overline{Q})/Q$$

$\Longleftrightarrow$ $\langle$ Hypothesis: $P\backslash Q = (Q\backslash P)^{\smile} \Longleftrightarrow P^{\smile};\overline{Q} = \overline{P^{\smile}};Q$ $\rangle$

$$P^{\smile} \subseteq (\overline{P^{\smile}};Q)/\overline{Q} \;\wedge\; \overline{P^{\smile}} \subseteq (\overline{P^{\smile}};Q)/Q$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) &
Complement both sides & Proposition 2.2.1(i) $\rangle$

$$P \subseteq [(\overline{P^{\smile}};Q)/\overline{Q}]^{\smile} \;\wedge\; [\overline{(\overline{P^{\smile}};Q)/Q}] \subseteq P^{\smile}$$

$\Longleftrightarrow$ $\langle$ Definition 2.2.9(ii) $\rangle$

$$P \subseteq [\overline{\overline{\overline{P^{\smile}};Q};\overline{Q^{\smile}}}]^{\smile} \;\wedge\; [\overline{\overline{\overline{\overline{P^{\smile}};Q};Q^{\smile}}}] \subseteq P^{\smile}$$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.1(i) & Proposition 2.2.1(vii) $\rangle$

$$P \subseteq \overline{\overline{Q};\overline{\overline{Q^{\smile};\overline{P}}}} \;\wedge\; \overline{\overline{P^{\smile}};Q};Q^{\smile} \subseteq P^{\smile}$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) &
Proposition 2.2.1(vii) $\rangle$

$$P \subseteq \overline{\overline{Q};\overline{Q^{\smile};\overline{\overline{P}}}} \;\wedge\; Q;\overline{Q^{\smile};\overline{\overline{P}}} \subseteq P$$

$\Longleftrightarrow$ $\langle$ $Q\backslash P$ is surjective $\Longleftrightarrow \overline{Q;\overline{Q^{\smile};\overline{\overline{P}}}} \subseteq \overline{Q};\overline{Q^{\smile};\overline{P}} \Longleftrightarrow \overline{\overline{Q};\overline{Q^{\smile};\overline{\overline{P}}}} \subseteq$
$\overline{\overline{Q;\overline{Q^{\smile};\overline{\overline{P}}}}}$ $\rangle$

$$P \subseteq \overline{\overline{Q};\overline{Q^{\smile};\overline{\overline{P}}}} \subseteq \overline{\overline{Q;\overline{Q^{\smile};\overline{\overline{P}}}}} \;\wedge\; Q;\overline{Q^{\smile};\overline{\overline{P}}} \subseteq P$$

$\Longrightarrow$ $\langle$ Proposition 2.2.1(i) & Transitivity of $\subseteq$ $\rangle$

$$P \subseteq Q;\overline{Q^{\smile};\overline{\overline{P}}} \;\wedge\; Q;\overline{Q^{\smile};\overline{\overline{P}}} \subseteq P$$

$\Longleftrightarrow$ $\langle$ Definition 2.2.9(i) $\rangle$

$$P \subseteq Q;(Q\backslash P) \;\wedge\; Q;(Q\backslash P) \subseteq P$$

$\iff$ ⟨ Proposition 2.2.4(viii) & Identity of $\wedge$ ⟩

$P \subseteq Q ; (Q \backslash P)$

(iii) $P \backslash Q = (Q \backslash P)^\smile$

$\iff$ ⟨ Proposition 2.2.4(ii) ⟩

$P \backslash Q = P^\smile / Q^\smile$

$\iff$ ⟨ Definition 2.2.9(i) & Definition 2.2.9(ii) ⟩

$\overline{P^\smile ; \overline{Q}} = \overline{\overline{P^\smile} ; Q}$

$\iff$ ⟨ Complement both sides & Proposition 2.2.1(i) ⟩

$P^\smile ; \overline{Q} = \overline{P^\smile} ; Q$

$\iff$ ⟨ Antisymmetry ⟩

$P^\smile ; \overline{Q} \subseteq \overline{P^\smile} ; Q \ \wedge \ \overline{P^\smile} ; Q \subseteq P^\smile ; \overline{Q}$

$\iff$ ⟨ Proposition 2.2.3(ii) ⟩

$\overline{Q} \subseteq P^\smile \backslash (\overline{P^\smile} ; Q) \ \wedge \ Q \subseteq \overline{P^\smile} \backslash (P^\smile ; \overline{Q})$

$\iff$ ⟨ Definition 2.2.9(i) & Proposition 2.2.1(ii) ⟩

$\overline{Q} \subseteq \overline{P ; \overline{\overline{P^\smile} ; Q}} \ \wedge \ Q \subseteq \overline{\overline{P} ; \overline{P^\smile ; \overline{Q}}}$

$\iff$ ⟨ Complement both sides & Proposition 2.2.1(i) ⟩

$P ; \overline{\overline{P^\smile} ; Q} \subseteq Q \ \wedge \ Q \subseteq \overline{\overline{P} ; \overline{P^\smile ; \overline{Q}}}$

$\iff$ ⟨ Definition 2.2.9(i) & Definition 2.2.9(ii) ⟩

$P ; (P^\smile / Q^\smile) \subseteq Q \ \wedge \ Q \subseteq \overline{\overline{P} ; (P \backslash Q)}$

$\iff$ $\langle$ P is a bijection $\wedge$ Q is surjective $\Longrightarrow$ $P\backslash Q$ is surjective & $P\backslash Q$ is surjective $\iff$ $\overline{\overline{P};(P\backslash Q)} \subseteq \overline{\overline{P}};(P\backslash Q)$ $\rangle$

$P;(P^{\smile}/Q^{\smile}) \subseteq Q \ \wedge\ Q \subseteq \overline{\overline{P};(P\backslash Q)} \subseteq \overline{\overline{P}};(P\backslash Q)$

$\iff$ $\langle$ Proposition 2.2.1(i) & Transitivity of $\subseteq$ $\rangle$

$P;(P^{\smile}/Q^{\smile}) \subseteq Q \ \wedge\ Q \subseteq P;(P\backslash Q)$

$\iff$ $\langle$ Hypothesis: $Q \subseteq P;(P\backslash Q)$ $\rangle$

$P;(P^{\smile}/Q^{\smile}) \subseteq Q \ \wedge$ true

$\iff$ $\langle$ Identity of $\wedge$ & Proposition 2.2.3(i) $\rangle$

$P \subseteq Q/(P^{\smile}/Q^{\smile})$

$\iff$ $\langle$ Definition 2.2.9(ii) $\rangle$

$P \subseteq \overline{\overline{Q};(P^{\smile}/Q^{\smile})^{\smile}}$

$\iff$ $\langle$ Proposition 2.2.4(i) $\rangle$

$P \subseteq \overline{\overline{Q};(Q\backslash P)}$

$\iff$ $\langle$ Q is a bijection $\wedge$ P is surjective $\Longrightarrow$ $Q\backslash P$ is surjective & $Q\backslash P$ is surjective $\iff$ $\overline{\overline{Q};(Q\backslash P)} \subseteq \overline{\overline{Q}};(Q\backslash P)$ $\rangle$

$P \subseteq \overline{\overline{Q};(Q\backslash P)} \subseteq \overline{\overline{Q}};(Q\backslash P)$

$\iff$ $\langle$ Proposition 2.2.1(i) & Transitivity of $\subseteq$ $\rangle$

$P \subseteq Q;(Q\backslash P)$

$\iff$ $\langle$ Hypothesis: $P \subseteq Q;(Q\backslash P)$ $\rangle$

true

□

2.3 Conclusion

The objective of this chapter is to give readers the required mathematical background of our approach. We have presented set theory and relation algebra since we will be representing streams of communicated information as relations whereby the definition of a relation is defined using sets. The notion of residue will play an important role in defining tests to detect the leak of information via covert channels and will be discussed further in Chapter 4.

Chapter 3

Survey of Covert Channels

In this chapter, we aim to provide some insight as to how covert channels can be created and used to smuggle confidential information into, or out of an organization. Understanding how covert channels are established and used to secretly communicate information is a significant step in developing a formal technique to detect the use of covert channels in computer systems. In Section 3.1, I give a short discussion of the debate regarding the relationship between covert channel communication and steganography. In Sections 3.2 and 3.3, I present a non-exhaustive summary of known techniques to establish covert channels over several common protocols. In Section 3.4 we discuss how a relationship between covert channel communication and steganography can be seen through the techniques presented.

3.1 Covert Channels vs. Steganography

According to the literature, [Ber07, BR05, GKT05, HZD05, PSCS07, PAK99, Sar06, ZAB07], there has been a debate whether a relationship between steganography and covert channel communication exists. In [BR05], Bidou and Raynal suggest that there is a distinction between steganography and covert channel communication. It is argued that, in the case of steganography, the communication channel is known, so it is concluded that there is no relationship with covert channel communication. Steganography merely hides information into some form of cover such as images, audio, video, etc. However, in the case of covert channel communication, not only is the information hidden in some form of cover, but the communication channel itself is also hidden in some way. In the literature, the counterargument is also present, arguing that steganography is simply a special case of covert channel communication. In [PSCS07], Patel et al. claim that steganography is an example of covert channel communication. Also, in [ZAB07], Zander et al. discuss the similarities of covert channel communication and steganography. Both concepts require some form of cover to hide information and to carry the information to its destination. The debate as to whether there is a relationship between steganography and covert channel communication is ongoing; however, we intend to show that a relationship between the two concepts does in fact exist.

The debate that exists regarding the relationship between covert channel communication and steganography is an indication that there is no concrete understanding of covert channel communication. There is much confusion in the area of covert channels. There is a need for a better understanding of covert channel communication and the implications of its existence as a security concern.

3.2 Network Protocols

In this section, I present a number of covert channeling techniques which exploit several common network protocols. Covert channeling techniques which employ network protocols as covert message carriers are very popular and are among the easiest to use.

3.2.1 IP: Internet Protocol

The *Internet Protocol* (IP) is the most commonly used protocol in the network layer, which is responsible for routing packets for delivery. It is used to move all traffic across the Internet. IP is distributed widely across the globe, providing both public and private networking connectivity over packet-switching networks [TJ10]. It is an interesting carrier for covert channels as data can be hidden in the header of an IP datagram. In Figure 3.1, we illustrate an IP datagram. The grey fields in the figure are the fields of interest for hiding data for use in a covert channel. The figure is borrowed from [Com05].

IP Identification

The first field which can be used to conceal information for covert communication in an IP datagram header is the 16-bit IP *Identification* field. It is used to uniquely identify an IP datagram within a flow of datagrams that share the same source and destination. In [SK06], it is shown that since the value for the IP *Identification* field

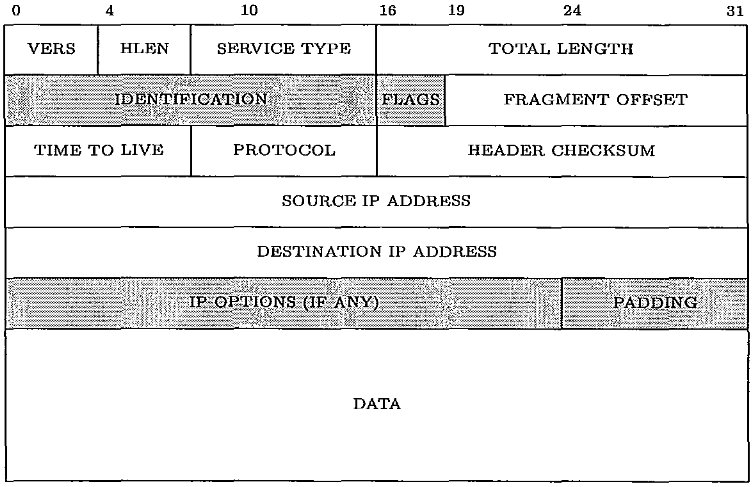


Figure 3.1: Format of an IP datagram, the basic unit of transfer in a TCP/IP internet.

should be chosen at random, it is possible to choose a non-random value for the IP *Identification* field without interrupting the IP mechanism. By using a non-random value for the *Identification* field, 16 bits of covert data can be sent to any other networked system.

IP Flags

The IP *Flags* field is a 3-bit field used to handle fragmentation issues. The IP *Flags* field is optional, meaning that whether the IP *Flags* are set or not make no large scale difference in the IP mechanism. Therefore, as shown in [SK06], the IP *Flags* can be set or unset to carry a covert message, despite the fact that it can only transit at most 3 bits per datagram.

IP Options

The IP *Options* field is optional and most IP datagrams do not make use of it [TJ10]. IP *Options* provide control functions that are useful in some situations but are unnecessary for the most common forms of communication over IP. According to [SK06], the 24-bit options, since they are seldom used in the common IP communication, can be used to transmit data to other networks.

IP Padding

The IP *Padding* field is an 8-bit field used to pad the IP *Options* to construct a 32-bit word in the IP datagram header. Generally speaking, padding should contain only zeros, but as outlined in [SK06], the IP *Padding* can be used to transmit covert information by sending 8 bits of the covert message rather than padding with zeros.

Because of the popularity of IP in today's society, IP provides numerous opportunities to develop and establish covert channels for communication. The vastness of the Internet Protocol and its ability to harbour covert channels leads to a seemingly unsurmountable task of eliminating covert channels which use IP as a covert message carrier.

3.2.2 TCP: Transmission Control Protocol

The *Transmission Control Protocol* (TCP) is one of the core protocols of the Internet Protocol (IP) Suite. TCP is a transport-layer protocol that allows for reliable data transmission. TCP is a suitable carrier for covert channels since data can be hidden in the TCP segment header. In Figure 3.2, we illustrate a TCP segment. The grey fields in the figure are the fields of interest for hiding data for use in a covert channel. The figure is borrowed from [Com05].

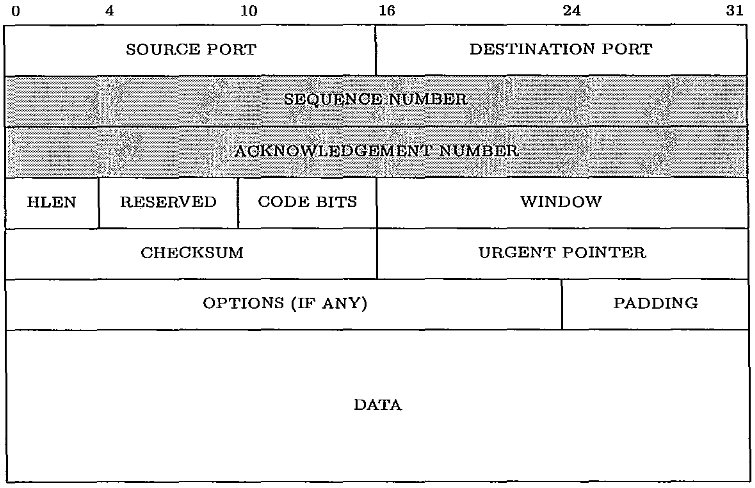


Figure 3.2: Format of a TCP segment with a TCP header followed by data.

SYN/ACK Bounce

Bouncing techniques take advantage of third party equipment that will relay information between the source and destination. Furthermore, the bouncing host is not necessarily aware that it is being used in such a way. According to [BR05], the principle of bouncing can be applied to synchronization (SYN) segments and acknowledgement (ACK) segments under the TCP mechanism. A TCP connection is established by a three way handshake as described in [Com05] where:

(i) The client transmits a segment with the SYN bit but not the ACK bit set. The segment contains the clients initial sequence number, x.

(ii) The server acknowledges the SYN segment by transmitting a segment with both the SYN and ACK bits set. The segment contains the server's initial sequence number y. The segment's acknowledgment number is $x + 1$.

(iii) The client acknowledges the SYN/ACK segment by transmitting a segment with the ACK bit but not the SYN bit set. The segment's acknowledgment value is $y + 1$.

According to [BR05], because of this standard behaviour, it is possible to transmit information by using any server and spoofing the source IP address so that it points to the intended receiver on the other end of the covert channel. For example, suppose that agent A and agent B wish to communicate by using a SYN/ACK bounce covert channel through a server S in order to transmit the message "`service ssh start\n`", which is 18 bytes long and encoded in hexadecimal as (7365 7276 6963 6520 7373 6820 7374 6172 740A). Using the 32-bit sequence number field of the TCP segment, this message will be divided in 5 blocks:

- Sequence 1 : 0x73657276
- Sequence 2 : 0x69636520
- Sequence 3 : 0x73736820
- Sequence 4 : 0x73746172
- Sequence 5 : 0x740A0000

where in Sequence 5, two null bytes have been used for padding. In order to establish the channel, 5 SYN segments need to be sent to the server S on port 80, with the address of B as a source address and with the sequence numbers that have been calculated above. S will then send the SYN/ACK segments to the spoofed source

B where, in these SYN/ACK segments, the acknowledgement sequence numbers will be respectively 0x73657277, 0x69636521, 0x73736821, 0x73746173, and 0x740A0001, which are the previous sequence numbers sent by A and increased by 1. Reassembly on the part of B is then trivial to regain the message.

TCP Sequence Number

A suitable TCP header field used for establishing covert channels is the TCP *Sequence Number* field. The TCP *Sequence Number* is a 32-bit field that provides a means for segment (re)ordering on the arrival of the segment at the receiver. The TCP *Sequence Number* offers reliability and the ability for retransmission of individual segments. Under normal circumstances, when a TCP connection is established, the first TCP segment to be sent (a SYN segment) contains a random initial sequence number (ISN). The receiver generally acknowledges the receipt of the first segment with a SYN/ACK segment where the sequence number field contains ISN+1. According to [SK06], the TCP *Sequence Number* field can be used to transmit 32 bits of covert information. Rather than using a random value for the ISN, a user is able encode 32 bits of covert information in the TCP *Sequence Number* field and send it to any other system on the network without interrupting the TCP mechanism.

TCP Acknowledgement Number

Another suitable TCP header field used for establishing covert communication is the TCP *Acknowledgement Number* field. The 32-bit TCP *Acknowledgement Number* field is used to acknowledge the receipt of a TCP segment to its source. The TCP *Acknowledgement Number* field must always contain the sequence number of the sender incremented by 1. According to [SK06], the TCP *Acknowledgement Number* field can be used for covert communication if the source of a TCP segment is spoofed, allowing for the receiving host to acknowledge to an arbitrary host with the input bytes encoded into the acknowledgement number field.

The use of TCP as a carrier of covert information is a remarkable choice due to the widespread use of TCP in modern day computer networks. Nearly every computer in today's society uses TCP to connect to the Internet, which leads to an enormous scope of possibilities for covert channels to be established worldwide.

3.2.3 ICMP: Internet Control Message Protocol

The *Internet Control Message Protocol* (ICMP) is a network-layer protocol that is used for generating error, test, and informational messages related to IP-based communication on the Internet. The availability of ICMP is critical for the diagnosis of network problems and for regular IP-based networking.

ICMP Error Bounce

Another bouncing technique can be applied to ICMP error messages. For ICMP destination/port unreachable, ICMP time exceeded, and ICMP redirect error message types, it is specified in [Com05], that the data field of the packet must contain the IP header as well as the first 48 bytes of data from the packet that generated the error. According to [BR05], if we have two agents, A and B wishing to establish a covert channel by bouncing off of an agent, C, the channel can be established by the following protocol:

(i) A sends a packet that will generate an error. The source address of this packet is spoofed to be the address of B and the covert data that is to be transmitted can be either in the IP header or in the first 48 bytes of the IP data.

(ii) C identifies the error and sends the appropriate ICMP message to the source address which is now the address of B.

(iii) B receives the packet and gets the data.

ICMP Echo Request/Reply

The format of an ICMP echo request and reply message is given in Figure 3.3. The grey field is the field of interest for hiding data for use in a covert channel. The figure is borrowed from [Com05].

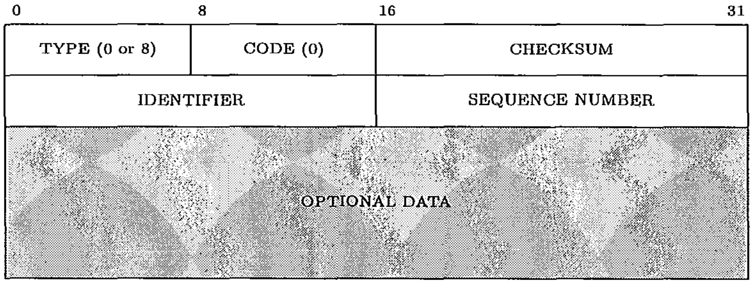


Figure 3.3: ICMP echo request or reply message format.

The field of interest is the *Optional Data* field which is listed as optional because it is a variable length field that contains data to be returned to the sender. Since this field is variable in length, [ZLSN05] suggests that it can be used to conceal information other than the data that the ICMP would normally return to the sender.

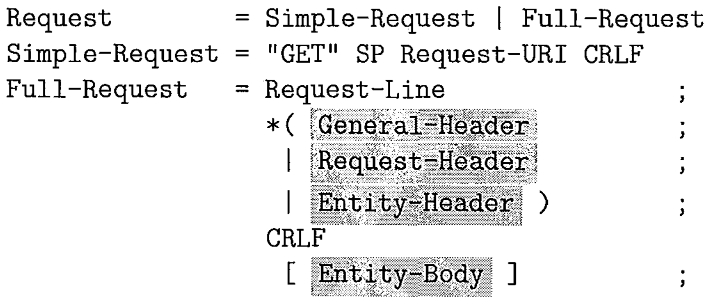
The availability of ICMP makes it an excellent carrier for covert information. It is reasonably simple to use and provides plenty of opportunities for transmitting covert messages in large capacity.

3.2.4 HTTP: Hyper-Text Transport Protocol

The *Hyper-Text Transport Protocol* (HTTP) is an application-layer protocol used to transfer information across the Internet. HTTP offers itself to be a formidable choice for a covert message carrier in that for many organizations, HTTP is the only protocol that is allowed from the internal network to the Internet since it is required for web-browsing [VH06].

HTTP Request

The specification of a HTTP request message is given in Figure 3.4. The grey fields are the fields of interest for hiding data for use in a covert channel. The figure is borrowed from [SK06].



```
Request        = Simple-Request | Full-Request
Simple-Request = "GET" SP Request-URI CRLF
Full-Request   = Request-Line                 ;
                 *( General-Header            ;
                  | Request-Header            ;
                  | Entity-Header )           ;
                 CRLF
                  [ Entity-Body ]             ;
```

Figure 3.4: Specification of an HTTP request.

HTTP request messages may contain multiple headers. In the *General-Header*, *Request-Header*, and *Entity-Header* fields, it is shown in [SK06], that along with common header line in a HTTP request, covert information may be sent via HTTP request messages by including arbitrary headers. Also, in an HTTP request message, the *Entity-Body* field is only present during POST requests which are requests used to send data to a server to be processed in some way. However, in [SK06], Smeets and Koot suggest that since it is not explicitly stated that the *Entity-Body* field should be excluded from other types of requests, it can be used to convey covert information.

HTTP Response

The specification of a HTTP response message is given in Figure 3.5 which is borrowed from [SK06]. The grey fields are the fields of interest for hiding data for use in a covert channel.

```
Response        = Simple-Response | Full-Response
Simple-Response = [ Entity- Body ]
Full-Response   = Status-Line                     ;
                  *( General-Header               ;
                   | Response-Header              ;
                   | Entity-Header )              ;
                  CRLF
                   [ Entity-Body ]                ;
```

Figure 3.5: Specification of an HTTP response.

Similar to HTTP request messages, HTTP response messages may contain multiple headers. As such, in the same way that HTTP request headers and the *Entity-Body* field can be used to transmit covert information, HTTP response headers and the *Entity-Body* field can be used. In [SK06], Smeets and Koot, state that combination of a HTTP request message covert channel with a HTTP response message covert channel allows for the creation of synchronous communication on the covert channel.

Because of the widespread use and availability of the Hyper-Text Transfer Protocol and the ability to easily conceal information in the request and response messages, HTTP makes an excellent choice for a carrier protocol for covert communication.

3.2.5 DNS: Domain Name System

The *Domain Name System* (DNS) is a transport-layer protocol that is used for storing and querying information of domain names in the distributed DNS database. Rather than using 32-bit integer IP addresses to identify machines, DNS allows for the identification of machines by pronounceable, easily remembered names.

DNS Message Header

The format of a DNS message is given in Figure 3.6 where the grey fields are the fields of interest for hiding data for use in a covert channel. The figure is borrowed from [Com05].

In a DNS message header, the *Identification* field is a 16-bit field allowing for the tracking of different queries that are made. According to [SK06], the *Identification* field can be used to transmit covert information if a scheme is created whereby a smaller space can be used to identify individual queries such that the remaining bits of the field can transmit information covertly. The *Number of Questions*, *Number of Answers*, *Number of Authority*, and *Number of Additional* fields can also be used to

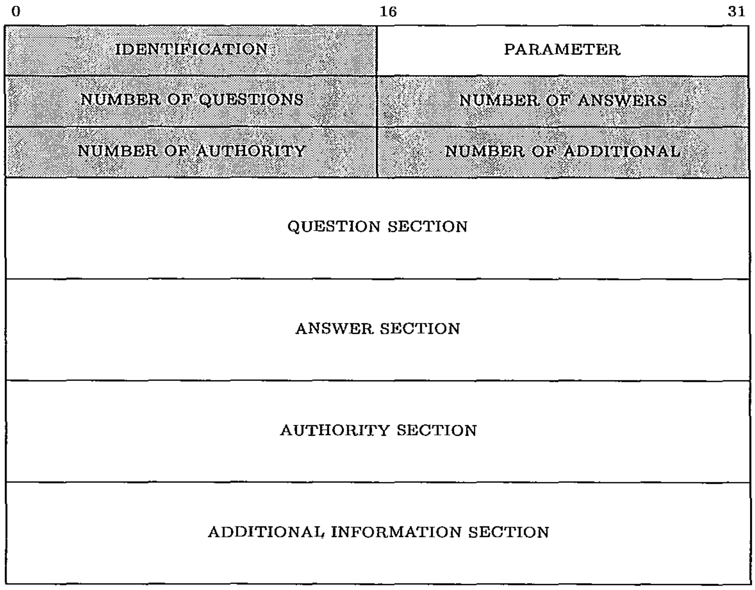


Figure 3.6: DNS message format.

transmit covert information if an agent can gain control over a DNS server. Each field can transmit 16 bits of information. It is important to note that if an agent spoofs any of the *Number of Questions*, *Number of Answers*, *Number of Authority*, or *Number of Additional* fields, then the corresponding sections must match the given number so as not to raise any suspicion of tampering.

DNS Query Header

When a DNS query is made, the DNS Message header is followed by a DNS Query Header which is shown in Figure 3.7, where the grey field is the field of interest for hiding data for use in a covert channel. The figure is taken from [Com05].

In [SK06], Smeets and Koot show how the *Query Domain Name* field, which represents the string of text entered as the actual query, can be used to transmit secret

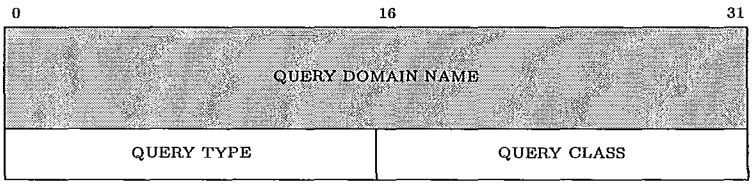


Figure 3.7: Format of entries in the *Question Section* of a DNS message.

information. The *Query Domain Name* field should be in the form of the *Full Qualified Domain Name (FQDN)*, which specifies the exact name and location of a machine in the tree hierarchy of the DNS protocol. The *Query Domain Name* field is limited to the maximum length of a FQDN which means that an agent can send up to 255 bytes of information as long as it complies to the limit of 63 octets per label in the FQDN. This means that a large amount of data can be transmitted per DNS message.

DNS Answer Header

Every answer to a DNS query consists of the DNS message header and the DNS query header as well as the DNS answer header given in Figure 3.8, where the grey field is the field of interest for hiding data for use in a covert channel. The figure is borrowed from [Com05]. The DNS answer header is the same for the *Answer Section*, *Authority Section*, and *Additional Information Section.*

According to [SK06], the *Resource Domain Name* field can transmit covert information in the same way as the *Query Domain Name* field. The *Resource Domain Name* field represents a FQDN to which the resource record pertains. Again, the *Resource Domain Name* field must comply to the rules for a FQDN.

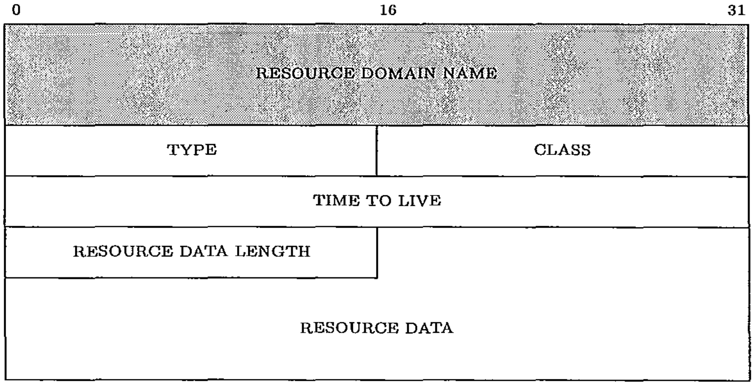


Figure 3.8: Format of entries in the *Answer Section*, *Authority Section*, and *Additional Information Section* of a DNS message.

Due to its complexity, the DNS protocol provides multiple carriers for covert channels making it a suitable choice for transmitting covert messages across the Internet.

3.2.6 FTP: File Transfer Protocol

The *File Transfer Protocol* (FTP) is a commonly used network protocol that allows for the exchange and manipulation of files over a TCP/IP based network. The client software for FTP integrates a function guaranteeing that at least one command will be sent in a fixed period of time. This function is referred to as the idle prevention scheme [ZLSN05]. Covert channels based on FTP exploit the use of this idle prevention scheme. To demonstrate the principal of this type of covert channel, suppose that we use three commands such as NOOP, ABOR, and ALLO in the idle prevention scheme. Also, assume that each of these commands can be sent at will by the user.

Command Mapping

In FTP, clients and servers exchange commands and responses to control the file transfer action. According to [ZLSN05], each command can be encoded into a bit-string wherein we can select a command space Q and encode each command by a fixed length number of bits. Therefore we can send NOOP to represent 01001 and ABOR to represent 01010, etc. If each of the 32 FTP commands are encoded in such a way, we can send 5 bits per command. This channel provides robustness in that in order for it to be eliminated, the normal FTP communication must be cut off since there is no covert information embedded into the packets, which can be destroyed just by rewriting the corresponding bits.

Sequence Length

According to [ZLSN05], a covert channel can be established by using a special sequence of the NOOP, ABOR, and ALLO commands to encode covert information. The simplest of these channels requires the use of only two of the above mentioned commands. This channel utilizes the number of sent NOOP commands to encode one byte of information. In order to transmit one byte of information, the user enters between zero and 255 NOOP commands followed by an ABOR command to represent the beginning of the next byte of information. The authors of [ZLSN05] suggest that this covert channel has good concealment because it uses a normal scheme in that it seems to simply send NOOP and ABOR commands at random to prevent the FTP control connections from entering an idle state.

Hierarchical Sequence Length

In [ZLSN05], Zou et al. describe a covert channel that utilizes the NOOP, ABOR, and ALLO commands sequence to perform a hierarchical encoding. We can divide N bits of covert information into two parts: the high order M bits and the low order $N - M$ bits. The number of high order bits are encoded with the number of ALLO commands sent in the commands sequence and the low order bits are encoded with the number of NOOP commands sent in the command sequence. The ABOR commands in the commands sequence mark the beginning of every N-bits of information. According to [ZLSN05], on average 16 commands are required to transmit 8 bits of covert information.

3.3 Document Formats

Lee and Tsai in [LT10] state that hiding data in various media with steganographic effects is a good way towards covert communication. The authors write that recently more and more investigations on the uses of various types of document formats for data hiding include PDF and Microsoft Word are being performed.

3.3.1 PDF: Portable Document Format

The *Portable Document Format* (PDF) is very popular nowadays. Therefore, using PDF as a carrier for covert information is convenient. In [LT10], Lee and Tsai describe a technique for using PDF as a carrier for covert information by viewing a message as a string of bits or characters encoded with a special ASCII code by binary or unitary coding embedded at the *between-word* or *between-character* location in the text of a PDF file. The study shows that the ASCII code A0, which is used as a non-breaking space, when embedded in a string of characters, becomes invisible in the windows of several popular PDF readers including Adobe Reader. It is suggested that there exists two ASCII codes which appear to be exactly the same white space, those codes being A0 and 20. Therefore, A0 and 20 can be used interchangeably as a *between-word* space to encode a message bit b according to the following encoding technique:

- if $b = 1$, then replace 20 between two words by A0
- if $b = 0$, then make no change

Another type of invisibility in PDF files can be created by setting the width of the ASCII code A0 to be zero. If this A0 is then embedded between two characters, it appears to be nothing. Suppose that we enumerate each of the characters of the message to have an integer index. Then a message character C can be hidden in a PDF file by a unitary encoding at a between-character location L by embedding m consecutive A0's at L, when the index of C is m.

The two above techniques are just a few of likely many ways to employ PDF as a carrier for covert information. Again, its popularity allows it to be a simple and convenient carrier for covert information.

3.3.2 doc: Microsoft Word Document

Microsoft Word is a popular word processor and is used worldwide. In [LT07], Liu and Tsai propose a method for covert communication using the Track Changes feature of

Microsoft Word. The basic idea is to degenerate the contents of a cover document D to arrive at another document D' by embedding a message in D. The degeneration introduces errors into the degenerated document D' so as to appear to be a preliminary work in a larger collaboration. The covert message is then transmitted in a document S which is produced from D' by revising D' back to D with the changes being tracked, making it appear as if an author corrected the errors in D'. This type of covert channel can be applied to more than Microsoft Word documents. The widespread use of Microsoft Word and the value of the Track Changes feature makes this type of covert channel an effective way of transmitting covert information.

3.4 Discussion

The covert channels presented in the previous sections show that in the cases when a network protocol is used as the covert message carrier, we are interested in a header field of the packet structure. The header format that is used can simply be abstracted as a data structure of some dimension and the field which is used can be represented as an element of that data structure. The same goes for covert channels which use document formats as covert message carriers since a document can be viewed as a two-dimensional data structure. Although many of the data structures for document formats and the header formats of network protocols are two-dimensional, it is not required that the data structure of the covert message carrier be restricted to two dimensions. We can have data structures of n dimensions. Usually through marshalling data before sending it on a channel and its unmarshalling at the channel destination, we are concretely sending a stream of bytes. However, abstractly and without taking the marshalling into account, we can see that we are sending complex data structures of dimension n. The model for covert channel communication includes the dimensions of time and channel. We must add the dimension of time in order to model the fact that the information that is sent in the communication is sent as a stream of discrete data packets over time and we must add the dimension of the communication medium (i.e., channel) in order to model the means by which the information can flow from the sender to the receiver. This means for any data structure of n dimensions, we will have an $n+2$-dimensional model. This means that the more complex the data that is being transmitted in the communication, the more complex the model becomes. This is a testament to the overall complexity of the issue of covert channels in computer systems and the difficulty in finding suitable detection and prevention mechanisms. It may be the case that the model can be simplified when a single communication channel is used or when only a single data structure is transmitted.

The model that I propose for protocol-based covert channel communication is given in Figure 3.9, in which we capture our perception of protocol-based covert channel communication. Our model represents multiple communication channels, each transmitting a stream of data which is in turn is a series of data structures sent over a period of time. This yields a four-dimensional model for covert channel communication. It is important to note that in the context of covert communication channels, it is assumed that the confidential information which is to be leaked is embedded into the data structure is some form.

Our model gives us the ability to account for the complexity of covert channels which

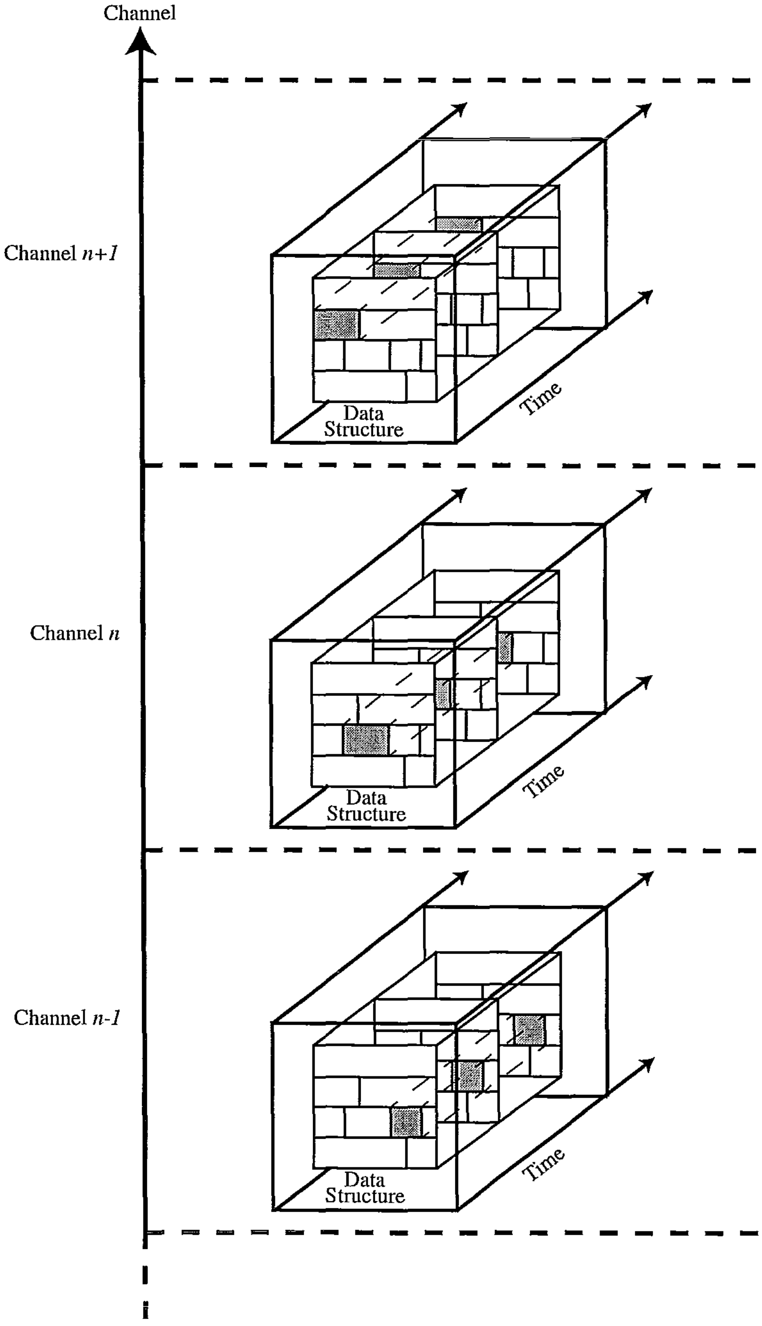


Figure 3.9: A model for protocol-based covert channel communication.

can lead to a better understanding of covert channel communication and its implications as a threat to computer and information security. Our model allows for the perception of protocol-based covert channel communication. A clear perception of the problem often leads to a better understanding of the problem. A better understanding of covert channel communication, enables us to use our model to represent a variety of covert channel usages in computer and information systems. For example, our model is able to represent simple cases of covert channel use, such as the use of a single communication channel, as well as more complex cases of covert channel use, such as one where confidential information may be leaked in parts over many communication channels. Our model captures the ability to combine two covert channels, say Channel n and Channel $n+1$, in order to verify whether any confidential information has been leaked in some form in the combination of the two channels.

Furthermore, we can use our understanding of covert channel communication to explore the relationship between covert channels and steganography. In steganography, information is embedded into some form of cover, i.e., images, audio, video, etc. It is clear that we can view the form of cover as a data structure of some dimension. For example, an image containing hidden information is simply a two-dimensional data structure, more specifically a pixel matrix. The image needs to be transmitted to its destination in some way, meaning that it would need to be sent at some point in time or at a series of points in time and through one or more communication channels.

If we take an acrostic, which is a widely used method of linguistic steganography, we can see it as a data structure that is a list of strings. For example, in the following acrostic, wc hide the word SKY.

So nice and blue,
Keeps the stars from falling bellow,
You should seek its guidance on where to go.

The acrostic is a list of verses that each starts with a letter from the hidden message SKY. One can think of putting the letters of the hidden word in the second position or any other agreed upon position. Our data is then of two dimensions. One can think that the message can be hidden in several acrostics written at different times, which adds a time dimension. Furthermore, several poets can collude together to send a message, which adds the channel dimension to the problem.

Many current steganographic systems put the elements of the hidden message at random positions (already agreed upon by the sender and the receiver). One can think about doing the same on the time dimension and on the channel dimension as well, which further obscures the message's trail.

As presented in Section 3.1, some argue that steganography is different from covert channel communication because covert channel communication uses unknown channels. According to the proposed model, a message can be sent on several channels. In this case, it is hard for an information leakage monitor to be aware of all the channels used. Therefore, we can have a covert channel while the monitor is not aware of all the channels used. Hence, requiring the awareness of an information leakage monitor of the channel(s) used to distinguish between steganography and covert channels is untenable.

The ability to represent steganography as a form of covert channel communication yields an inherent relationship between the two concepts. Since we are able to model both steganography and protocol-based covert channel communication using the same model, we can argue that steganography is a special case of covert channel communication.

3.5 Conclusion

The ability to create covert channels lies almost everywhere. Many of the most common network protocols and document formats allow for a wide variety of covert channels to be established easily and effectively. In this chapter, we have covered many common and simple ways to establish covert communication channels based on common protocols, however, it is important to note that it is very likely that there are many other ways of concealing information within these and other protocols. We have also explored the relationship between covert channel communication and steganography. We have seen that steganography can be viewed simply as a specific case of covert channel communication.

Chapter 4

Formulation of a Detection Technique

In this chapter, we formulate a new technique for detecting the leak of confidential information via covert channels in computer and information systems. In Section 4.1, we list our assumptions. In Section 4.2, we give a clear mathematical representation of the problem of covert channels. In Section 4.3, we present our technique.

4.1 Assumptions

In formulating the problem of covert channel communication in computer and information systems, we make the following assumptions:

(i) The communicating agents have a predefined scheme regarding how the information is transmitted from its source to its destination. This includes an agreement on the protocol to be exploited and the fields of the data structure to be used. This is a common assumption of covert channel communication.

(ii) The communication is recorded by monitors which begin recording when a communication channel is established.

(iii) The monitors maintain an unbounded history of all of the communication which has taken place. This allows for simplicity in reasoning about the abilities of the monitors the system.

(iv) The monitor always knows the set of confidential information protected by the security policy.

(v) The analysis is done in a forensics context, meaning that it is performed after the information has already been sent.

4.2 Representing Covert Channels As Relations

Finding an appropriate abstract representation for the information being sent on a channel is a crucial step in solving the problem of detecting the use of covert channels to leak confidential information. Without a good representation for information, we are unable to accurately model the scenarios in which confidential information is leaked via a covert channel. In Section 3.4, we discussed how we can view information sent over covert channels as being encapsulated in a data structure of some dimension. This data structure has fields in which the information is embedded. In this thesis, we represent the information sent on a channel as a relation, i.e., a series of data structures which are sent over time. At each time, we have an element of information sent. Therefore, one can see a stream of information as a subset of the Cartesian product of time and the state space of a data structure. If we model time by $\mathbb{N}$, and the set of information (or data) by $\mathbb{D}$, then a stream S is a subset of $\mathbb{N} \times \mathbb{D}$.

Therefore, it is a relation and more precisely, it is a function when we consider only one channel (without noise). We associate each datum with the time stamp at which it was received. For example, if the data sent on the channel was the sequence of characters 'h', 'e', 'l', 'l', 'o' to form the word "hello", the information that is sent on the stream is formed as the relation $R = \{(1, \text{`h'}), (2, \text{`e'}), (3, \text{`l'}), (4, \text{`l'}), (5, \text{`o'})\}$, where the character 'h' was sent at time 1, the character 'e' was sent at time 2 and so on.

The proposed representation offers simplicity when carrying out the computations required in the detection of a leak of confidential information over covert channels. Since a stream is a discrete sequence of data, indexed by time, a stream provides major advantages in that it allows us to take intervals of data from the channel and examine each interval, leading to computations of finite relations rather than infinite ones. As well as gaining simplicity from the use of a stream representation, we also gain simplicity from the use of relations. Relations are simple mathematical concepts. They also offer a certain level of abstraction in the model of covert channels, which, as discussed later in this chapter, gives much more power and flexibility in the ability to model particular types of covert channels. We do not rely on heuristics, even though they may be quite useful in the area of covert channel detection. Instead, we rely on mathematics since they provide us with a rigorous and formal way to uncover the use of covert communication channels to leak confidential information.

In order to uncover a covert channel, we show that it is sufficient to find an abstraction relation between the confidential information which we do not want to be leaked and the stream of information observed to be sent on the channel. An abstraction relation X can be seen as a simulation relation between two relations P and Q. In Figure 4.1, the relation X is an abstraction relation that relates $\mathbb{D}_P$ to $\mathbb{D}_Q$.

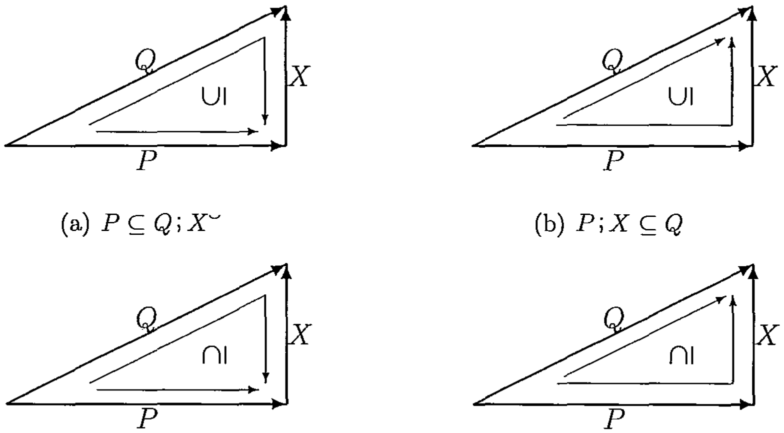
$$\begin{tikzcd}
 & & \mathbb{D}_Q \\
\mathbb{N} \arrow[urr, "Q"] \arrow[rr, "P"'] & & \mathbb{D}_P \arrow[u, "X"']
\end{tikzcd}$$

Figure 4.1: Diagram representing the relationship between the relation P and Q via the abstraction relation X.

In Figure 4.1, we have the relation P representing the confidential information which should not be sent on the channel, the relation Q representing the information that is observed by a monitor watching the information transmitted over the communication channel, and the relation X representing an abstraction relation that gives the relationship between P and Q. An abstraction relation X, requires that the diagram given in Figure 4.1 commutes. We can see that the diagram in Figure 4.1 can commute in four ways as described in Figure 4.2.

However, we can see that the diagrams in Figure 4.2 can be reduced to two diagrams. For Figure 4.2(a) and Figure 4.2(c) we have $P \subseteq Q ; X^{\smallsmile} \ \wedge \ Q ; X^{\smallsmile} \subseteq P$ which is equivalent to

$$Q ; X^{\smallsmile} = P \qquad \text{which is equivalent to} \qquad X ; Q^{\smallsmile} = P^{\smallsmile} \tag{4.1}$$



(a) $P \subseteq Q;X^\smile$

(b) $P;X \subseteq Q$

(c) $Q;X^\smile \subseteq P$

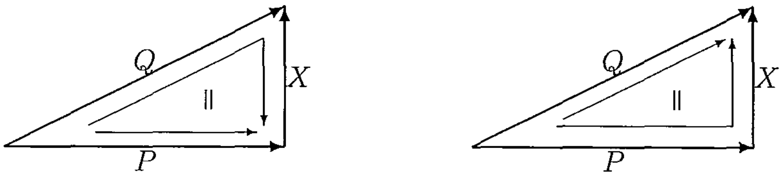
(d) $Q \subseteq P;X$

Figure 4.2: Four ways in which Figure 4.1 can commute.

and for Figure 4.2(b) and Figure 4.2(d) we have $P;X \subseteq Q \;\wedge\; Q \subseteq P;X$ which is equivalent to

$$P;X = Q \qquad \text{which is equivalent to} \qquad X^\smile;P^\smile = Q^\smile \tag{4.2}$$

So, we have the following two diagrams, given in Figure 4.3, for which we are able to solve their corresponding equations for an abstraction relation X.



(a) $Q;X^\smile = P$

(b) $P;X = Q$

Figure 4.3: Reduction of the ways in which Figure 4.1 can commute.

In each case, the confidential information represented by P is known. The observed information represented by Q is known only after observing the information that is sent on the communication channel(s). We are looking to find a solution to Equations 4.1 or 4.2. The solution is an abstraction relation X relating the relation P and the relation Q. In terms of covert channels, the solution is an abstraction relation relating the confidential information and the observed information that has been sent on the communication channel(s).

The mathematics required to solve for the abstraction relation, X, in the diagrams given in Figure 4.3 was first introduced by Bertrand Russell who wrote on the similarity of relations in 1919 [Rus93]. Russell wrote that two relations are similar when there is at least one abstraction relation between them. When two relations are similar, they share all properties that do not depend upon the actual terms of their fields. This means that if we can find an abstraction relation between the confidential information and the observed information sent on a covert channel, then we can conclude that there is a similarity between the confidential information and the observed information. To simplify, this means that each element of the confidential information can be mapped to one or more elements of the observed information. This is indeed what we see in Figure 4.1, whereby composing the relation P which the abstraction relation X, we get the relation Q. Putting this in terms of the communication of information over covert channels, we have that the transformation of the confidential information by the abstraction relation, gives the information observed on the communication channel. Again we emphasize that it is not the communication between agents that is in violation of a security policy, but rather, the information that is being communicated. We are only interested in finding an abstraction relation between the confidential information that is known to the monitor and the observed information sent on the channel(s).

When considering the detection of the use of covert channels in a system, we must determine the necessary and sufficient conditions which constitute the existence of a covert channel in violation of a security policy. We consider a covert channel to be detected if and only if there exists an abstraction relation between the confidential information and the information that is sent on the channel such that the abstraction relation is different from $\emptyset$ and $\mathbb{L}$.

4.3 The Proposed Technique

The proposed technique for the detection of the leak of confidential information via covert channels has two components: monitoring the information sent on the communication channels and finding an abstraction relation relating the confidential information to the information observed to be sent on the communication channel(s).

4.3.1 Monitoring the Communication Channels

As an illustrative example, we consider a system for which two agents are communicating as given in Figure 4.4.

Suppose that agent A is communicating from within an organization and suppose that agent B is communicating from outside the organization. Suppose that the

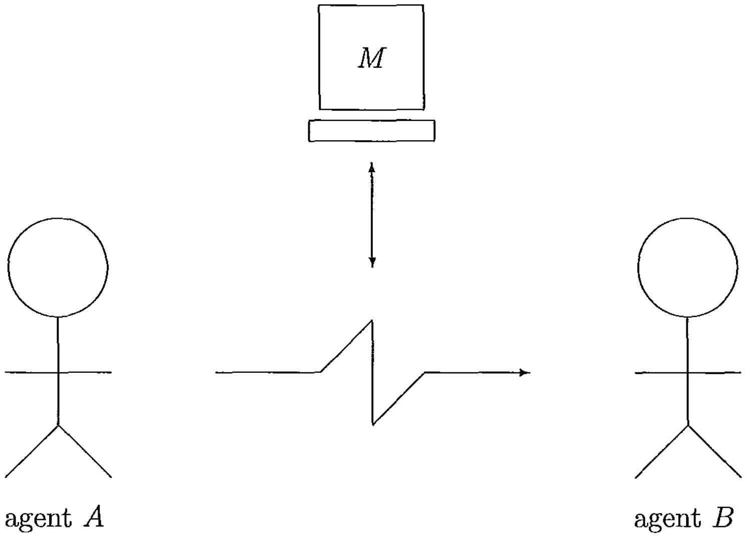


Figure 4.4: A scenario consisting of two agents communicating while being monitored.

organization has a security policy which defines a confidential information of the organization to be the sequence of the first six digits of the number π, i.e., we have $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$ which is a representation of the sequence $3,1,4,1,5,9$. Assume that the organization from which agent A is communicating wishes to detect if this confidential information is being leaked to an agent outside of the organization. The organization installs a monitor M on all of the communication channels from which an agent from within the organization can communicate with an agent outside the organization. The monitors that are installed keep a history of the communication that has been observed on each channel, denoted by Q. The monitors of the communication channels that are installed by the organization can perform either a post-mortem analysis[1] or a real-time analysis looking for an abstraction relation between P and Q.

[1] Post-mortem analysis refers to the fact that the analysis is being done in a digital forensics context whereby confidential information may have already been leaked and the damage may already be done.

We assume that agent A and agent B agree on a scheme for transmitting the confidential information. It is decided that agent A exploits the Internet Protocol, in particular, the IP *Identification* field in order to leak the confidential information to agent B. Suppose that using the 16-bit IP *Identification* field, agent A agrees to send, in a sequence of six IP packets, the set of confidential information of its organization. Also, suppose that in order to attempt to mask the data being sent, agent A first encrypts the information before embedding it into the IP header. For this purpose, agent A uses a public key encryption technique (though it is not important how the information is encrypted) to encrypt the information. The encryption generates the sequence[2] $\{(1,12),(2,1),(3,16),(4,1),(5,17),(6,18)\}$ in place of the sequence $\{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$. Now that agent A has the information which it intends to leak to agent B, it is simply a matter of embedding the information into the IP *Identification* field as agreed upon. This means that agent A sends six IP packets to agent B with the respective IP *Identification* fields corresponding to

- Packet 1: 0000 0000 0000 1100
- Packet 2: 0000 0000 0000 0001
- Packet 3: 0000 0000 0001 0000
- Packet 4: 0000 0000 0000 0001
- Packet 5: 0000 0000 0001 0001
- Packet 6: 0000 0000 0001 0010

[2] This set is generated using RSA encryption with $p = 3, q = 7, N = 21, e = 5, d = 41$.

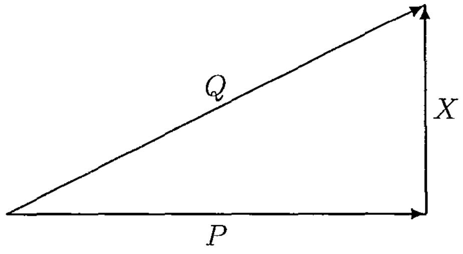
Recall that since the IP *Identification* field is 16 bits in length, the binary numbers which are sent need to be padded to be 16 bits in length as well.

As mentioned, we assume that the organization for which agent A is employed, has monitors installed to listen in on the communication channels from within the organization to destinations outside of the organization. The monitors begin monitoring the communication of over a channel when the communication channel is established. This is to say that when the IP connection is created between agent A and agent B, the monitor M is launched and watches and records the packets that are being transmitted. We can view the monitor as a specialized and customizable packet sniffer in the case of covert channels exploiting the use of network protocols. In order for the monitor to be effective, we must assume that it is configured with the following necessary information:

- *Protocol*: This refers to the protocol in which the communicating agents are using in order to communicate. In our example, the protocol that is being used is the Internet Protocol (IP).
- *Header Field*: This refers to the field in the header of the particular protocol which is being used as the carrier for the covert information channel. In our example, the field that is being exploited in the IP *Identification* field.
- *Confidential Information*: The monitor must know what the set of confidential information is in order to perform the required analysis. In our example, the set of confidential information that the monitor must know is the set $\{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$.

- *Analysis Tools*: This refers to the set of tests which can be run in order to verify whether there is an abstraction relation between the confidential information and the information that is observed by the monitor. This will be discussed further in Section 4.3.2.

The monitor watches the stream of packets being transmitted from agent A to agent B, constructing and recording the relational stream of information being sent. This is to say that if as we previously assumed, agent A sends the confidential sequence of information $3, 1, 4, 1, 5, 9$ as the encrypted sequence of information $12, 1, 16, 1, 17, 18$, then the relation constructed by the monitor would be given as $Q = \{(1, 12), (2, 1), (3, 16), (4, 1), (5, 17), (6, 18)\}$. Recall that the monitor is already equipped with the relation corresponding to the set of confidential information, i.e., $P = \{(1, 3), (2, 1), (3, 4), (4, 1), (5, 5), (6, 9)\}$. The monitor now knows the stream of confidential information which should not be sent according to the security policy and the stream of information observed to have been sent on the communication channel. The monitor needs to determine whether the confidential information has been leaked in any capacity. The monitor decides this by attempting to find an abstraction relation X which relates the relation constructed as P, the confidential information, and the relation constructed as Q, the observed information. The diagram for the given example is shown in Figure 4.5. In Section 4.3.2, we provide a number of tests which the monitor can carry out in order to analyze its known information and collected information to determine whether an abstraction relation exists and in some cases allows for the abstraction relation to be computed.



$$\begin{aligned} P &= \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\} \\ Q &= \{(1,12),(2,1),(3,16),(4,1),(5,17),(6,18)\} \end{aligned}$$

Figure 4.5: Diagram reflecting the illustrative example outlined in Section 4.3.1.

4.3.2 Finding an Abstraction Relation

Since the monitor knows the relation corresponding to the set of confidential information, $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$, and the relation corresponding to the set of observed information, $Q = \{(1,12),(2,1),(3,16),(4,1),(5,17),(6,18)\}$, the existence of an abstraction relation X which relates the confidential information, P, and the observed information, Q can be verified using Proposition 4.3.1.

Proposition 4.3.1. *$X \,;P = Q$ has a solution if and only if $Q = (Q/P)\,;P$*

Proof.

$(\Longleftarrow)$ $Q = (Q/P)\,;P \implies X\,;P = Q$ has a solution

$X\,;P = Q$ has a solution

$\iff$ $\langle$ Formalization $\rangle$

$\exists(X \mid: X ; P = Q)$

$\Longleftrightarrow$ $\langle$ Antisymmetry $\rangle$

$\exists(X \mid: X ; P \subseteq Q \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.3(i) $\rangle$

$\exists(X \mid: X \subseteq Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Definition of $\subseteq$ $\rangle$

$\exists(X \mid: (X = Q/P \vee X \subset Q/P) \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Distributivity of $\exists$ over $\vee$ $\rangle$

$\exists(X \mid: X = Q/P \wedge Q \subseteq X ; P) \vee \exists(X \mid: X \subset Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Trading $\rangle$

$\exists(X \mid X = Q/P : Q \subseteq X ; P) \vee \exists(X \mid: X \subset Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ One-Point Axiom $\rangle$

$Q \subseteq X ; P[X := Q/P] \vee \exists(X \mid: X \subset Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Substitution $\rangle$

$Q \subseteq (Q/P) ; P \vee \exists(X \mid: X \subset Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Hypothesis: $Q = (Q/P) ; P$ $\rangle$

$\mathsf{true} \vee \exists(X \mid: X \subset Q/P \wedge Q \subseteq X ; P)$

$\Longleftrightarrow$ $\langle$ Zero of $\vee$ $\rangle$

true

$(\Longrightarrow)$ $X\,;P = Q$ has a solution $\Longrightarrow$ $Q = (Q/P)\,;P$

$X\,;P = Q$ has a solution

$\Longleftrightarrow$ $\langle$ Formalization $\rangle$

$\exists(X \mid: X\,;P = Q)$

$\Longleftrightarrow$ $\langle$ Antisymmetry $\rangle$

$\exists(X \mid: X\,;P \subseteq Q \ \wedge\ Q \subseteq X\,;P)$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.3(i) $\rangle$

$\exists(X \mid: X \subseteq Q/P \ \wedge\ Q \subseteq X\,;P)$

$\Longrightarrow$ $\langle$ Isotony of ; $\rangle$

$\exists(X \mid: X\,;P \subseteq (Q/P)\,;P \ \wedge\ Q \subseteq X\,;P)$

$\Longrightarrow$ $\langle$ Transitivity of $\subseteq$ & Idempotency of $\wedge$ $\rangle$

$\exists(X \mid: Q \subseteq X\,;P \subseteq (Q/P)\,;P \ \wedge\ Q \subseteq (Q/P)\,;P)$

$\Longleftrightarrow$ $\langle$ Distributivity of $\wedge$ over $\exists$ $\rangle$

$Q \subseteq (Q/P)\,;P \ \wedge\ \exists(X \mid: Q \subseteq X\,;P \subseteq (Q/P)\,;P)$

$\Longrightarrow$ $\langle$ Weakening $\rangle$

$Q \subseteq (Q/P)\,;P$

$\Longrightarrow$ $\langle$ Proposition 2.2.4(vii) $\rangle$

$Q \subseteq (Q/P)\,;P \subseteq Q$

$\Longleftrightarrow$ $\langle$ Antisymmetry $\rangle$

$Q = (Q/P) ; P$

□

Proposition 4.3.1 is used as a test to verify if there is an abstraction between the observed information and the confidential information. This test is directly related to Figure 4.5 in that if the test holds, we can say that the diagram in Figure 4.5 commutes and we can find an abstraction relation X that satisfies Equation 4.1 or Equation 4.2 which is not the empty relation or the universal relation,. Therefore, we can say that the confidential information P seems to have been leaked using the abstraction given by X which was sent as the observed information Q.

Corollary 4.3.1. *Let P be the relation containing confidential information. Let Q be a relation representing an information observed on the monitored communication channel. The confidential information contained in P is being leaked as that represented by Q if and only if $P = Q ; (Q \backslash P) \ \vee \ Q = P ; (P \backslash Q)$.*

Proof.

According to the problem formulation illustrated by Figure 4.3, we need to find solutions to either Equation 4.1 or Equation 4.2.

Therefore,

$X ; Q^{\smile} = P^{\smile}$ or $X^{\smile} ; P^{\smile} = Q^{\smile}$ have solutions

$\Longleftrightarrow$ $\langle$ Proposition 4.3.1 $\rangle$

$P^{\smile} = (P^{\smile}/Q^{\smile})\,;Q^{\smile} \;\vee\; Q^{\smile} = (Q^{\smile}/P^{\smile})\,;P^{\smile}$

$\iff$ ⟨ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vii) &

Proposition 2.2.4(i) ⟩

$P = Q\,;(Q\backslash P) \;\vee\; Q = P\,;(P\backslash Q)$

□

The test outlined in Corollary 4.3.1 is automated in RELVIEW as Program A.2.1 (see Appendix A).

Now that we have verified whether an abstraction relation does in fact exist. The next step is to compute the abstraction relation giving us the abstraction that is used to relate the confidential information and information that is sent on the channel. Using Proposition 4.3.2, we are able to compute the abstraction relation X. The proposition also allows for filtering on the abstraction relation to look for an abstraction which maps the particular elements of the confidential information to particular elements of the observed information. The filtering relation is designed by the analyst and is represented by R.

Proposition 4.3.2. *Let P and Q be relations. $X\,;P = Q$ has a solution $X = R \cap (Q/P)$ if and only if $Q \subseteq \big(R \cap (Q/P)\big)\,;P$.*

Proof.

$(\Longleftarrow)$ $Q \subseteq \big(R \cap (Q/P)\big) ; P \implies X ; P = Q$ has $R \cap (Q/P)$ as a solution

$X ; P = Q$ has $R \cap (Q/P)$ as a solution

$\iff$ $\langle\, X = R \cap (Q/P) \quad \& \quad \text{Substitution} \,\rangle$

$\big(R \cap (Q/P)\big) ; P = Q$

$\iff$ $\langle$ Antisymmetry $\rangle$

$\big(R \cap (Q/P)\big) ; P \subseteq Q \;\wedge\; Q \subseteq \big(R \cap (Q/P)\big) ; P$

$\iff$ $\langle$ Hypothesis: $Q \subseteq \big(R \cap (Q/P)\big) ; P$ $\rangle$

$\big(R \cap (Q/P)\big) ; P \subseteq Q \;\wedge\; \mathsf{true}$

$\iff$ $\langle$ Definition 2.2.9(ii) & Identity of $\wedge$ $\rangle$

$(R \cap \overline{\overline{Q} ; P^{\smile}}) ; P \subseteq Q$

$\iff$ $\langle$ Identity of $\wedge$ & Proposition 2.2.2 & Proposition 2.2.1(vi) $\rangle$

$\overline{Q} ; P^{\smile} \subseteq (\overline{R} \cup \overline{Q} ; P^{\smile})$

$\Longleftarrow$ $\langle$ Weakening $\rangle$

$\overline{Q} ; P^{\smile} \subseteq \overline{Q} ; P^{\smile}$

$\Longleftarrow$ $\langle$ Reflexivity of $\subseteq$ $\rangle$

true

$(\Longrightarrow)$ $X \,;P = Q$ has $R \cap (Q/P)$ as a solution $\Longrightarrow Q \subseteq \big(R \cap (Q/P)\big) \,; P$

$$\begin{aligned}
& X\,;P = Q \ \wedge\ X = R \cap (Q/P) \\
\Longrightarrow \quad & \langle \text{ Substitution of } X \ \rangle \\
& \big(R \cap (Q/P)\big)\,;P = Q \\
\Longleftrightarrow \quad & \langle \text{ Antisymmetry } \rangle \\
& \big(R \cap (Q/P)\big)\,;P \subseteq Q \ \wedge\ Q \subseteq \big(R \cap (Q/P)\big)\,;P \\
\Longrightarrow \quad & \langle \text{ Weakening } \rangle \\
& Q \subseteq \big(R \cap (Q/P)\big)\,;P
\end{aligned}$$

□

The relation R plays the role of a filter. A filter allows for the removal of some unwanted elements of the transmitted sequences. The filter R allows us to select only those elements of the transmitted sequences which we are interested in examining further. In its most general case, if we consider the filter R to be the universal relation, $\mathbb{L}$, we are interested in all of the elements of the transmission and Proposition 4.3.2 gives us a way to compute the abstraction relation X. Otherwise, we can select the elements of the range of the confidential information for which we wish to find an abstraction by choosing different filtering relations for R. By computing an abstraction relation which is not the empty relation or the universal relation we can say that we have uncovered a leak of confidential information on the communication channel.

Corollary 4.3.2. *Let P be the relation containing confidential information. Let Q be a relation representing an information observed on the monitored communication channel. Let R be a filtering relation allowing for the selection of particular elements of the relations P and Q. The confidential information included in P is being leaked as that represented by Q via the abstraction relation*

(i) $X = R \cap (Q\backslash P)^{\smile}$ *if and only if* $P \subseteq Q ; \big(R^{\smile} \cap (Q\backslash P)\big)$

(ii) $X = R \cap (P\backslash Q)$ *if and only if* $Q \subseteq P ; \big(R \cap (P\backslash Q)\big)$

(iii) $X = R \cap \mathit{syq}(P, Q)$ *if and only if* $P \subseteq Q ; \big(R^{\smile} \cap (Q\backslash P)\big) \ \wedge \ Q \subseteq P ; \big(R \cap (P\backslash Q)\big)$

Proof.

(i) $P \subseteq Q ; \big(R^{\smile} \cap (Q\backslash P)\big)$

$\iff$ ⟨ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vi) & Proposition 2.2.1(vii) ⟩

$P^{\smile} \subseteq \big(R \cap (Q\backslash P)^{\smile}\big) ; Q^{\smile}$

$\iff$ ⟨ Proposition 4.3.2 ⟩

$\exists(X \mid X = R \cap (Q\backslash P)^{\smile} : X ; Q^{\smile} = P^{\smile})$

(ii) $Q \subseteq P ; \big(R \cap (P\backslash Q)\big)$

$\iff$ ⟨ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vi) & Proposition 2.2.1(vii) ⟩

$Q^{\smile} \subseteq \big(R^{\smile} \cap (P\backslash Q)^{\smile}\big) ; P^{\smile}$

$\Longleftrightarrow$ $\langle$ Proposition 4.3.2 $\rangle$

$\exists(X \mid X = R \cap (P\backslash Q) \;:\; X^{\smile} ; P^{\smile} = Q^{\smile})$

(iii) $P \subseteq Q ; \big(R^{\smile} \cap (Q\backslash P)\big) \land Q \subseteq P ; \big(R \cap (P\backslash Q)\big)$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vi) & Proposition 2.2.1(vii) & Proposition 2.2.4(ii) $\rangle$

$P^{\smile} \subseteq \big(R \cap (P^{\smile}/Q^{\smile})\big) ; Q^{\smile} \land Q^{\smile} \subseteq \big(R^{\smile} \cap (Q^{\smile}/P^{\smile})\big) ; P^{\smile}$

$\Longleftrightarrow$ $\langle$ Proposition 4.3.2 $\rangle$

$\exists(X \mid X = R \cap (P^{\smile}/Q^{\smile}) \land X^{\smile} = R^{\smile} \cap (Q^{\smile}/P^{\smile}) \;:\; X ; Q^{\smile} = P^{\smile} \land X^{\smile} ; P^{\smile} = Q^{\smile})$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) $\rangle$

$\exists(X \mid X = R \cap (P^{\smile}/Q^{\smile}) \land X = R \cap (P\backslash Q) \;:\; X ; Q^{\smile} = P^{\smile} \land X^{\smile} ; P^{\smile} = Q^{\smile})$

$\Longleftrightarrow$ $\langle$ Golden Rule Axiom: $X = P \land X = Q \Longleftrightarrow X = (P \cup Q) \land X = (P \cap Q)$ $\rangle$

$\exists(X \mid X = \big[\big(R \cap (P^{\smile}/Q^{\smile})\big) \cap \big(R \cap (P\backslash Q)\big)\big] \land X = \big[\big(R \cap (P^{\smile}/Q^{\smile})\big) \cup \big(R \cap (P\backslash Q)\big)\big] \;:\; X ; Q^{\smile} = P^{\smile} \land X^{\smile} ; P^{\smile} = Q^{\smile})$

$\Longleftrightarrow$ $\langle$ Distributivity of $\cap$ of $\cup$ $\rangle$

$\exists(X \mid X = R \cap \big((P^{\smile}/Q^{\smile}) \cap (P\backslash Q)\big) \land X = R \cap \big((P^{\smile}/Q^{\smile}) \cup (P\backslash Q)\big) \;:\; X ; Q^{\smile} = P^{\smile} \land X^{\smile} ; P^{\smile} = Q^{\smile})$

$\Longleftrightarrow$ $\langle$ Golden Rule Axiom: $X = R \cap (P \cap Q) \land X = R \cap (P \cup Q) \Longleftrightarrow$

$X = R \cap P \cap Q \land (P \cap Q) = (P \cup Q)$ $\rangle$

$$\exists(X \mid X = R \cap ((P^{\smallsmile}/Q^{\smallsmile}) \cap (P\backslash Q)) \wedge [((P\backslash Q) \cap (P^{\smallsmile}/Q^{\smallsmile})) = ((P\backslash Q) \cup (P^{\smallsmile}/Q^{\smallsmile}))] : X;Q^{\smallsmile} = P^{\smallsmile} \wedge X^{\smallsmile};P^{\smallsmile} = Q^{\smallsmile})$$

$\iff$ ⟨ Proposition 2.2.4(ii) ⟩

$$\exists(X \mid X = R \cap ((P^{\smallsmile}/Q^{\smallsmile}) \cap (P\backslash Q)) \wedge \mathsf{true} : X;Q^{\smallsmile} = P^{\smallsmile} \wedge X^{\smallsmile};P^{\smallsmile} = Q^{\smallsmile})$$

$\iff$ ⟨ Definition 2.2.10 & Identity of $\wedge$ ⟩

$$\exists(X \mid X = R \cap \mathsf{syq}(P,Q) : X;Q^{\smallsmile} = P^{\smallsmile} \wedge X^{\smallsmile};P^{\smallsmile} = Q^{\smallsmile})$$

□

The computations outlined in Corollary 4.3.2 are automated in RELVIEW as Program A.2.2 (see Appendix A).

It is possible when the confidential information and the information observed to be sent on the channel have certain properties, namely if they are bijections, that we can have a specialized case of Corollary 4.3.2 where the test is simplified based on the results of Proposition 2.2.5. This simplified test and computation is given in Corollary 4.3.3.

Corollary 4.3.3. *Let P be a bijection containing confidential information. Let Q be a bijection representing an information observed on the monitored communication channel. Let R be a filtering relation allowing for the selection of particular elements of the relations P and Q. The confidential information contained in P is being leaked as that represented by Q via the abstraction relation $X = R \cap (P\backslash Q)$ if and only if $P\backslash Q = (Q\backslash P)^{\smallsmile}$*

Proof.

According to the problem formulation illustrated by Figure 4.3, we need to find solutions to Equation 4.1 and Equation 4.2.

$$X = R \cap (P \backslash Q)$$

$\Longleftrightarrow$ $\langle$ Corollary 4.3.2 $\rangle$

$$P \subseteq Q ; \big(R^{\smile} \cap (Q \backslash P)\big) \ \wedge \ Q \subseteq P ; \big(R \cap (P \backslash Q)\big)$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vi) & Proposition 2.2.1(vii) & Proposition 2.2.4(ii) $\rangle$

$$P^{\smile} \subseteq \big(R \cap (P^{\smile}/Q^{\smile})\big) ; Q^{\smile} \ \wedge \ Q^{\smile} \subseteq \big(R^{\smile} \cap (Q^{\smile}/P^{\smile})\big) ; P^{\smile}$$

$\Longleftrightarrow$ $\langle$ Proposition 4.3.2 $\rangle$

$$\exists(X \mid X = R \cap (P^{\smile}/Q^{\smile}) \wedge X^{\smile} = R^{\smile} \cap (Q^{\smile}/P^{\smile}) \ : \ X ; Q^{\smile} = P^{\smile} \wedge X^{\smile} ; P^{\smile} = Q^{\smile})$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) $\rangle$

$$\exists(X \mid X = R \cap (P^{\smile}/Q^{\smile}) \wedge X = R \cap (P \backslash Q) \ : \ X ; Q^{\smile} = P^{\smile} \wedge X^{\smile} ; P^{\smile} = Q^{\smile})$$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.4(ii) $\rangle$

$$\exists(X \mid X = R \cap (Q \backslash P)^{\smile} \wedge X = R \cap (P \backslash Q) \ : \ X ; Q^{\smile} = P^{\smile} \wedge X^{\smile} ; P^{\smile} = Q^{\smile})$$

$\Longleftrightarrow$ $\langle$ P is a bijection & Q is a bijection & $P \subseteq Q ; (Q \backslash P)$ & $Q \subseteq P ; (P \backslash Q)$ & Proposition 2.2.5(v) $\rangle$

$$\exists(X \mid X = R \cap (P \backslash Q) \wedge X = R \cap (P \backslash Q) \ : \ X ; Q^{\smile} = P^{\smile} \wedge X^{\smile} ; P^{\smile} = Q^{\smile})$$

$\Longleftrightarrow$ $\langle$ Idempotency of $\wedge$ $\rangle$

$$\exists(X \mid X = R \cap (P \backslash Q) \ : \ X ; Q^{\smile} = P^{\smile} \wedge X^{\smile} ; P^{\smile} = Q^{\smile})$$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.3(i) $\rangle$

$$\exists(X \mid X = R \cap (P\backslash Q) : X = P^{\smile}/Q^{\smile} \wedge X^{\smile} = Q^{\smile}/P^{\smile})$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.4(i) $\rangle$

$$\exists(X \mid X = R \cap (P\backslash Q) : X = (Q\backslash P)^{\smile} \wedge X = P\backslash Q)$$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.4(i) $\rangle$

$$\exists(X \mid X = R \cap (P\backslash Q) : (Q\backslash P)^{\smile} = P\backslash Q \wedge X = P\backslash Q)$$

$\Longleftrightarrow$ $\langle$ One-Point Axiom & $R = \mathbb{L}$ & Identity of $\cap$ $\rangle$

$$\big((Q\backslash P)^{\smile} = P\backslash Q \wedge X = P\backslash Q\big)[X := P\backslash Q]$$

$\Longleftrightarrow$ $\langle$ Substitution $\rangle$

$$(Q\backslash P)^{\smile} = P\backslash Q \wedge P\backslash Q = P\backslash Q$$

$\Longleftrightarrow$ $\langle$ Reflexivity of $=$ & Identity of $\wedge$ $\rangle$

$$(Q\backslash P)^{\smile} = P\backslash Q$$

□

Examples illustrating Corollary 4.3.3 are mainly trivial since they involve P and Q being bijections. Although applications of Corollary 4.3.3 may be limited, it is still an important result in terms of the overall complexity of computing the abstraction relation X due to the simplified conditions.

It is important to indicate when the test outlined in Proposition 4.3.1 fails. The test fails when an element of the confidential information maps to more than one element of the information observed to be sent on the communication channel, of which another element of the confidential information is already mapped. It is again important to elaborate on what it means for the test to fail. In many cases, the failure of the test means that there is no abstraction between the confidential information and the information observed to be sent on the channel. However, it is possible that an abstraction exists between part of the confidential information and the information observed to be sent on the channel. This issue of finding an abstraction relation for part of the confidential information will be discussed further in Chapter 7. The failure of the test can be best illustrated through Example 4.3.1.

Example 4.3.1. *Consider a case where the set of confidential information is represented as* $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$. *Suppose that agent A sends the information* $Q = \{(1,12),(2,1),(3,16),(4,12),(5,17),(6,18)\}$ *on the communication channel. We verify the existence of an abstraction relation by applying Corollary 4.3.1.*

$$P = Q\,;(Q\backslash P) \ \vee \ Q = P\,;(P\backslash Q)$$

$\iff$ $\langle$ *Substitution:* $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$ *and* $Q = \{(1,12),(2,1),(3,16),(4,12),(5,17),(6,18)\}$ *&* *Computation of* $Q\,;(Q\backslash P)$ *and* $P\,;(P\backslash Q)$ $\rangle$

$$\{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\} = \{(2,1),(3,4),(5,5),(6,9)\} \quad \vee$$
$$\{(1,12),(2,1),(3,16),(4,12),(5,17),(6,18)\} = \{(1,12),(3,16),(5,17),(6,18)\}$$

$\iff$ $\langle$ *Equality* $\rangle$

$\mathsf{false} \vee \mathsf{false}$

$\iff$ $\langle$ *Idempotency of* $\vee$ $\rangle$

false

Therefore, the test is false and has failed. Thus, we can conclude that an abstraction relation cannot be computed and that the confidential information represented by P is not being leaked as that represented by Q. This is due to the inconsistency introduced at times 1, 2, and 4 where the data observed at each of these times generates an abstraction relation which cannot be used to accurately uncover the confidential information at those times.

The ideas presented in Proposition 4.3.1 and Proposition 4.3.2 are the core of the detecting whether confidential information has been leaked via covert channels. Equipped with each of the above propositions and corollaries, and under our assumptions, it is possible to detect the leak of confidential information via covert channels in a digital forensics context, i.e., investigation after the information has already been sent. The above propositions represent tests for which we can determine the existence of an abstraction relation defining the relationship between the confidential information and the information observed to be sent on the communication channel(s). Again, the existence of an abstraction relation is often enough to raise suspicion that confidential information has been leaked via a covert channel. Hence, we have formulated a mathematical framework for the post-mortem detection of the leak of confidential information via covert channels.

4.4 Conclusion

The problem of confidentiality being compromised by the use of covert channel communication is growing in importance. In this chapter, we provided a mathematical representation of the problem of covert channels and presented a technique to detect the leak of confidential information through covert channel communication. The formulation was based on a number of assumptions particularly that the analysis and detection occurs after the information has already been transmitted. This leads to an inherent relationship between our proposed technique and digital forensics investigations.

Chapter 5

Application of the Detection Technique

In this chapter, we look at how the covert channel detection technique formulated in Chapter 4 can be applied to different scenarios involving the leak of confidential information via covert channels. Through a series of examples, we will see the versatility of the detection technique and how it can be used to detect whether confidential information has been leaked via covert channels.

In this chapter, we automate the given examples using the RELVIEW tool. For more information regarding the use of the RELVIEW tool, refer to Appendix A.

5.1 Information Leaked on a Single Channel

We continue with the illustrative example introduced in Section 4.3. Recall that the confidential information is defined by the security policy to be the first six digits of the number π which is represented by the relation $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$ and that agent A and agent B have a scheme for transmitting the confidential information on a single communication channel as the sequence $Q = \{(1,12),(2,1),(3,16),(4,1),(5,17),(6,18)\}$.

This scenario illustrates the case when we are looking for a solution to an equation of the form $X\,;P = Q$ where

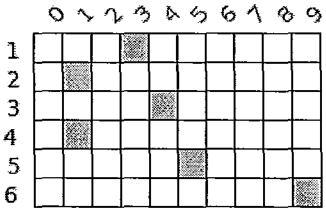
- X is the abstraction relation,
- P is the relation representing the confidential information, and
- Q is the relation representing the information observed to be sent on the communication channel.

Through Example 5.1.1 we will show that we can detect that the confidential information has been leaked via a single communication channel.

Example 5.1.1. *Consider a case where the set of confidential information is represented as* $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$. *Suppose that agent* A *sends this information encrypted over a single communication channel as* $Q = \{(1,12),(2,1),(3,16),(4,1),(5,17),(6,18)\}$.

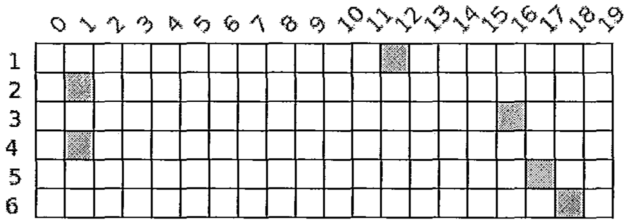
We define the relations P *and* Q *in* `RELVIEW` *as follows:*

We verify the existence of an abstraction relation by applying Corollary 4.3.1 using



	0	1	2	3	4	5	6	7	8	9
1				▨						
2		▨								
3					▨					
4		▨								
5						▨				
6										▨

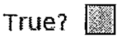
Figure 5.1: Relation P for Example 5.1.1.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1													▨							
2		▨																		
3																	▨			
4		▨																		
5																		▨		
6																			▨	

Figure 5.2: Relation Q for Example 5.1.1.

RELVIEW. By executing Program A.2.1 (Result $=$ *Test*(P, Q)*), we obtain the following result:*

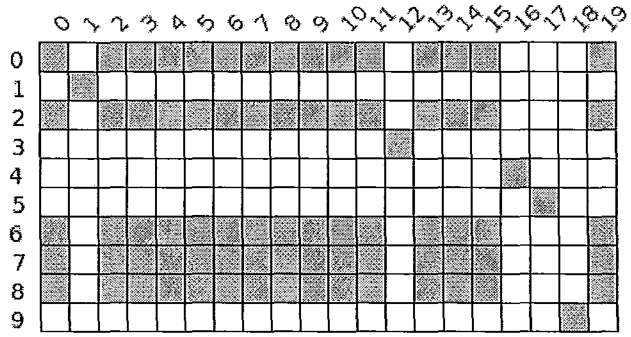


True? ▨

Figure 5.3: Relation *Result* for Example 5.1.1.

Therefore, the test has passed meaning that there exists an abstraction relation relating the confidential information to the information observed to be sent on the communication channel. This means that we can apply Corollary 4.3.2 by executing Program A.2.2 ($X = Compute(P, Q, \mathbb{L})$*) to obtain the abstraction relation,* X.

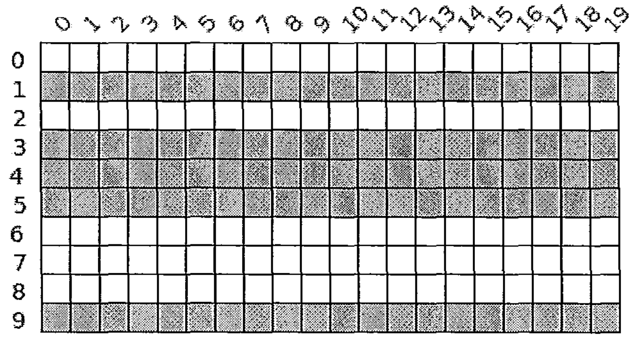
From this result, we can see that there are some digits which are related to information which we do not necessarily have an interest in, i.e., we are only concerned with



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨				▨
1		▨																		
2	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨				▨
3													▨							
4																	▨			
5																		▨		
6	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨				▨
7	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨				▨
8	▨		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨				▨
9																			▨	

Figure 5.4: Abstraction relation X for Example 5.1.1.

the confidential information which consists of the digits $1, 3, 4, 5,$ *and* 9. *Therefore we can design a filter* R *which can be used to refine the abstraction relation* X. *We define* R *in* ***RELVIEW*** *as follows:*



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
2																				
3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
4	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
5	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
6																				
7																				
8																				
9	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Figure 5.5: Filtering relation R for Example 5.1.1.

By executing Program A.2.2 with the filter R, $(X_{filtered} = Compute(P, Q, R))$, *we obtain the abstraction relation,* $X_{filtered}$

From the abstraction relation $X_{filtered}$ from Example 5.1.1, we can see that the digit 1 was transmitted as the number 1, the digit 3 was transmitted as the number 12, the

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1		▨																		
2																				
3													▨							
4																	▨			
5																		▨		
6																				
7																				
8																				
9																			▨	

Figure 5.6: Abstraction relation $X_{filtered}$ for Example 5.1.1.

digit 4 was transmitted as the number 16, the digit 5 was transmitted as the number 17, and the digit 9 was transmitted as the number 18.

Information that is encrypted and sent over a single communication channel is often the simplest case for covert channel communication. This type of communication can be detected by a direct application of Corollary 4.3.1 and Corollary 4.3.2. Because the monitor which is watching the transmission knows the confidential information and what it has observed to have been sent on the communication channel, there are no assumptions which need to be made in regards to the way the information is transmitted. However, we will see in the following sections how, with more complex schemes for communicating information over covert channels, we will need to begin to assume how information can be combined and reconstructed to form the abstraction of the confidential information.

5.2 Modulating the Confidential Information

Suppose that agent A and agent B develop a new idea to mask their covert communication of confidential information. Assume that the agents decide to modulate the confidential information prior to its encryption. This means that they will modify the confidential information by some scheme and then encrypt the modulated information so as to add another level of abstraction to the transmitted information.

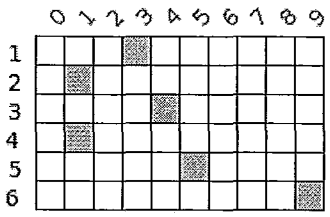
This idea illustrates the case when we are looking for a solution to an equation of the form $X \,;P \,;M = Q$ where

- X is the abstraction relation,
- P is the relation representing the confidential information,
- M is the modulation relation, and
- Q is the relation representing the information observed to be sent on the communication channel, which in this case is encrypted based on the modulated P.

This idea is best illustrated through Example 5.2.1.

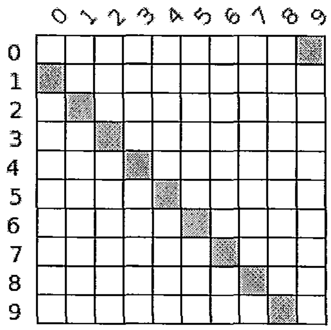
Example 5.2.1. *Assume the set of confidential information is given as* $P = \{(1,3),(2,1),(3,4),(4,1),(5,5),(6,9)\}$*. In order to obscure the transmission of the information, agent A modulates the confidential information by a relation represented by* $M = \{(0,9),(1,0),(2,1),(3,2),(4,3),(5,4),(6,5),(7,6),(8,7),(9,8)\}$ *prior to its encryption. Then, the new relation representing the confidential information is given by* $(P \,; M) = \{(1,2),(2,0),(3,3),(4,0),(5,4),(6,8)\}$*. This information is encrypted and* *sent on a single communication channel as* $Q = \{(1,11),(2,0),(3,12),(4,0),(5,16),(6,8)\}$.

We define the relations P, M and Q in **RELVIEW** *as follows:*



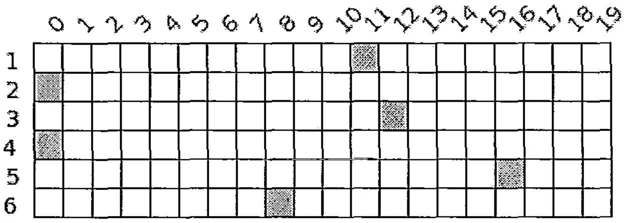
	0	1	2	3	4	5	6	7	8	9
1				▨						
2		▨								
3					▨					
4		▨								
5						▨				
6										▨

Figure 5.7: Relation P for Example 5.2.1.



	0	1	2	3	4	5	6	7	8	9
0										▨
1	▨									
2		▨								
3			▨							
4				▨						
5					▨					
6						▨				
7							▨			
8								▨		
9									▨	

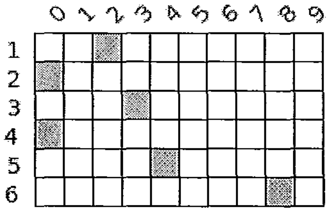
Figure 5.8: Modulation relation M for Example 5.2.1.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1												▨								
2	▨																			
3													▨							
4	▨																			
5																	▨			
6									▨											

Figure 5.9: Relation Q for Example 5.2.1.

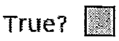
The modulated confidential information is represented in RELVIEW *as follows:*



	0	1	2	3	4	5	6	7	8	9
1			▨							
2	▨									
3				▨						
4	▨									
5					▨					
6									▨	

Figure 5.10: Relation $(P\,;M)$ for Example 5.2.1.

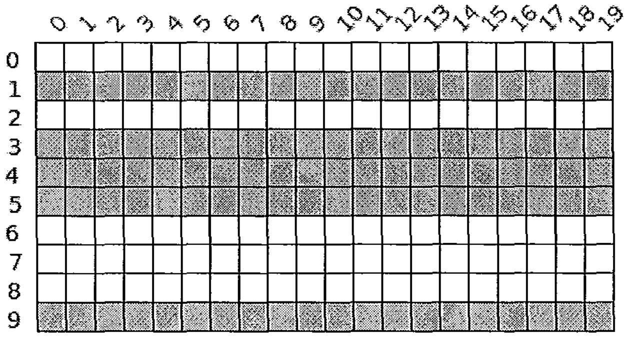
We verify the existence of an abstraction relation by executing Program A.2.1 ($Result = Test(P, Q)$). In this case we are looking for an abstraction relation relating the confidential information, P, and the information sent on the communication channel, Q, which corresponds to the encrypted modulated confidential information.



True? ▨

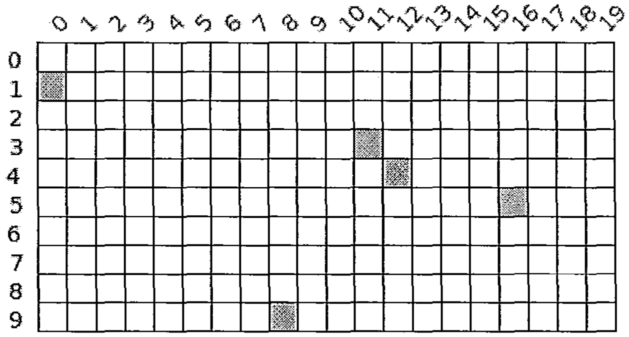
Figure 5.11: Relation *Result* for Example 5.2.1.

Therefore, the test has passed so we can compute the abstraction relation by executing Program A.2.2 ($X = Compute(P, Q, R)$) where R is the filtering relation.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
2																				
3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
4	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
5	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
6																				
7																				
8																				
9	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Figure 5.12: Filtering relation R for Example 5.2.1.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1	▨																			
2																				
3												▨								
4													▨							
5																	▨			
6																				
7																				
8																				
9									▨											

Figure 5.13: Abstraction relation X for Example 5.2.1.

From the abstraction relation X from Example 5.2.1, we can see that the digit 1 was transmitted as the number 0, the digit 3 was transmitted as the number 11, the digit 4 was transmitted as the number 12, the digit 5 was transmitted as the number 16, and the digit 9 was transmitted as the number 8.

Example 5.2.1 illustrates how the modulation of the confidential information prior to the encryption and transmission makes no difference on the ability to detect whether it has been leaked in some form. This highlights the point that the encryption of the information does not matter. Since we know the confidential information and we observe the information that is being sent on the channel, we do not need to know how the information was encrypted. If an abstraction relation exists between the confidential information, P, and the information observed to be sent on the channel, then even if a modulation of P by M is transmitted as Q, we can still find an abstraction relation relating P and Q without knowing what M is. This idea is presented in Corollary 5.2.1.

Corollary 5.2.1. *Let P be a relation containing confidential information. Let M be a total and injective relation that modulates the confidential information in some way such that the modulated confidential information is represented by $(P\,;M)$. Let Q be a relation representing an information observed on the monitored communication channel. If the confidential information contained in P is being leaked as that represented by Q then any modulation of the confidential information contained in P is also being leaked as that represented by Q, i.e.,* $\exists(X \mid: P\,;X = Q) \implies \exists(Y \mid: P\,;M\,;Y = Q)$.

Proof.

$\exists(X \mid: P\,;X = Q)$

$\iff \quad \langle$ Converse both sides & Proposition 2.2.1(vii) $\rangle$

$\exists(X \mid: X^{\smile}\,;P^{\smile} = Q^{\smile})$

$\iff \quad \langle$ Proposition 4.3.1 $\rangle$

$Q^{\smile} = (Q^{\smile}/P^{\smile})\,;P^{\smile}$

$\Longleftrightarrow$ $\langle$ Converse both sides & Proposition 2.2.1(ii) & Proposition 2.2.1(vii) & Proposition 2.2.4(i) $\rangle$

$Q = P\,;(P\backslash Q)$

$\Longleftrightarrow$ $\langle$ Definition 2.2.9(i) $\rangle$

$Q = P\,;\overline{P^\smile;\overline{Q}}$

$\Longrightarrow$ $\langle$ M is total $\Longleftrightarrow$ $\mathbb{I} \subseteq M\,;M^\smile$ $\rangle$

$Q = P\,;\overline{P^\smile;\overline{Q}} \subseteq P\,;M\,;M^\smile;\overline{P^\smile;\overline{Q}}$

$\Longleftrightarrow$ $\langle$ $M^\smile$ is deterministic $\Longleftrightarrow$ $M^\smile$ is injective $\rangle$

$Q = P\,;\overline{P^\smile;\overline{Q}} \subseteq P\,;M\,;\overline{M^\smile;P^\smile;\overline{Q}}$

$\Longleftrightarrow$ $\langle$ Proposition 2.2.1(vii) $\rangle$

$Q = P\,;\overline{P^\smile;\overline{Q}} \subseteq P\,;M\,;\overline{(P\,;M)^\smile;\overline{Q}}$

$\Longrightarrow$ $\langle$ Transitivity of $\subseteq$ $\rangle$

$Q \subseteq P\,;M\,;\overline{(P\,;M)^\smile;\overline{Q}}$

$\Longleftrightarrow$ $\langle$ Definition 2.2.9(i) & Proposition 2.2.4(vii) $\rangle$

$Q \subseteq P\,;M\,;\big((P\,;M)\backslash Q\big) \subseteq Q$

$\Longleftrightarrow$ $\langle$ Antisymmetry $\rangle$

$Q = P\,;M\,;\big((P\,;M)\backslash Q\big)$

$\Longleftrightarrow$ $\langle$ Proposition 4.3.1 $\rangle$

$\exists(Y \mid: P\,;M\,;Y = Q)$

□

In the case of modulating the confidential information, we require that the modulation relation be total and injective. The totality of the modulation relation ensures that, when composed with the relation containing the confidential information, no information is lost, i.e., all of the confidential information is represented in some form in the modulated confidential information. The injectivity of the modulation relation ensures that no inconsistencies are introduced which will cause the tests to fail. The introduction of an inconsistency in the modulated confidential information leads to a loss of information. The idea behind the conditions on the modulation relation is to ensure that no information is lost during the modulation of the confidential information.

5.3 Conclusion

The proposed technique has the ability to detect many different forms of covert channels employing various schemes for transmitting the information. The technique is used to carry out a post-mortem analysis of the information that has been observed on the communication channels. Knowing the confidential information and the information that has been sent, we are able to perform an analysis to detect whether the confidential information has been leaked in some form via covert channels. This sort of investigation relates to computer forensics investigations whereby with some systematic assumptions, we are able to detect whether or not the information was leaked in one way or another. It is important to note that we do not necessarily care how the information was leaked, but simply whether it has been leaked.

Chapter 6

Discussion

In this chapter, we discuss various aspects of the problem of covert channels and how this thesis aids in combatting them. In Section 6.1, we discuss some possible application domains for which the detection technique presented in Chapter 4 is suitable. We also discuss the importance of such techniques and applications. In Section 6.2, we assess the strengths and weaknesses of the main contributions.

6.1 Discussion

The problem of covert channels is a major concern as a threat to computer and information security. It is important to highlight the fact that we are aiming to develop techniques in order to detect the use of covert channels, not techniques to build better and more obscure covert channels. If we can gain a better understanding of how to detect the use of covert channels in computer systems, then we will be able to build the best covert channels since we will have the understanding of what can and cannot be detected.

The detection technique proposed in Chapter 4 is suitable for detecting the use of covert channels employed with various schemes. However, the technique can also be suitable in the computer forensics context. The technique analyzes the transmitted information in a manner which is after-the-fact, which by its very nature leads to an inherent relationship with a forensic investigation. Consider a case where the communication between two suspected criminals is intercepted and suspected to contain some type of important information, i.e., a date, a location, a name, etc. A forensic analyst equipped with the technique proposed in Chapter 4, is able to perform an investigation into the observed information transmitted between the suspected criminals. For example, the analyst can run the test for an abstraction relation between the observed information and the information for which he/she suspects may have been sent, i.e., the analyst may suspect that the transmission contains a suspect's name or a suspected location where the crime has taken place. If we allow these assumptions to form the confidential information, the proposed technique can be used to find an abstraction relation. If an abstraction relation can be found relating a suspect's name, for example, and the communicated information, it may be enough to begin to build a case against that suspect. It may also lead to the ability to uncover more information from the transmission between the two suspected criminals in that a cipher key may begin to form.. This sort of analysis is a suitable application for the detection technique presented in this thesis. To put this application of the detection technique into a real-world perspective, we will briefly look at the case of the Zodiac Killer [Dao07]. The Zodiac case illustrates where the proposed detection technique may be applied in a forensics context similar to the example given above.

The Zodiac was a serial killer who terrorized Northern California in the late 1960's. The serial killer sent four ciphers to local newspapers [Dao07]. The first cipher was separated into three different parts and each part was sent to three different newspapers: the Vallejo Times-Herald, the San Francisco Chronicle, and the San Francisco Examiner. The Zodiac requested each part be published on the front page of the respective newspapers such that the combination of all three parts formed a 408-character cipher, which was decrypted one week after it was received [Dao07]. The Zodiac also sent a 340-character cipher that remains unsolved to this day. The case of the Zodiac killer remains open in Napa County in California [Dao07]. By applying the proposed detection technique in the forensics context mentioned above, it may be possible to search the cipher-text for keywords that the killer may have used in an attempt to break the cipher key. The keywords may include words such as "kill", "zodiac", and others which the killer may be suspected of using. If an abstraction relation can be found for some of these words in the cipher-text, an analyst may be able to begin constructing the cipher key based on the obtained results.

The use of a covert channel detection technique in a computer forensics context is a new concept for generating investigative support for confidentiality. To the best of my knowledge, an application of a mathematical-based covert channel detection technique for computer forensics investigations is non-existent in the literature. The importance and necessity for this type of application seems to be growing day by day. Covert channels are real and are being used in real-world scenarios to smuggle information. For example, the recent investigation in the United State of America with regards to a Russian spy ring, has unfolded to reveal that the alleged spies were using various forms of covert communication including steganography, covert channels through email protocols and even Morse Code-like radio signals [Wil10]. According to [Ade10], new stenography programs use ephemeral channels that disappear when the communication has been completed. These new programs are being used to exfiltrate sensitive data in corporate espionage and state sponsored espionage. This example exemplifies the real threat of covert communication on security. The implications of the use of covert channel communication on the scale of international espionage highlights the importance of developing techniques to detect and prevent the use of covert channels in computer systems.

6.2 Assessment of the Contributions

In this section, we discuss the strengths and weaknesses of the main contributions presented in this thesis. It is important to highlight both the strengths and weaknesses of the models and techniques that are developed so that we are able to further refine a solution to the problem of covert channels and possibly one day eliminate their use completely.

6.2.1 Strengths of the Contributions

The ideas presented in this thesis aid in fight against the use of covert channels for leaking confidential information. We provided a new classification of the types of covert channels. The new classification organizes covert channels into *protocol-based covert channels* and *environment-based covert channels*. As far as I know, this classification is not found in the literature. The new classification offers a more clear and concise classification than those in the existing literature. A clear classification of covert channels gives a better understanding of covert channel communication.

We also proposed a model for protocol-based covert channels. The versatility and flexibility of the model allows it to be a powerful representation of protocol-based covert channel communication, which enables building better detection and prevention mechanisms for covert channels in computer systems. Its ability to represent steganography as a form of covert channel communication yields an inherent relationship between the two concepts. Since we are able to model both steganography and covert channel communication using the same model, we can argue that steganography is a special case of covert channel communication, something which is largely debated in the literature.

We presented a covert channel detection technique based on mathematics, in particular, relations, which offers a degree of simplicity and elegance. The technique is unlike anything that was found in the literature. It does not rely on heuristics to uncover the use of covert channels but rather gives a more formal and rigorous approach to detecting the leak of confidential information through covert channel communication. The technique provides simple tests to verify if an abstraction relation exists and computations to find the abstraction relation if it does exist. These tests and computations are expandable allowing for the technique to handle complex scenarios which may involve modulating the confidential information as demonstrated in Chapter 5.

6.2.2 Weaknesses of the Contributions

The detection technique proposed in this thesis does have some drawbacks. First and foremost, the tests do have the ability to fail under some conditions when there is an inconsistency between the confidential information and the information that is observed to be sent on the communication channel(s). Although this generally means that an abstraction relation does not exist which relates the confidential information to the observed information, it is possible that we can still find part of an abstraction relation which relates a large portion of the confidential information to the observed information. It is also unknown if the technique will scale well with larger systems. In addition, the technique performs a post-mortem analysis of the communicated information. This can be seen as a weakness of the technique since the damage may already be done in terms of confidential information being leaked and falling into the wrong hands. It would perhaps be better if the analysis could be done in real time. Lastly, the technique is restricted by the assumptions given in Section 4.1.

6.3 Conclusion

Covert channels present an ever-growing threat to computer security. We have discussed how the ideas presented in this thesis aid in providing a better understanding of covert channels. We looked at some real-world cases where covert channels have been used in order to smuggle information and how the proposed detection technique may be suitable for use in a computer forensics context. We also discussed the strengths and weaknesses of the main contributions of this thesis in order to allow for future work in the area of covert channels and investigative support for confidentiality.

Chapter 7

Conclusion and Future Work

In this thesis, we saw a new classification of covert channels. Covert channels can be classified as protocol-based covert channels or environment-based covert channels. We looked at some of the existing techniques for detecting and preventing covert channels in computer systems. We also examined the literature for common protocols which are used to carry covert information in protocol-based covert channels. Based on this survey, we developed a model for protocol-based covert channels and looked at how it can be used to model specialized cases of covert channels such as steganography and watermarking. We then formulated a new technique, based on relational algebra, to detect the leak of confidential information via covert communication channels. The detection technique offers tests and computations for which we can verify the existence of an abstraction relation which relates the confidential information with the information that has been observed to be sent on the communication channel. By examining some applications of the proposed technique through illustrative examples, we were able to demonstrate the ability of the technique to handle various cases of covert channel communication including the modulation of confidential information. Lastly, we discussed the idea of applying the proposed detection technique in a computer forensics context as investigative support tool for maintaining the confidentiality of information.

7.1 Future Work

In this section, we look at future work in the area of covert channels. We look at the possibility for modifications to the model proposed in Chapter 3 and the detection technique presented in Chapter 4. We look at the possibilities of extending the problem of covert channels to more general cases. We also discuss the application of the technique on real systems and networks and the possibility for new tools and automation of the technique.

7.1.1 Theory: Models and Techniques

There are a number of open issues in the proposed model and detection technique.

(i) Our model for covert channels is restricted to protocol-based covert channels. A model for environment-based covert channels needs to be developed which in turn would lead to a comprehensive model for general covert channels. Recall that with environment-based covert channels, resources available in the shared environment are used to communicate information. This means that the agents involved in the use of the covert channel will have some shared knowledge of the environment in which they are communicating. This leads to the necessity to develop a model which is able to account for modelling the knowledge of each agent involved in the communication.

(ii) It is uncertain how well the tests and computations of the proposed technique will scale. As the complexity of the covert channels increases and as the amount of information that is transmitted increases, will the tests and computations be able to be computed in a reasonable amount of time? An investigation into the computability and complexity of the tests and computations is needed in order to gain a understanding of the scalability of the proposed detection technique.

(iii) It is required that research be done to investigate how to combine information that has been sent across multiple covert communication channels in order to detect if there has been a leak of confidential information. This will involve characterizing operations which may be used in order to effectively combine the information observed on multiple channels such that no information is lost in the combination.

(iv) It is unclear how communication channels can be effectively sampled for random testing to determine if any confidential information is being leaked. When we sample a large stream of information, it is possible that we will be sampling a portion of the communication stream which was leaking confidential information. However, rather than the sample containing the confidential information in its entirety, we may only have a portion of it. There is a need to be able to detect whether a part of the confidential information is contained in the sampled communication stream. This will involve an investigation into the *part of* relation [JK01].

(v) The proposed model and detection technique handle only a specific set of covert channels, i.e., protocol-based covert channels where the common knowledge the sequence of the information (time). It is required that this idea be extended in order to tackle the most general covert channel possible where we have a combination of environmental and protocol-based knowledge for each agent.

7.1.2 Applications

There are a number of applications of the proposed covert channel detection technique which need to be investigated.

(i) The tests and computations need to be implemented on real networks with large-scale traffic. With the tests running on real network traffic with covert information being transmitted, we can better understand the complexities involved in detecting covert channels in real-world scenarios and develop insight into the scalability of the proposed detection technique.

(ii) A push towards real-time analysis of the communication channels needs to be examined. This may involve work on developing a hierarchical monitoring system and an effective sampling mechanism in order to monitor the communication channels for a potential leak of confidential information.

(iii) More research into the applicability of the proposed detection technique for computer forensics investigations is needed. This research will likely involve more exploration into developing alternative tools and mechanisms for investigative support for confidentiality.

7.1.3 Tools/Automation

The tests and computations related to the produced detection technique are currently automated using the RELVIEW tool. However, the use of RELVIEW to handle large scale covert channels may be limited in that certain combining operations that can be defined on relations are unavailable and are difficult to implement. The development of a more sophisticated and configurable automation system to handle larger scale covert channels and to implement the capabilities of the communication monitors presented in Section 4.3.1 would be ideal.

7.2 Closing Remarks

The quest to eliminate the use of covert channels from computer systems is an ongoing struggle. It is argued that covert channels can never be completely eliminated from computer systems. Of course, covert communication will constantly evolve in order to elude detection mechanisms. There will always be a human aspect to covert channel communication which means that the challenge of completely eliminating covert channels is indeed a daunting task. However, with an evolving understanding of covert channel communication, it is possible to at least make it difficult for covert channel communication to be effective.

Appendix A

RELVIEW

To aid in the computation of the examples presented in this thesis, we use a tool called RELVIEW. RELVIEW is an interactive tool for computer-aided manipulation of relations represented as Boolean matrices. It is developed at the Department of Computer Science and Applied Mathematics at Christian-Albrechts-University in Kiel, Germany [Ber09]. This appendix presents an overview of working with RELVIEW and the programs developed in RELVIEW to automate the tests and computations of the corollaries presented in Chapter 4.

A.1 Working With RELVIEW

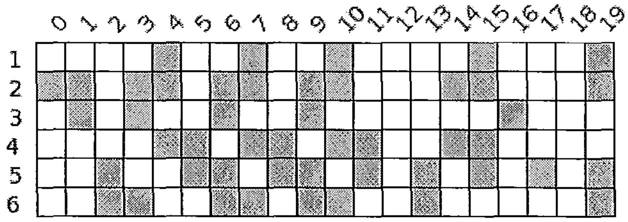
In this section, we look at how to work with relations using the RELVIEW tool. The information presented in this section is taken from [BBS09].

A.1.1 Representing Relations

RELVIEW is able to represent relations both as Boolean matrices and as an ASCII description.

Boolean Matrix Representation

The Boolean matrix representation of relations in RELVIEW is a graphical representation. A relation is given as a matrix where the rows represent the domain of the relation and the columns represent the range of the relation. A filled in cell of the matrix represents that element being included in the relation. An example of Boolean matrix representation of a relation in RELVIEW is given in Figure A.1.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1					▨			▨			▨					▨				▨
2	▨	▨		▨	▨		▨	▨		▨	▨				▨	▨				▨
3		▨		▨			▨			▨							▨			
4					▨	▨		▨	▨		▨	▨			▨	▨				
5			▨			▨	▨		▨	▨		▨		▨		▨		▨		▨
6			▨	▨			▨	▨		▨	▨			▨						▨

Figure A.1: Example of the Boolean matrix representation of a relation in RELVIEW.

ASCII Representation

The ASCII representation of relations in RELVIEW is a textual representation. A relation is given as a list entries of the form "Domain : Range". The ASCII representation of the relation given in Figure A.1 is given below.

```
R (6, 20)
1 : 5, 8, 11, 16, 20
2 : 1, 2, 4, 5, 7, 8, 10, 11, 15, 16, 20
3 : 2, 4, 7, 10, 17
4 : 5, 6, 8, 9, 11, 12, 15, 16
5 : 3, 6, 7, 9, 10, 12, 14, 16, 18, 20
6 : 3, 4, 7, 8, 10, 11, 14, 20
```

A.1.2 Operations

Syntax	Description
$-R$	Complement of relation R
$R \mid S$	Union (join) of R and S
$R \,\&\, S$	Intersection (meet) of R and S
$R + S$	Relational sum of R and S

Table A.1: Boolean Operations

Syntax	Description
$R\hat{}$	Converse of relation R
$R * S$	Composition of R and S

Table A.2: Relational Algebraic Operations

Syntax	Description
S/R	Left residue of R and S
$R\backslash S$	Right residue of R and S
$syq(R,S)$	Symmetric quotient of R and S

Table A.3: Residuals and Symmetric Quotients

Syntax	Description
$eq(R, S)$	Test, whether R and S are equal
$incl(R, S)$	Test, whether R is included in S

Table A.4: Relational Tests

A.1.3 Labels

Labels are organized into sets which are mappings from natural numbers to labels or identifiers.

The labels that are used in this thesis are given below:

```
Digit = { 1 ”0”, 2 ”1”, 3 ”2”, 4 ”3”, 5 ”4”, 6 ”5”, 7 ”6”, 8 ”7”, 9
    ”8”, 10 ”9” }

Encryption = { 1 ”0”, 2 ”1”, 3 ”2”, 4 ”3”, 5 ”4”, 6 ”5”, 7 ”6”, 8
    ”7”, 9 ”8”, 10 ”9”, 11 ”10”, 12 ”11”, 13 ”12”, 14 ”13”, 15 ”14”,
    16 ”15”, 17 ”16”, 18 ”17”, 19 ”18”, 20 ”19” }

Bool = { 1 ”True?” }
```

These labels are used in the Boolean matrix representation of relations making them easier to read and understand. The label “Digit” corresponds to the digit data type, the label “Encryption” corresponds to the natural numbers which can be used to encrypt the digits, and the label "Bool" simply adds a descriptive label for boolean results.

It is important to note that when representing relations using labels, the ASCII representation of the relation must correspond to the natural number and not the label or identifier. For example, if we want to represent the digit 4 being sent at time 1, i.e., $(1, 4)$ we must use "1 : 5" in the ASCII representation so that the label corresponds to the digit 4.

A.1.4 Truth Values

In `RELVIEW`, the result of a Boolean operation is a 1×1 Boolean matrix with the truth values corresponding to $\mathbb{L} = \mathsf{true}$ and $\emptyset = \mathsf{false}$. This is to say that the truth values are given by the Boolean matrices given in Figure A.2.

(a) True (b) False

Figure A.2: `RELVIEW` representation of truth values.

A.2 `RELVIEW` Programs

Program A.2.1 represents the test outlined in Corollary 4.3.1.

Program A.2.1.

```
Test(p,q)
    DECL test1 , test2 , res
    BEG  test1 = eq(p,q*(q\p));
         test2 = eq(q,p*(p\q));
         res = test1 | test2
         RETURN res
    END.
```

Program A.2.2 corresponds to the computations presented in Corollary 4.3.2.

Program A.2.2.

```
Compute(p, q, r)
    DECL test1 , test2 , res
    BEG  test1 = incl(p,q*(r^ & (q\p)));
         test2 = incl(q,p*(r  & (p\q)));
         IF test1 & test2
            THEN res = r & syq(p,q)
            ELSE IF test1
               THEN res = r & (q\p)^
               ELSE IF test2
                  THEN res = r & (p\q)
                  ELSE res = false
               FI
            FI
         FI
         RETURN res
    END.
```

Program A.2.3 automates the computation given in Corollary 4.3.3.

Program A.2.3.

```
ComputeBij(p, q, r)
    DECL test1 , test2 , res
    BEG  IF eq((p\q),(q\p)^)
            THEN res = r & (p\q)
            ELSE res = false
         FI
         RETURN res
    END.
```

Bibliography

[Ade10] S. Adee. Russian spies thwarted by old technology? IEEE Spectrum, June 29 2010.

[AR80] G.R. Andrews and R.P. Reitman. An axiomatic approach to information flow in programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 2(1):56 – 76, January 1980.

[BBS09] R. Behnke, R. Berghammer, and P. Schneider. Machine support of relational computations: The kiel relview system. Technical Report 9711, Christian-Albrechts-University of Kiel, 2009.

[Ber07] H. Berghel. Hiding data, forensics, and anti-forensics. *Communications of the ACM*, 50(4):15–20, April 2007.

[Ber09] Rudolf Berghammer. Relview. Available: `http://www.informatik.uni-kiel.de/~progsys/relview.shtml` (Accessed: July 27, 2010), July 2009.

[Bis02] Matt Bishop. *Computer Security: Art and Science.* Addison Wesley, Boston, MA, November 2002.

[BR05] R. Bidou and F. Raynal. Covert channels. November 2005.

[Bro94] R. Browne. Mode security: An infrastructure for covert channel suppression. In *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 39 – 55, Los Almitos, CA, USA, 1994.

[Com05] D.E. Comer. *Internetworking with TCP/IP*, volume 1. Prentice Hall, fifth edition, 2005.

[Dao07] T. Dao. Analysis of the zodiac 340-cipher. Master's thesis, San Jose State University, December 2007.

[DoD85] United States of America Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria.* Number DoD 5200.28-STD in Defense Department Rainbow Series. Department of Defense / National Computer Security Center, Fort George G. Meade, Maryland, December 1985.

[FK98] H. Furusawa and W. Kahl. A study on symmetric quotients. Technical Report 1998-06, Fakultät für Informatik, Universität der Bundeswehr München, December 1998.

[GKT05] A. Grusho, A. Kniazev, and E. Timonina. Detection of illegal information flow. In *Proceedings of the Third International Workshop on Mathematical Methods, Models, and Architectures for Computer Networked Security, MMM-ACNS 2005*, number 3685 in Lecture Notes in Computer Science, pages 235 – 244, Berlin, Germany, 2005.

[Gli93] V.D. Gligor. *A Guide to Understanding Covert Channel Analysis of*

Trusted Systems. Number NCSC-TG-030 in NSA/NCSC Rainbow Series. National Security Agency / National Computer Security Center, Fort George G. Meade, Maryland, November 1993.

[GM82] J.A. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the 1982 Symposium on Security and Privacy*, pages 11 – 20, New York, NY, USA, 1982.

[Gra00] J. W. Gray. Countermeasures and tradeoffs for a class of covert timing channels. Technical Report HKUST-CS94-18, Hong Kong University of Science and Technology, 2000.

[GS93] D. Gries and F.B. Schenider. *A Logical Approach to Discrete Math.* Springer Texts And Monographs In Computer Science. Springer-Verlag, New York, 1993.

[HH86a] C.A.R. Hoare and J. He. The weakest prespecification, part i. *Fundamenta Informaticae*, 1986.

[HH86b] C.A.R. Hoare and J. He. The weakest prespecification, part ii. *Fundamenta Informaticae*, 1986.

[Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576 – 580, October 1969.

[HZD05] L. Hélouët, M. Zeitoun, and A. Degorre. Scenarios and covert channels: Another game... *Electronic Notes in Theoretical Computer Science*, 119:93 – 116, 2005.

[HZJ03] L. Hélouët, M. Zeitoun, and C. Jard. Covert channels detection in protocols using scenarios. In *Proceedings of Security Protocols Verification, SPV'03*, pages 21 – 25, 2003.

[Jac90] J. Jacob. Separability and the detection of hidden channels. *Information Processing Letters*, 34(1):27 – 29, February 1990.

[JK01] R. Janicki and R. Khedri. On a formal semantics of tabular expressions. *Science of Computer Programming*, 39:189 – 213, March 2001.

[Kem83] R.A. Kemmerer. Shared resource matrix methodology: An approach to identifying storage and timing channels. *ACM Transactions on Computer Systems*, 1(3):256 – 77, August 1983.

[Khe98] R. Khedri. *Concurrence, Bisimulation et Équation d'Interface: Une Approche Relationnelle.* PhD thesis, Université Laval, April 1998.

[KM93] M.H. Kang and I.S. Moskowitz. A pump for rapid, reliable, secure communication. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 119 – 129, Fairfax, VA, USA, 1993.

[KP91] R.A. Kemmerer and P.A. Porras. Covert flow trees: A visual approach to analyzing covert storage channels. *IEEE Transactions on Software Engineering*, 17(11):1166 – 1185, November 1991.

[Lam73] B.W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613 – 615, October 1973.

[LMST+04] R. Lanotte, A. Maggiolo-Schettini, S. Tini, A. Troina, and E. Tronci. Automatic covert channel analysis of a multilevel secure component. In

Proceedings of the 6th International Conference, ICICS 2004, number 3269 in Lecture Notes in Computer Science, pages 249 – 261, Berlin, Germany, 2004.

[Low02] G. Lowe. Quantifying information flow. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop, CSFW-15*, pages 18 – 31, Los Alamitos, CA, USA, 2002.

[LT07] T.Y. Liu and W.H. Tsai. A new steganographic method for data hiding in microsoft word documents by a change tracking technique. *IEEE Transactions on Information Forensics and Security*, 2(1):24 – 30, March 2007.

[LT10] I. Lee and W. Tsai. A new approach to covert communication via pdf files. *Signal Processing*, 90(2):557 – 565, 2010.

[NCSC93] United States of America National Computer Security Center. *A Guide to Understanding Covert Channel Analysis of Trusted System.* Number NCSC-TG-030 in NSA/NCSC Rainbow Series. Department of Defense / National Computer Security Center, Fort George G. Meade, Maryland, November 1993.

[NW06] N. Nagatou and T. Watanabe. Run-time detection of covert channels. In *Proceedings of the First International Conference on Availability, Reliability and Security, ARES 2006*, pages 577 – 584, Vienna, Austria, 2006.

[PAK99] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn. Information hiding - a survey. *Proceedings of the IEEE*, 87(7):1062 – 1078, July 1999.

[PK91] P.A. Porras and R.A. Kemmerer. Covert flow trees: A technique for identifying and analyzing covert storage channels. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 36 – 51, Los Alamitos, CA, USA, 1991.

[PSCS07] A. Patel, M. Shah, R. Chandramouli, and K.P. Subbalakshmi. Covert channel forensics on the internet: Issues, approaches, and experiences. *International Journal of Network Security*, 5(1):41 – 50, July 2007.

[RGI06] N. Ravi, M. Gruteser, and L. Iftode. Non-inference: An information flow control model for location-based services. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Systems*, pages 206 – 215, Piscataway, NJ, USA, 2006.

[RMMG01] P. Ryan, J. McLean, J. Millen, and V. Gligor. Non-interference: Who needs it? In *Proceedings of the 14th IEEE workshop on Computer Security Foundation*, pages 237 –238, Washington, DC, USA, 2001. IEEE Computer Society.

[Rus93] B. Russel. *Introduction to Mathematical Philosophy.* Routledge, 1993.

[Sar06] B. Sartin. Anti-forensics - distorting the evidence. *Computer Fraud and Security*, 2006(5):4 – 6, May 2006.

[SC99] S. Shieh and A.L.P. Chen. Estimating and measuring covert channel bandwidth in multilevel securc operating systems. *Journal of Information Science and Engineering*, 15(1):91 – 106, 1999.

[SK06] M. Smeets and M. Koot. Research report: Covert channels. Master's thesis, University of Amsterdam, February 2006.

[SKJ09] K.E. Sabri, R. Khedri, and J. Jaskolka. Verification of information flow in agent-based systems. In G. Babin, P. Kropf, and M. Weiss, editors, *Proceedings of the 4th International MCETECH Conference on e-Technologies*, volume 26 of *Lecture Notes in Business Information Processing*, pages 252 – 266. Springer Berlin / Heidelberg, May 2009.

[Sri06] S. Srinivasan. Security and privacy in the computer forensics context. In *Prooceedings of the 2006 International Conference on Communication Technology*, page 3, Piscataway, NJ, USA, November 2006. IEEE Computer Society.

[SS93] G. Schmidt and T. Ströhlein. *Relations and Graphs: Discrete Mathematics for Computer Science*. Springer-Verlag, 1993.

[TJ10] Z. Trabelsi and I. Jawhar. Covert file transfer protocol based on the ip record route option. *Journal of Information Assurance and Security*, 5(1):64–73, 2010.

[VH06] M. Van Horenbeeck. Deception on the network: Thinking differently about covert channels. In *Proceedings of the 7th Australian Information Warfare and Security Conference*. 174 - 184, December 2006.

[VS97] D. Volpano and G. Smith. Eliminating covert flows with minimum typings. In *Proceedings of the 10th Computer Security Foundations Workshop*, pages 156 – 168, Los Alamitos, CA, USA, 1997.

[Wil10] C. Williams. Russian spy ring bust uncovers tech toolkit. The Register, June 29, 2010.

[WW90] R.J. Wilson and J.J. Watkins. *Graphs: An Introductory Approach.* Wiley, New York, January 1990.

[ZAB07] S. Zander, G. Armitage, and P. Branch. Covert channels and countermeasures in computer network protocols. *IEEE Communications Magazine*, 45(12):136 – 142, December 2007.

[ZLSN05] X. Zou, Q. Li, S. Sun, and X. Niu. The research on information hiding based on command sequence of ftp protocol. In *Proceedings of 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, volume 3683 of *Lecture Notes in Computer Science*, pages 1079–1085. Springer Berlin / Heidelberg, 2005.

Index