The 9th International Conference on Future Networks and Communications (FNC-2014)

Investigative Support for Information Confidentiality
Part I: Detecting Confidential Information Leakage via Protocol-Based Covert Channels

Jason Jaskolka\textsuperscript{a,}\textsuperscript{*}, Ridha Khedri\textsuperscript{a}, Khair Eddin Sabri\textsuperscript{b}

\textsuperscript{a} Department of Computing and Software, Faculty of Engineering, McMaster University, Hamilton, Ontario, Canada
\textsuperscript{b} Department of Computer Science, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

Abstract

This is Part I in a two-part series discussing the development of investigative support for information confidentiality. In this paper, we propose a technique to detect confidential information leakage via protocol-based covert channels based on relation algebra. It provides tests to verify the existence of an information leakage via a monitored covert channel as well as computations which show how the information was leaked if a leakage exists. We also report on a prototype tool that allows for the automation of the proposed technique. Our focus is limited to protocol-based covert channels and instances where covert channel users modulate the sent information by some form of encoding such as encryption.

1. Introduction and Motivation

Today, the protection of large amounts of sensitive information is at the forefront of security concerns. In modern organisations, the prospect of the leakage of sensitive information ranks among the highest fears of any executive\textsuperscript{1}. As many of these organisations depend on broad and heterogeneous communication networks, there are numerous possibilities for the exfiltration of sensitive information and the detection of such threats presents many challenges. With the ongoing threat of insiders with the intent to inappropriately access and divulge information, we must strive to devise new support to investigate those individuals who may be responsible for the breach of confidentiality.

A security policy, or more specifically, a confidentiality policy limits access to certain information. As a result, confidential information is defined as the information that is protected by a confidentiality policy. This paper focusses on detecting confidential information leakage via protocol-based covert channels. Covert channels can be used to transmit confidential information in a secret manner. A covert channel is any communication means that can be exploited to transfer information in a manner that violates the security policy\textsuperscript{2}. In particular, a protocol-based covert channel uses the communication protocol to convey messages that violate the security policy. A thorough survey of covert channels and a model for their morphology is given in Jaskolka & Khedri\textsuperscript{3}.

\textsuperscript{*} Corresponding author. Tel.: +1-905-525-9140 ; fax: +1-905-524-0340.
\textit{E-mail address:} jaskolj@mcmaster.ca (Jason Jaskolka).
There are shortcomings in the science, mathematics, and fundamental theory to deal with covert channels in modern computer systems. Due to a lack of use of formal methods, protocol specifications often allow for uses in unanticipated or unintended ways which enable the establishment of covert channels whilst adhering to the specification. The resulting transmissions cannot be considered anomalous, leading to the difficulty of detecting such channels. We propose mathematical formulations for protocol-based covert channels and develop a formal and rigorous technique for the detection of confidential information leakage via protocol-based covert channels using algebraic techniques. A formal technique gives us more power and flexibility than that which could be done with heuristics. It also allows us to mechanise 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 for which we strive for confidentiality.

To the best of our knowledge, a formal technique such as the one we propose does not exist. Several works (e.g., Héloü et al., Roumy) examine covert channel detection and analysis from the perspective of information theory by, for instance, analysing channel capacities. We propose a technique that tackles the problem from a different perspective. This leads to fertile grounds for developing a theory of covert channels and provides us with new and innovative ways to handle the threat of covert channels to the confidentiality of information. In particular, we look to help in detecting streams of confidential information being leaked from a data store over an extended period of time as a means of providing investigative support for information confidentiality.

In Section 2, we introduce the required mathematical background including relations and their operations. In Section 3, we describe the process by which we formulate a technique to detect confidential information leakage via protocol-based covert channels. In Section 4, we present the proposed technique. In Section 5, we examine the automation of the proposed technique using a prototype tool implemented in the functional programming language Haskell. In Section 6, we provide a brief discussion of the proposed technique and related work. Finally, in Section 7, we draw conclusions and suggest future work.

2. Mathematical Background

Let \( X \) and \( Y \) be two sets. A relation \( R \) on \( X \times Y \) is a subset of the Cartesian product \( X \times Y \). In the rest of this paper, \( \mathbb{1} \) denotes the identity relation, \( \mathbb{L} \) denotes the universal relation, and \( \emptyset \) denotes the empty relation.

**Definition 1 (e.g., Schmidt & Ströhlein).** Let \( P, Q \subseteq X \times Y \) be relations.

1. \( P \circ Q \triangleq \{ (x, z) \mid \exists y \in Y : (x, y) \in P \land (y, z) \in Q \} \) is the relational composition of \( P \) and \( Q \).
2. \( P^- \triangleq \{ (x, y) \mid (y, x) \in P \} \) is the converse of \( P \).
3. \( \overline{P} \triangleq \{ (x, y) \mid (x, y) \notin P \} \) is the complement of \( P \).

**Definition 2 (e.g., Schmidt & Ströhlein).** If \( R \) is a relation, then we say

1. \( R \) is total \( \iff \forall S : R \circ S \subseteq R \circ S \)
2. \( R \) is univalent \( \iff \forall S : R \circ S \subseteq R \circ S \)
3. \( R \) is a mapping \( \iff \forall S : R \circ S = R \circ S \)
4. \( R \) is surjective \( \iff R^- \) is total
5. \( R \) is injective \( \iff R^- \) is univalent
6. \( R \) is bijective \( \iff R \) is surjective and injective

Residues are special operations on relations that help solve equations of the form \( P \circ X = Q \) or \( X \circ P = Q \).

**Definition 3 (Schmidt & Ströhlein).** Let \( P \) and \( Q \) be two relations.

1. \( \overline{Q} : P^- \) is the left residue of \( Q \) by \( P \)
2. \( P \setminus Q \triangleq P^- \setminus Q \) is the right residue of \( Q \) by \( P \)

The left residue gives the greatest relation \( X \) such that \( P \circ X \subseteq Q \) (see Proposition 1(1)). Similarly, the right residue gives the greatest relation \( X \) such that \( X \circ P \subseteq Q \) (see Proposition 1(2)).

**Proposition 1.** Let \( P, Q \) and \( X \) be relations.

1. \( X \subseteq P \cup Q \iff X \subseteq P \cup Q \)
2. \( X \subseteq P \cup Q \iff X \subseteq P \cup Q \)

**Proof.** The proof can be found in Schmidt & Ströhlein. □

In some cases, it is required that a relation be a left residue and right residue simultaneously.

**Definition 4 (e.g., Schmidt & Ströhlein).** For relations \( P \) and \( Q \), the symmetric quotient is defined as

\[ \text{syq}(P, Q) \triangleq P^- \setminus Q \cup P \setminus Q^- = (P \setminus Q) \cap (Q \setminus P^-). \]

The symmetric quotient \( \text{syq}(P, Q) \) is defined as the greatest relation \( X \) such that \( P \circ X \subseteq Q \) and \( X \circ Q^- \subseteq P^- \).
3. Formulation of a Detection Technique

3.1. Assumptions

Suppose that we have a system with at least two communicating agents. We assume that 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 assumption is common in the literature and ensures that the receiving agent is able to recover the communicated information. We also assume that the communication among agents is recorded by monitors which begin recording when a communication channel is established and that they maintain an unbounded history of all of the communication which has taken place. This assumption ensures that the monitors have a record of all of the transmissions between the communicating agents. A similar assumption is made in the literature, where the monitors or wardens have access to all of the messages passed between the communicating agents. Finally, we assume that the monitor always knows the set of confidential information that is responsible to watch.

3.2. Representing Information Streams as Relations

A crucial step in the formulation of the problem of detecting confidential information leakage via covert channels is finding an appropriate abstract representation for the information being sent on a channel. 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, an element of information is sent. This is commonly the case when data is serialised during communication. In this way, the data is concretely being transmitted as a stream of information. A stream of information can be seen 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 \( R = \{(1, 'h'), (2, 'e'), (3, 'l'), (4, 'l'), (5, 'o')\} \), where ‘h’ was sent at time 1, ‘e’ was sent at time 2, and so on.

In order to uncover a confidential information leakage via a protocol-based covert channel, we show that it is sufficient to find an abstraction relation between the confidential information and the stream of information observed to have been sent on the channel. An abstraction relation \( X \) can be seen as a simulation relation between two relations \( P \) and \( Q \). In Figure 1, the relation \( P \) represents the confidential information which should not be sent on the channel, the relation \( Q \) (henceforth referred to as the observed information) represents the information observed by a monitor watching the information sent on the communication channel, and the relation \( X \) represents an abstraction relation that relates \( P \) and \( Q \) (or \( \mathbb{D}_P \) to \( \mathbb{D}_Q \)).

An abstraction relation \( X \) requires that the diagram given in Figure 1 commutes. After simplification, we find that the diagram in Figure 1 commutes in two ways, characterised by the following equations.

\[
\begin{align*}
(1) \quad Q \cdot X^{-} &= P \\
(2) \quad P \cdot X &= Q
\end{align*}
\]

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. We are looking to find a solution to Equation (1) or Equation (2) which is an abstraction relation \( X \) relating the relations \( P \) and \( Q \). In

---

Footnote:

* The diagram in Figure 1 actually commutes in four ways. The reader can find details on the commutativity of Figure 1 and the simplification of the diagrams in Jaskolka et al.11.
terms of covert channels, the solution is an abstraction relation relating the confidential information and the observed information.

An information leakage is considered to be detected if and only if there exists an abstraction relation between the confidential information and the observed information such that the abstraction relation is different from $\emptyset$ and $\mathbb{L}$. The case where an abstraction relation is equal to $\emptyset$ indicates that there is no abstraction and thus no relationship between the confidential information and the observed information. The case where an abstraction relation is equal to $\mathbb{L}$ indicates that all information is related to all other information. In this case, the abstraction becomes irrelevant. We allocate a complete subsection (see Section 6.1) to discuss this matter and its impact on the scope of applicability of the proposed technique.

4. The Proposed Detection Technique

The proposed technique for detecting confidential information leakage via protocol-based covert channels has two major components: monitoring the information sent on the communication channel and finding and computing an abstraction relation relating the confidential information to the observed information.

4.1. Monitoring the Communication Channels

If an organisation wishes to detect if its confidential information is being leaked outside of the organisation, it can install a monitor on the known communication channels from which an agent from within the organisation can communicate with an agent outside the organisation. The monitor can perform either a post-mortem analysis\(^b\) or a periodic real-time analysis looking for an abstraction relation between the confidential information and the observed information. We can view the monitor as a specialised and customisable 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 protocol in which the communicating agents are using in order to communicate, the header of the particular protocol which is being used as the carrier for the covert information, the confidential information, and 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.

The monitor watches the stream of packets being transmitted on a communication channel. Based on its configuration, the monitor can either extract the header field from the protocol packets that it is monitoring as they are being transmitted and store them, or it can mirror and store the packets of the protocol and then extract the prescribed header field at a later time so as not to interrupt the performance of the communication channel. The monitor needs to determine whether the confidential information has been leaked in any capacity by verifying the existence of an abstraction relation $X$ which relates the confidential information and the observed information.

4.2. Finding an Abstraction Relation

In the general context, the existence of an abstraction relation $X$ which relates two relations $P$ and $Q$ can be verified using Proposition 2.

**Proposition 2.** $X : P = Q$ has a solution if and only if $Q = (Q/P) : P$.

**Proof.** The proof involves the application of Proposition 1, trading rules, the One-Point Axiom and the isotony of relational composition. The detailed proof can be found in Jaskolka et al.\(^{11}\).

Proposition 2 is used as a test to verify if there is an abstraction between the relations $P$ and $Q$. It should be noted that in Proposition 2, $P$ and $Q$ are merely relations and do not explicitly represent the confidential information or observed information in particular. However, it is easy to see that if we take the converse of both sides of $X : P = Q$, we obtain $P^{-1} : X^{-1} = Q^{-1}$. Then, we can have relations $P' = Q^{-1}$ and $Q' = P^{-1}$ representing the confidential and observed

\(^b\) 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.
information respectively. As a result, we are left with Equation (1). In a similar fashion, we can obtain an equation corresponding to Equation (2). In this way, the test is directly related to Figure 1 in that if the test holds, we can say that the diagram in Figure 1 commutes and we can find an abstraction relation $X$ that satisfies Equation (1) or Equation (2). Therefore, we can say that the confidential information $P$ seems to have been leaked using the abstraction given by $X$ and was sent as the observed information $Q$.

**Corollary 1.** Let $P$ be the relation containing confidential information. Let $Q$ be a relation representing the observed information on a monitored communication channel. The confidential information contained in $P$ is being leaked as that represented by $Q$ if and only if $P = Q \cdot (Q \setminus P) \lor Q = P \cdot (P \setminus Q)$.

**Proof.** The proof involves the application of Proposition 2 and basic properties of relations and residues. The detailed proof can be found in Jaskolka et al.\textsuperscript{11}.

In Corollary 1, we verify whether the diagram in Figure 1 commutes in at least one of the two ways characterised by Equation (1) and Equation (2). Each term of the disjunction in the test corresponds to one of the ways in which Figure 1 can commute. Therefore, as long as we can satisfy at least one of the ways in which the diagram commutes, we can find an abstraction relation $X$ that satisfies Equation (1) or Equation (2). Then, provided that the abstraction relation $X$ is not the empty relation $\emptyset$ or the universal relation $\mathbb{I}$, we can conclude that the confidential information has been leaked via the communication channel on which the observed information was sent. In this way, the condition in Corollary 1 establishes the necessary and sufficient conditions for the commutativity of Figure 1.

### 4.3. Computing the Abstraction Relation

Using Proposition 3, 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 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 3 provides the general solution to an equation $X : P = Q$.

**Proposition 3.** Let $P$, $Q$, and $R$ be relations. $X : P = Q$ has a solution $X = R \cap (Q / P)$ if and only if $Q \subseteq (R \cap (Q / P)) : P$.

**Proof.** The proof involves the application of Definition 3, basic properties of relations and residues, as well as anti-symmetry and weakening. The detailed proof can be found in Jaskolka et al.\textsuperscript{11}.

The relation $R$ plays the role of a filter. A filter allows for the removal of some unwanted elements of the transmitted sequences. In its most general case, if we consider the filter $R$ to be the universal relation $\mathbb{I}$, then we are interested in all of the elements of the transmission. 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 $R$. For instance, if we suspect that the confidential information is sent using only a subset $S_P$ of $D_P$, then we can filter using the relation $R = \{ (d_p, d_q) \mid d_p \in S_P \land d_q \in D_Q \}$. By computing an abstraction relation which is not the empty relation $\emptyset$ or the universal relation $\mathbb{I}$, we can say that we have uncovered a leakage of confidential information on the communication channel.

**Corollary 2.** Let $P$ be the relation containing confidential information. Let $Q$ be a relation representing the observed information 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 $X$ such that

1. $X = R \cap (Q \setminus P)^\sim$ if and only if $P \subseteq Q : (R^\sim \cap (Q \setminus P))$
2. $X = R \cap (P \setminus Q)$ if and only if $Q \subseteq P : (R \cap (P \setminus Q))$
3. $X = R \cap \text{sy}(P, Q)$ if and only if $P \subseteq Q : (R^\sim \cap (Q \setminus P)) \land Q \subseteq P : (R \cap (P \setminus Q))$

**Proof.** The proofs involve the application of Proposition 3 and basic properties of relations. The proof for (3) also involves the Golden Rule Axiom. The detailed proof can be found in Jaskolka et al.\textsuperscript{11}.

Corollary 2 gives three cases for which we can compute the abstraction relation $X$ in our specific context. In each case, we compute the abstraction relation $X$ according to the way(s) in which Figure 1 commutes.
4.4. Modulating the Confidential Information Before Transmission

Suppose that two communicating agents modulate the confidential information by some agreed upon scheme and then encrypt the modulated information so as to add another level of abstraction to the transmitted information. Proposition 4 shows 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 some kinds of encryption of the information do 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 observed information \( Q \), then even if a modulation of \( P \) by some relation \( M \), that is total and injective, is transmitted as \( Q \), we can still find an abstraction relation relating \( P \) and \( Q \) without knowing \( M \).

**Proposition 4.** 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 the observed information on a monitored communication channel. If the confidential information contained in \( P \) is being leaked as that represented by \( Q \) then the modulation using \( M \) of the confidential information contained in \( P \) is also being leaked as that represented by \( Q \) (i.e., \( \exists X \mid P \cdot X = Q \) \( \Rightarrow \exists Y \mid P \cdot M \cdot Y = Q \)).

**Proof.** The proof involves the properties of total and injective relations, as well as the application of Proposition 2, and basic properties of relations and residues. The detailed proof can be found in Jaskolka et al.\(^{11}\).

We require that the modulation relation be total and injective. To ensure that no information is lost during the modulation, the totality of the modulation relation ensures that, when composed with the relation containing the confidential information, 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 lead to a loss of information and can cause the tests to be averted. The reader can find more details on the aversion of the tests in Jaskolka et al.\(^{11}\).

5. A Prototype Tool for Automating the Proposed Detection Technique

We implemented a prototype tool, written in the functional programming language Haskell, to automate the process of detecting confidential information leakage via protocol-based covert channels. The prototype tool employs a layered architecture. Each layer of the tool specialises in a set of related activities. One of the main services provided by the prototype tool is covert channel analysis. The covert channel analysis service offers tools to detect confidential information leakage via protocol-based covert channels. These tools correspond to the corollaries of the proposed technique from Section 4 and can be used to verify whether confidential information has been leaked and to compute the corresponding abstraction relation if it is found that one does indeed exist. A brief example illustrating the usage of the prototype tool to automate the proposed technique is given in Section 5.1. For more discussion of the architecture design and more examples on the usage of the prototype tool we refer the reader to Part II of this series of papers\(^{12}\) and to Jaskolka et al.\(^{11}\).

5.1. An Illustrative Example of the Automation of the Proposed Detection Technique

Consider a system where two agents, \( A \) and \( B \), are communicating. Suppose that \( A \) is communicating from within an organisation and that \( B \) is communicating from outside the organisation. Suppose that the organisation has a security policy which defines its confidential information to be the sequence of the first ten digits of the number \( \pi \). This is to say that we have \( P = \{(1, 3), (2, 1), (3, 4), (4, 1), (5, 5), (6, 9), (7, 2), (8, 6), (9, 5), (10, 3)\} \) which is a representation of the sequence \(3, 1, 4, 1, 5, 9, 2, 6, 5, 3\).

Assume that \( A \) and \( B \) agree on a scheme for transmitting the confidential information. It is decided that \( A \) will exploit the Internet Protocol (IP), in particular, the IP Identification field in order to leak the confidential information to \( B \). The IP Identification field is used to uniquely identify an IP datagram within a flow of datagrams that share the same source and destination. Since the value for the IP Identification field should be chosen at random, it is possible
to choose a non-random value for the field without interrupting the IP mechanism\(^5\). Suppose that \(A\) uses the 16-bit IP Identification field to send, in a sequence of IP datagrams, the set of confidential information of its organisation. In order to attempt to mask the data being sent, \(A\) first encrypts the information before embedding it into the IP header. For this purpose, \(A\) uses a public key encryption technique to encrypt the information. The encryption generates the sequence\(^c\) \((12, 1, 16, 1, 17, 18, 11, 6, 17, 12)\) in place of the sequence \((3, 1, 4, 1, 5, 9, 2, 6, 5, 3)\).

Suppose that the organisation from which \(A\) is communicating installs a monitor configured to extract the IP Identification field from the transmitted IP datagrams associated with the communication between \(A\) and \(B\). Then, the information that the monitor records is the sequence \((12, 1, 16, 1, 17, 18, 11, 6, 17, 12)\). This observed information is stored in a file which, from this point forward, we call observed.seq. Also, assume that the monitor is already configured with the set of confidential information, which in this case is the sequence \((3, 1, 4, 1, 5, 9, 2, 6, 5, 3)\). This sequence is stored in a file which we call confidential.seq.

We wish to analyse the information that the monitor observed to have been sent on the communication channel to verify whether the confidential information has been leaked in some form. We start by loading the prototype tool into the prototype tool modules in the Glasgow Haskell Compiler’s interactive environment (ghci) and we create a new data store file called ExampleDB to store all of the relations for this session so that we can quickly recall them for later use. This is done with the following commands:

\[
\begin{align*}
&> \text{load PrototypeTool} \\
&> \text{new "ExampleDB"}
\end{align*}
\]

Next, we load the files into the prototype tool to construct the internal relational representation of the information by issuing the following commands:

\[
\begin{align*}
&> \text{loadRel "observed.seq" "observed" "ExampleDB"} \\
&> \text{observed <- select "observed" "ExampleDB"} \\
&> \text{loadRel "confidential.seq" "confidential" "ExampleDB"} \\
&> \text{confidential <- select "confidential" "ExampleDB"}
\end{align*}
\]

These commands construct the relational representation of the information contained in observed.seq and confidential.seq as the relations \{\((1, 12), (2, 1), (3, 16), (4, 1), (5, 17), (6, 18), (7, 11), (8, 6), (9, 17), (10, 12)\)\} and \{\((1, 3), (2, 1), (3, 4), (4, 1), (5, 5), (6, 9), (7, 2), (8, 6), (9, 5), (10, 3)\)\}, respectively, and stores them in the data store ExampleDB. We then select the relations from the data store, where they are represented as set-valued maps, to be used in the tests and computations.

Once the files have been loaded, we can perform the first test: verifying the existence of an abstraction relation. The test function corresponds to Corollary 1.

\[
\begin{align*}
&> \text{print (test confidential observed)} \\
&\text{True}
\end{align*}
\]

Here the result is True. This means that there exists an abstraction relation which relates the confidential information to the observed information meaning that the confidential information has been leaked in some form on the communication channel.

As we have verified that an abstraction relation does indeed exist, we can compute the abstraction relation relating the confidential information to the observed information. In the prototype tool, the universal relation \(L\) is represented as the restricted universal relation on \(P\) and \(Q\), denoted \(\mathbb{L}_{P,Q}\), where \(P\) and \(Q\) are relations and \(\mathbb{L}_{P,Q} \overset{\text{def}}{=} \{(x, y) \mid x \in \text{ran}(P) \land y \in \text{ran}(Q)\}\). This representation is required since, in the implementation of the prototype tool, we need to define the universal relation on a finite space. For the computation, the filter \(R\) from Corollary 2 is the universal relation restricted on the relations confidential and observed and is denoted by \(\text{top}\). The computation of the abstraction relation is done with the \(\text{compute}\) function which corresponds to Corollary 2.

\[
\begin{align*}
&> \text{printRel (compute confidential observed top)}
\end{align*}
\]

As a result, we find that the abstraction relation is given by \(X = \{(1, 1), (2, 11), (3, 12), (4, 16), (5, 17), (6, 6), (9, 18)\}\). As an example, the prototype tool outputs a relation which contains \"1\" \(\rightarrow\) \"1\" and \"2\" \(\rightarrow\) \"11\", showing that in the confidential information, a 1 was transmitted as 1 in the observed information and a 2 was transmitted as 11, respectively.

---

\(^c\) This sequence is generated using RSA encryption with \(p = 3, q = 7, N = 21, e = 5, d = 41\).
With this short illustrative example, we have shown the use of the prototype tool to verify the existence of and to compute the abstraction relation relating the confidential information to the information observed to be sent on a communication channel.

6. Discussion and Related Work

The proposed relation model offers simplicity when carrying out the computations required in detecting confidential information leakage via protocol-based covert channels. The stream representation allows us to take intervals of data from the channel and examine each interval, leading to computations of finite relations rather than infinite ones. Furthermore, relations are simple mathematical concepts that offer a certain level of abstraction in the model of covert channels which gives much more power and flexibility in the ability to model particular types of covert channels.

With the proposed technique, we are interested in uncovering a stream of data being leaked from a data store over some period of time. This means that we do not handle cases where, for example, a single key is sent in one data package. In this way, the proposed technique is similar to other techniques such as those which use machine learning techniques, such as neural networks (e.g., Tumovian & Anikeev\textsuperscript{13}), to analyse data streams for anomalies indicating the possible existence of a covert channel. Since machine learning techniques require training data, it is impossible to discover a covert channel using a single data package\textsuperscript{13}.

6.1. What the Proposed Technique Tells Us About Detecting Information Leakages

It is generally agreed that it is impossible to completely eliminate covert channels\textsuperscript{14}. Because of this, it is not surprising that the proposed technique points to ways in which covert channel users can avert the test for the existence of an abstraction relation so that users of protocol-based covert channels would be able to transmit confidential information without detection.

Consider a scenario where the confidential information is a sequence of four bits: \((0, 0, 1, 1)\). Suppose the encoding scheme is to complement every second bit, yielding \((0, 1, 1, 0)\). Therefore, the confidential information is represented by \(P = \{(1, 0), (2, 0), (3, 1), (4, 1)\}\) and the observed information is represented by \(Q = \{(1, 0), (2, 1), (3, 1), (4, 0)\}\). By applying the proposed technique, we can see that the resulting abstraction relation is \(X = \emptyset\) (i.e., \(X\) is the universal relation). Similarly, consider a scenario where we assume that there is a single confidential information \(p\). Also, suppose that the sequence of communication is \((q_1, q_2, q_3, q_4)\). Then, the confidential information is represented by \(P = \{(1, p), (2, p), (3, p), (4, p)\}\) and the observed information is represented by \(Q = \{(1, q_1), (2, q_2), (3, q_3), (4, q_4)\}\). Again, the proposed technique finds that the abstraction relation is \(X = \{(p, q_1), (p, q_2), (p, q_3), (p, q_4)\}\) or equivalently, \(X = \emptyset\) on the ranges of \(P\) and \(Q\). These simple examples illustrate that if the abstraction relation \(X = \emptyset\), we do not obtain any information that can aid in uncovering confidential information leakage, since we find that it is possible that the observed information could be any other information.

More precisely, the proposed technique reveals that we are unable to uncover the leakage of confidential information if it has been encoded using a non-injective abstraction relation. When the abstraction relation is non-injective, we cannot be sure which elements of the observed information correspond to the elements of the confidential information. Even if the abstraction relation is injective, it cannot be guaranteed that we will still be able to properly decode the observed information to determine if the confidential information has been leaked as it may be the case that we have a situation similar to the second example given above, where \(X\) is injective but we only have a single data packet which does not provide enough information to distinguish whether the confidential information has been leaked.

By revealing that we are unable to detect confidential information leakage via protocol-based covert channels when the abstraction relation is non-injective, the proposed technique formally confirms a selection of results which we informally hold to be true. For instance, the use of any sort of sophisticated encoding schemes, such as block-ciphers which might use cipher block chaining techniques, will lead to an encoding of the confidential information with a non-injective abstraction relation. Therefore, the proposed technique will be unable to uncover this leakage. Similarly, the use of a uniformly random one-time pad will allow for the test for the existence of an abstraction relation to be averted. This is not to say that we are unable to develop heuristics which can hint at whether there has been a leakage of confidential information via protocol-based covert channels, but formally we cannot uncover such a leakage using the proposed technique. However, it is possible for agents communicating confidential information via a protocol-
based covert channel to use an unsophisticated encoding scheme or possibly no encoding scheme at all (e.g., Smeets & Koo15). In such cases, the proposed technique offers a means for detecting confidential information leakages.

6.2. A Comment on the Feasibility

With regard to the feasibility of the proposed technique, concerns might be raised relating to the ability to handle the potentially enormous relations representing the confidential and observed information as well as the ability to record and store all of the information that is transmitted during a communication session. However, the technology required to manage these concerns does indeed exist. For instance, with regard to handling and managing the potentially enormous relations, one can use Binary Decision Diagrams (BDDs) and existing libraries such as the BuDDY BDD library, which can handle relations of sizes about $2^{32}$ where each node requires only about 20 bytes of memory15. With regard to the ability to record and store all of the information that is transmitted during a communication session, we find that technology exists which can aid in alleviating such concerns. A full packet capture of a saturated 1 Gbps link will yield approximately 6TB of data16. Such packet capture can be done using existing libraries such as LibPCAP which is used the common Wireshark packet capture application17. Other technologies, such as those provided by Emulex, including the EndaceProbe 7000 intelligent network recorder18, offer network monitoring and recording of communication links up to 10 Gbps with up to 64TB of storage. Such capabilities can be used to monitor and record the traffic existing on the whole network, however, the proposed technique only requires the monitoring and recording of communication on a single communication channel which will have significantly less flow than the entire network. Therefore, the issues concerning the ability to record and store all of the information that is transmitted during a communication session can be mitigated with existing technologies. Thus, the proposed technique is feasible, but due to resource limitations, we do not provide a full scale feasibility study in this paper.

6.3. A Comment on the Complexity

Let $n$ be the cardinality of a relation. Unary operators, such as the complement and converse, have an $O(n)$ computational complexity19. The computational complexity of binary operators is $O(n \log n)$ if each tuple of the first relation should be compared with each tuple in the second relation19. This includes the set operators, union and intersection, as well as the comparison operations, equality and containment. Relational composition has an $O(n^3)$ computational complexity20. Therefore, the computational complexity of the test described by Corollary 1 and computation described by Corollary 2 are $O(n^3)$. The $O(n^3)$ complexity arises from the need to compute the composition of relations in each of the tests and computations. As the complexity of relational composition is the most expensive operation, it dominates the computational complexity of the algorithms for verifying the existence of an abstraction relation and for computing the abstraction relation if it exists.

7. Conclusion and Future Work

We presented an algebraic technique using relations for detecting confidential information leakage via protocol-based covert channels which is part of the development of tools for investigative support for information confidentiality. The technique does not rely on heuristics to uncover the use of covert channels. It gives a more formal and rigorous approach and offers a degree of simplicity. The technique provides tests to verify the existence of an abstraction relation and computations to find the abstraction relation if it exists. These tests and computations are expandable allowing for the technique to handle complex scenarios which may involve modulating the confidential information, for example. We also provided a short illustrative example showing the usage of a prototype tool that automates the application of the proposed technique. The continued development of investigative support for information confidentiality with the application and further automation of the proposed technique is discussed in Part II of this series of papers12.

Further investigation into possible ways for improving the theoretical worst-case complexity for the tests and computations of the proposed technique is left as future work. Currently, 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 sample a portion of the communication stream which
was leaking confidential information. However, rather than the sample containing the confidential information in its entirety, it 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. Also, the proposed detection technique handles only protocol-based covert channels. We would like to extend the proposed technique in order to tackle the most general covert channel possible where communication consists of a combination of environmental and protocol-based knowledge. For example, in order to deal with channels based on the modulation of the timing of events, we can extend the proposed technique by allowing for projections on points in time to construct the relation representing the observed information based on the timing information. Finally, the continuing development of a more sophisticated and configurable automation system to handle large scale covert channels and to implement the capabilities of the communication monitors presented in Section 4.1 would be ideal.

Acknowledgements

This paper is a revised and extended version of the material presented in Jaskolka et al. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Grant Number RGPIN227806-09.

References

1. Scott, C.. Network covert channels: Review of current state and analysis of viability of the use of x.509 certificates for covert communica-
HICSS-44. Koloa, Kauai, HI, USA; 2011, p. 1–10.
yogy Directorate; Washington, DC, USA; 2009.
SP’03. 2003, p. 21–25.
0template1.php?601
12. Jaskolka, J., Khedri, R., Sabri, K.. Investigative support for information confidentiality part II: Applications in cryptanalysis and digital
forensics. Procedia Computer Science (FNC-2014 & MobiSPC-2014) 2014;.
Science and Technology; 2000.
20. Wallace, M., Kollias, S.. Two algorithms for fast incremental transitive closure of sparse fuzzy binary relations. International Journal of