ELSEVIER

CrossMark

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 266 – 275

Procedia
Computer Science

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

Investigative Support for Information Confidentiality Part II: Applications in Cryptanalysis and Digital Forensics

Jason Jaskolka[a,*], Ridha Khedri[a], Khair Eddin Sabri[b]

[a] *Department of Computing and Software, Faculty of Engineering, McMaster University, Hamilton, Ontario, Canada*
[b] *Department of Computer Science, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*

Abstract

This is Part II in a two-part series discussing the development of investigative support for information confidentiality. In Part I, we proposed a technique based on relation algebra to detect confidential information leakage via protocol-based covert channels. In this paper, we continue developing investigative support for information confidentiality. We examine the application of the technique for detecting confidential information leakage proposed in Part I in cryptanalysis and digital forensics to highlight its usefulness beyond the scope of covert channel analysis. By way of a short case study, we show the automation of the cryptanalysis application of the technique for detecting confidential information leakage using a prototype tool and a known-plaintext attack.

Keywords: cryptanalysis, confidentiality, digital forensics, formal methods, covert channels, security

1. Introduction and Motivation

In information security, the protection of information from unauthorised disclosure is known as confidentiality. Today, many organisations take significant measures in an effort to maintain the confidentiality of their information. One way to achieve confidentiality of information is by employing cryptographic techniques (e.g., Schneier[1]). Cryptographic techniques provide a means for encoding information using an enciphering algorithm and a cryptographic key for transforming the information that is to be protected, called the *plaintext*, to a form that is unreadable by any party that does not know the cryptographic key, called the *ciphertext*. In this way, it is required that anyone wishing to read the plaintext from the ciphertext must know the cryptographic key. However, with the use of cryptanalysis techniques, it is possible to uncover the plaintext from the ciphertext, even if the cryptographic key is unknown.

In this paper, we investigate the application of the algebraic technique for detecting confidential information leakage described in Part I of this series of papers[2] and in Jaskolka et al.[3] as a cryptanalysis technique. The proposed cryptanalysis technique uses the relational representation of information and the tests and computations provided by the technique for detecting confidential information leakage along with the idea of a known-plaintext attack (e.g.,

[*] Corresponding author. Tel.: +1-905-525-9140 ; fax: +1-905-524-0340.
E-mail address: jaskolj@mcmaster.ca (Jason Jaskolka).

doi:10.1016/j.procs.2014.07.022

Bishop[4]) which is an attack model where the attacker has samples of both the plaintext and its ciphertext. An application of the technique for detecting confidential information leakage in the area of cryptanalysis highlights its usefulness beyond the scope of covert channel analysis which was detailed in Part I of this series of papers[2].

The cryptanalysis technique proposed in this paper, in conjunction with the theoretical foundations of the relation algebraic technique for detecting confidential information leakage presented in Part I of this series of papers[2], are steps towards the generation of investigative support for information confidentiality. We aim to provide tools and mechanisms for maintaining the confidentiality of information. In particular, we look to develop formal methods for aiding in digital forensic investigations of violations of confidentiality as the annual costs associated with such violations have been estimated to average approximately \$11.6 million[5]. As more and more confidential information needs to be protected, the need for this kind of support is growing exponentially.

In Section 2, we present the necessary mathematical background of relations. In Section 3, we recall the highlights of the technique for detecting confidential information leakage described in Part I of this series of papers[2]. In Section 4, we examine the application of the technique for detecting confidential information leakage in the area of cryptanalysis. In Section 5, we discuss the automation of the proposed cryptanalysis technique using a prototype tool implemented in the *Haskell* functional programming language. In Section 6, we demonstrate the use of the prototype tool for performing a cryptanalysis on a simple case study. In Section 7, we provide a brief discussion of the application of the technique for detecting confidential information leakage in cryptanalysis and digital forensics. Lastly, in Section 8 provide some concluding remarks and discuss future work.

2. Mathematical Background

For two sets X and Y, a *relation* R on $X \times Y$ is a subset of the Cartesian product $X \times Y$. In the remainder of this paper, the *identity relation* is denoted by $\mathbb{I}$, the *universal relation* is denoted by $\mathbb{L}$ and, the *empty relation* is denoted by $\emptyset$. Additionally, we have three important operations on relations that are needed for the rest of the paper: composition, converse, and complement.

Definition 1 (e.g., Schmidt & Ströhlein[6]). Let $P \subseteq X \times Y$ and $Q \subseteq Y \times Z$ be relations.

1. The *relational composition* of P and Q is given as $P ; Q \overset{\text{def}}{=} \{(x,z) \mid \exists(y \mid y \in Y : (x,y) \in P \land (y,z) \in Q\,)\}$.
2. The *converse* of P is given as $P^{\smile} \overset{\text{def}}{=} \{(x,y) \mid (y,x) \in P\}$.
3. The *complement* of P is given as $\overline{P} \overset{\text{def}}{=} \{(x,y) \mid (x,y) \notin P\}$.

We also have a special operation on relations called the residue. Residues help solve equations of the form $P ; X = Q$ or $X ; P = Q$.

Definition 2 (Schmidt & Ströhlein[6]). Let P and Q be two relations.

1. $Q/P \overset{\text{def}}{=} \overline{\overline{Q} ; P^{\smile}}$ is the *left residue* of Q by P
2. $P\backslash Q \overset{\text{def}}{=} \overline{P^{\smile} ; \overline{Q}}$ is the *right residue* of Q by P

The left residue is the greatest relation X such that $X ; P \subseteq Q$ and the right residue is the the greatest relation X such that $P ; X \subseteq Q$. This result is summarised in Proposition 1.

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

1. $X ; P \subseteq Q \iff X \subseteq Q/P$
2. $P ; X \subseteq Q \iff X \subseteq P\backslash Q$

Proof. The proof can be found in Schmidt & Ströhlein[6]. □

Sometimes we require that a relation be a left residue and right residue simultaneously. This notion is called the symmetric quotient.

Definition 3 (e.g., Schmidt & Ströhlein[6]). Let P and Q be two relations. Then, $\mathrm{syq}(P,Q) \overset{\text{def}}{=} \overline{P^{\smile} ; \overline{Q}} \cap \overline{\overline{P^{\smile}} ; Q} = (P\backslash Q) \cap (P^{\smile}/Q^{\smile})$ is the *symmetric quotient* of P and Q.

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

3. A Technique for Detecting Confidential Information Leakage

In this section, we summarise the technique for detecting confidential information leakage on monitored communication channels. The details of the technique can be found in Part I of this series of papers[2].

3.1. A Relational Representation

Jaskolka & Khedri[7] discussed how we can view information sent over communication channels as being encapsulated in a data structure of some dimension. This data structure has fields in which the information is embedded. Because of this, in the technique for detecting confidential information leakage, we represent the information sent on a communication 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. 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 $\mathbb{N} \times \mathbb{D}$ and therefore it is a relation.

In order to uncover an information leakage, it is sufficient to find an abstraction relation between the confidential information and the stream of information observed to be have been sent on a communication channel. In Figure 1, P represents the confidential information which should not be sent on the channel, Q represents the information observed by a monitor watching the information sent on the communication channel, and X represents an abstraction relation relating P and Q. An abstraction relation X requires that the diagram given in Figure 1 commutes.

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

Fig. 1: A representation of the relationship between the relations P and Q via the abstraction relation X

We consider an information leakage to be detected if and only if there exists an abstraction relation, different from the empty relation $\emptyset$ and the universal relation $\mathbb{L}$, relating the confidential information and information observed to be have been sent on a communication channel.

3.2. The Technique

The technique for detecting confidential information leakage has two components: monitoring the information sent on the communication channel and finding and computing an abstraction relation relating the confidential information to the observed information.

A monitor that is pre-equipped with the relation corresponding to the set of confidential information P, defined by the system security policy, records the communication among agents when a communication channel is established. The monitor watches the stream of packets being transmitted from authorised agents to unauthorised agents and constructs the relational stream of information being sent Q. After observing the information that has been sent, the monitor needs to determine whether the confidential information has been leaked in any form by verifying the existence of an abstraction relation X relating P and Q.

Since the monitor knows the relation corresponding to the confidential information P and the relation corresponding to the observed information Q, the existence of an abstraction relation X relating 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 can be found in Jaskolka et al.[8]. □

Proposition 2 is a test to verify if there is an abstraction between the observed information and the confidential information. This test is directly related to commutativity of the diagram in Figure 1. If the test holds, 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 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. The proof can be found in Jaskolka et al.[8]. □

In Corollary 1, we verify the way(s) in which the diagram in Figure 1 commutes. Each term of the disjunction in the test corresponds to one of the ways in which Figure 1 can commute. As long as we can satisfy at least one of the ways in which the diagram commutes, we can find an abstraction relation X relating the confidential information P to the observed information Q.

We can compute the abstraction relation X with Proposition 3.

Proposition 3. *Let P and Q 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 can be found in Jaskolka et al.[8]. □

The relation R plays the role of a filter and 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. 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. Corollary 2 gives three cases for which we can compute the abstraction relation X.

Corollary 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 X such that*

1. *$X = R \cap (Q \backslash P)^{\smile}$ if and only if $P \subseteq Q ; (R^{\smile} \cap (Q \backslash P))$*
2. *$X = R \cap (P \backslash Q)$ if and only if $Q \subseteq P ; (R \cap (P \backslash Q))$*
3. *$X = R \cap \mathrm{syq}(P, Q)$ if and only if $P \subseteq Q ; (R^{\smile} \cap (Q \backslash P)) \ \wedge \ Q \subseteq P ; (R \cap (P \backslash Q))$*

Proof. The proof can be found in Jaskolka et al.[8]. □

The above tests and computations represented by Proposition 2 and Proposition 3, and consequently Corollary 1 and Corollary 2, provide the core for determining the existence of an abstraction relation between the confidential information and the information observed to have been sent on the communication channel. As shown in Part I of this series of papers[2], equipped with this technique, it is possible to detect the leakage of confidential information via monitored protocol-based covert channels in a digital forensics context (i.e., investigation after the information has already been sent).

4. A Cryptanalysis Technique

Consider a scenario where an encrypted communication between two agents is intercepted. Suppose that while attempting to decipher the encrypted message, the analyst has an intuition that the message contains some important pieces of information (i.e., a date, a location, a name, etc.). Equipped with the technique for detecting confidential information leakage, the analyst can perform an investigation into the observed (intercepted) information transmitted between the agents. For instance, the analyst can run the test for an abstraction relation (Corollary 1) between the observed information and the information for which he/she suspects may have been sent. For example, the analyst may suspect that the transmission contains the suspected location of where an important meeting has taken place. If we allow these assumptions to form a confidential information, then the technique for detecting confidential information leakage can be used to search for an abstraction relation relating the suspected plaintext to the intercepted ciphertext message. If an abstraction relation can be found relating some plaintext to the intercepted ciphertext message, the analyst can begin to develop the cryptographic key that may have been used. This procedure is very much like those developed by Alan Turing at Bletchley Park during World War II when deciphering the codes of the Enigma machine, which use known-plaintext attacks to analyse ciphertexts[9].

Consider the scenario where an analyst needs to decrypt a ciphertext, N characters in length. In performing the cryptanalysis, the analyst must begin by enumerating all of the ciphertext characters (if necessary). We denote the set of all ciphertext characters as $\mathbb{D}_C$ and the set of all plaintext characters as $\mathbb{D}_P$. Assume that the analyst models the position of each character in the ciphertext by $\mathbb{N}$. Then, the analyst is able to construct the relational representation of the ciphertext as a relation $C \subseteq \mathbb{N} \times \mathbb{D}_C$. Next, the analyst guesses a fragment of plaintext which is suspected to occur in the ciphertext. We denote the plaintext fragment as $P = p_1 p_2 \dots p_n$ where $\forall(i \mid 1 \leq i \leq n \leq N : p_i \in \mathbb{D}_P)$. The length of the plaintext fragment is denoted by n. The idea is to check if there exists an abstraction relation between the plaintext fragment and any ciphertext fragment of length n for all positions in the ciphertext. This relationship is shown in Figure 2. In order to test for an abstraction relation, the plaintext fragment needs to be represented as a set of relations $P_i \subseteq \mathbb{N} \times \mathbb{D}_P$ such that $\forall(i \mid 1 \leq i \leq N - n : P_i = \{(i, p_1), (i+1, p_2), \dots, (i+n-1, p_n)\})$. Similarly, we construct the corresponding fragments of the ciphertext as relations $C_i \subseteq \mathbb{N} \times \mathbb{D}_Q$ such that $\forall(i \mid 1 \leq i \leq N - n : Q_i = \{(i, c_i), (i+1, c_{i+1}), \dots, (i+n-1, c_{i+n-1})\})$.

$$\begin{tikzcd}
 & \mathbb{D}_C \\
\mathbb{N} \arrow[ur, "C_i"] \arrow[r, "P_i"'] & \mathbb{D}_P \arrow[u, "K_i"']
\end{tikzcd}$$

Fig. 2: A representation of the relationship between a plaintext fragment P_i and ciphertext fragment C_i via the abstraction relation K_i

The analyst runs the test given in Corollary 1 for each pair of corresponding plaintext, ciphertext pairs (i.e., $P_i = C_i ; (C_i \backslash P_i) \;\vee\; C_i = P_i ; (P_i \backslash C_i)$ for $1 \leq i \leq N - n$). Each positive test result indicates that there exists an abstraction relation between the guessed plaintext fragment and the corresponding ciphertext fragment. This indicates the presence of a possible cryptographic key fragment K_i. The analyst can compute each of the possible cryptographic key fragments by applying Corollary 2 for each (P_i, Q_i) pair yielding a positive test result. With the possible cryptographic key fragments, the analyst can apply the key fragment to the ciphertext, which can reveal part of the plaintext. With the additional information provided by applying the possible cryptographic key fragments, the analyst can generate a new, more refined guess at a plaintext fragment suspected of occurring in the ciphertext and repeat the process. The analyst may be able to infer a larger fragment of the message with some intuition of the neighbouring characters in the message. The process is very likely to converge on the complete cryptographic key thus decrypting the given ciphertext message.

5. Automated Cryptanalysis using a Prototype Tool

In Part I of this series of papers[2], we reported on a prototype tool, written in the functional programming language *Haskell*, that allows for the automation of the tests and computations presented in technique for detecting confidential information leakage. In addition to the *covert channel analysis* service, the prototype tool also provides a *cryptanalysis* service. The cryptanalysis service offers the ability to perform a cryptanalysis based on the technique for detecting confidential information leakage and a known-plaintext attack as described in Section 4. We demonstrate the use of the prototype tool for performing a cryptanalysis on a simple case study in Section 6. For more details regarding the usage of the tool for cryptanalysis and for the complete analysis of a larger example using the Zodiac 408 cipher[10], we refer the reader to Jaskolka et al.[8].

6. A Case Study of the Automation of the Proposed Cryptanalysis Technique

Using the prototype tool described in Jaskolka et al.[8], we have automated the process of applying the proposed cryptanalysis technique. In Deavours & Kruh[11], we find a ciphertext originally encrypted using a German Army Enigma machine. This message has since been decrypted and roughly translated into English in Weierud[12]. For illustrative purposes in this case study, we use a portion of the English translation of this message. Since we are dealing with a message that was sent by an army, the need for confidentiality cannot be stressed enough as the unauthorised disclosure of the plaintext of this message could have devastating consequences.

For the purpose of this case study, we have encrypted the message using a substitution cipher. Table 1 shows the ciphertext in its entirety. We provide the highlights of the tool usage for applying the cryptanalysis technique and show how the technique for detecting confidential information leakage, in conjunction with a known-plaintext attack, can break the cipher to uncover the message.

Table 1: Case study ciphertext message

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	ꓘ	O	I	⊖	Z	Ǝ	G	F	ꟼ	ᗡ	J	J	\	E	O	Ǝ	⊖	⅃	L	L	G
B	Ǝ	/	⊡	ᗡ	Λ	H	⊖	E	A	ꟼ	F	●	N	Я	O	Q	ᗡ	Я	L	△	Q
C	P	Ǝ	⅃	I	U	T	O	F	ꟻ	B	H	⊖	ᗡ	Y	R	⊖	Y	Λ	◪	ꓘ	/
D	W	◪	□	▲	I	L	⊖	ꓘ	K	Ω	ᗡ	Ω	W	D	H	I	⊖	T	F	N	J
E	ᗡ	\	L	ꓘ	Q	U	Ǝ	G	●	△	Ɪ	D	◬	Ω	Y	F	L	V	E	⊖	ꟼ
F	■	⌖	G	I	⅃	■	B	Ǝ	ᗡ	⊡	I	F	W	Ɔ	+	D	⅃	R	G	P	Φ
G	K	I	O	Y	Ω	E	Я	P	Ǝ	▲	■	■	□	F	Y	⊼	W	\	P	X	⊥
H	Q	ᗡ	\	Ǝ	N	F	Ǝ	ᗡ	Ω	Ω	G	Φ	⌖	ꟼ	Я	F	S	O	ꟻ	●	⊥
I	X	T	⊼	K	Ω	Y	◬	L	V	ꟼ	△	Ω	V	Y	W	ꟻ	A	ꓘ	L	⊖	H
J	⊖	ꟼ	M	T	O	ᗡ	Y	Я	T	Q	I	⊖	U	⊡	ꟻ	Y	H	□	P	D	⅃
K	Ǝ	Ǝ	X	\	ꟻ	G	Φ	Ǝ	W	A	ꓘ	L	M	T	Я	⌖	⊙	⊥	⊡	P	N
L	Ω	⊼	⊖	▲	F	P	K	+	H	M	G	I	ꓘ	⅃	B	ᗡ	D	ꟼ	⊖	S	Ɔ
M	Z	I	M	ꟼ	⊥	U	R	M	I	L	X	S	Y	L	M	T	Я	P	◬	ꟼ	H
N	⊖	E	Q	T	⊥	●	P	J	△	Ǝ	▲	L	U	T	Λ	F	L	ᗡ	V	E	S
O	V	G	O	⌖	T	Λ	Z	⌖	ꓘ	D	A	⊖	T	◪	+	T	⊥	⊼	⅃	\	L

As preparation for the analysis, we first need to enumerate each of the cipher characters so that we are able to represent them for use with the prototype tool. The enumeration is given in Table 2.

We start by loading the prototype tool modules in the Glasgow Haskell Compiler's interactive environment (`ghci`) as follows:

```
> :load PrototypeTool
```

We store the enumerated representation in a file for use with the prototype tool. From this point forward, we call the file containing the enumerated ciphertext `cipher.rel`. We construct the relational representation of the information contained in the `cipher.rel` file and store it in the newly created data store, `CryptanalysisDB`, by issuing the following commands:

Table 2: Case study ciphertext character enumeration

1	△	8	ꓘ	15	ꟻ	22	Ǝ	29	K	36	Ꞇ	43	⊥	50	A
2	◩	9	O	16	L	23	G	30	Ɪ	37	M	44	N	51	E
3	P	10	R	17	\	24	Y	31	Ω	38	J	45	Q	52	Я
4	/	11	⊼	18	Ɔ	25	F	32	■	39	Λ	46	D	53	⌖
5	Z	12	ᑫ	19	W	26	⊙	33	┘	40	I	47	●	54	□
6	U	13	X	20	V	27	H	34	ᗡ	41	◬	48	⊖		
7	B	14	Φ	21	+	28	⊡	35	▲	42	T	49	S		

```
> new "CryptanalysisDB"
> loadRel "cipher.rel" "cipher" "CryptanalysisDB"
> cipher <- select "cipher" "CryptanalysisDB"
```

The idea is to guess a known word or phrase that is likely to appear in the plaintext message corresponding to the given ciphertext. It is assumed that we know the context of the message. Therefore, we know that the cipher was written during a time of battle and it might be suspected that the author may have been conveying orders to a brigade of commanders and troops. So, we might expect to find words such as "orders", "troops", or "forces" in some reference to the conveyance of orders. These would offer formidable starting points for the analysis. However, for simplicity, brevity, and illustrative purposes, suppose that we have obtained a tip by some means (it is not important how) that the message may refer to an attack on a fortification which must not fall to the enemy. Therefore, we might guess that the plaintext message contains the phrase "FORTIFICATIONS MUST BE HELD". Using the prototype tool, we generate a relation based on this phrase. In our representation of the phrase, we use only uppercase letters and ignore spaces as we are simply trying to find a readable plaintext based on the assumption that spaces are not encoded. We generate a relation based on this phrase by issuing the following command with the prototype tool:

```
> let phrase = relFromString "FORTIFICATIONSMUSTBEHELD"
```

Next, we apply the cryptanalysis procedure based on the proposed technique described in Section 4. We use the following command:

```
> cryptanalysis phrase cipher
```

As a result of applying the cryptanalysis technique, we find that we have two fragmented possibilities for the cryptographic key. After examining the two possible plaintexts (which are generated by the prototype tool, but not shown here due to space limiations), we see that there is only one plaintext which appears to make any sense. Based on the sensible plaintext, we find that our phrase is succeeded by "A _ _ L _ C O _ _ S" which might suggest that the our guessed phrase is followed by the phrase "AT ALL COSTS". We can concatenate our original phrase and this new phrase to have a much more refined phrase. After this refinement, we apply the cryptanalysis with the phrase "FORTIFICATIONS MUST BE HELD AT ALL COSTS". The process of performing the cryptanalysis and its output are given below.

```
> let phrase' = relFromString "FORTIFICATIONSMUSTBEHELDATALLCOSTS"
> cryptanalysis phrase' cipher

The 1 fragmented key possibility is:
{
  "A" |-> ["23","33"]
  "B" |-> ["20"]
  "C" |-> ["22"]
  "D" |-> ["53"]
  "E" |-> ["12","51"]
  "F" |-> ["38","45"]
  "H" |-> ["48"]
  "I" |-> ["1","6","8"]
  "L" |-> ["32","7"]
  "M" |-> ["31"]
  "N" |-> ["46"]
  "O" |-> ["30","34"]
  "R" |-> ["17"]
  "S" |-> ["25","28","41"]
  "T" |-> ["16","40","47"]
  "U" |-> ["24"]
}

The 1 possible plaintext is:
I _ T H _ C A S E O F F R E _ C H
A T T A C _ S O _ _ H E _ E S T _
_ _ F O _ T I F _ C A T I _ _ S _
L _ H O U _ H U _ _ I _ _ _ _ _ T
T H I _ M O M _ N _ T H _ S _ F O
R T I F I C A T I O N S M U S T B
E H E L D A T A L L C O S T S _ _
_ N A _ A _ _ _ T _ U M E _ _ C _
L L _ S U _ _ R _ _ _ F O R C _ S
C O M M A _ D E _ S _ _ _ T _ _ _
_ _ M U S T B E I M B U _ _ _ I T
H _ H E _ _ _ O U _ _ F T H I S _
U _ _ _ N A C C _ R _ A _ C _ _ I
T _ _ _ D _ _ S _ _ M _ H _ S _ _
_ _ _ A T I A L O N E H _ _ _ T _
E _ I _ _ T T _ _ U T _ _ _ _ S E
_ H E F _ _ T _ F I C _ T I _ _ S
T O B E _ B A _ D _ _ _ D I N _ H
_ _ _ _ _ _ A R T
```

As a result of applying the cryptanalysis, we find that we are left with only one fragmented cryptographic key. As an example, the prototype tool outputs a relation which contains `"T" |-> ["16","40","47"]`, showing that the letter 'T' corresponds to ciphertext characters enumerated as 16, 40, and 47 (in particular "L", "I", and "•"). One can easily fill in many of the blanks in the possible plaintext to reconstruct the original message, which is given in Table 3.

We have demonstrated the application and automation of the technique for detecting confidential information leakage in the context of cryptanalysis. This illustrates the usefulness of the technique beyond the scope of covert channel analysis and detecting confidential information leakages. Using the short illustrative example above, we have shown that in a cryptanalytic investigation where we may be able to perform a known-plaintext attack, we are able to uncover the encrypted message using the proposed cryptanalysis technique.

7. Discussion

Successful cryptanalysis has helped influence history as far back as the 1500s[13]. The benefits of leveraging cryptanalysis for diplomatic, military, and commercial purposes is recognised by many governments to this day. While it can be perceived that cryptanalysis malevolently poses a threat to the confidentiality of information since it provides

Table 3: Case study plaintext message

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
A	I	N	T	H	E	C	A	S	E	O	F	F	R	E	N	C	H	A	T	T	A
B	C	K	S	O	N	T	H	E	W	E	S	T	E	R	N	F	O	R	T	I	F
C	I	C	A	T	I	O	N	S	A	L	T	H	O	U	G	H	U	N	L	I	K
D	E	L	Y	A	T	T	H	I	S	M	O	M	E	N	T	T	H	O	S	E	F
E	O	R	T	I	F	I	C	A	T	I	O	N	S	M	U	S	T	B	E	H	E
F	L	D	A	T	A	L	L	C	O	S	T	S	E	V	E	N	A	G	A	I	N
G	S	T	N	U	M	E	R	I	C	A	L	L	Y	S	U	P	E	R	I	O	R
H	F	O	R	C	E	S	C	O	M	M	A	N	D	E	R	S	A	N	D	T	R
I	O	O	P	S	M	U	S	T	B	E	I	M	B	U	E	D	W	I	T	H	T
J	H	E	H	O	N	O	U	R	O	F	T	H	I	S	D	U	T	Y	I	N	A
K	C	C	O	R	D	A	N	C	E	W	I	T	H	O	R	D	E	R	S	I	E
L	M	P	H	A	S	I	S	E	T	H	A	T	I	A	L	O	N	E	H	A	V
M	E	T	H	E	R	I	G	H	T	T	O	A	U	T	H	O	R	I	S	E	T
N	H	E	F	O	R	T	I	F	I	C	A	T	I	O	N	S	T	O	B	E	A
O	B	A	N	D	O	N	E	D	I	N	W	H	O	L	E	O	R	P	A	R	T

a means for subverting cryptographic techniques that are in place to maintain confidentiality, we argue that it can also have a benevolence when we consider that the information that is being protected by the cryptographic techniques may be the secrets of a large organisation that are being smuggled to an outsider. In this way, the cryptanalysis technique presented in this paper can be seen as an extension to the technique for detecting confidential information leakage proposed in Part I of this series of papers[2].

With the ongoing threat of insiders with malicious interest to cause harm or 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. As an example, we can examine the June 2010 investigation in the United States of America with regards to a Russian spy ring, known as *The Illegals Program*. The spies were allegedly using various forms of stealthy communications including steganography, cryptography, and even Morse Code-like radio signals[14]. This example stresses the real threat of the leakage of information on confidentiality and security. The implications of confidential information leakage on the scale of the international espionage highlights the importance of developing techniques for supporting the preservation of confidentiality in computer systems.

In this paper, we are developing investigative support for information confidentiality. This involves looking at an analysis from a digital forensics perspective. By its very nature, digital forensics is analysis after-the-fact[15]. 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, despite that it seems the policy is being respected. For example, in the course of a forensic investigation of a confidential information leakage from an organisation, an analyst may need to ascertain whether a suspect made a call to a particular person with their mobile phone which has an encrypted SIM card. In such a case, assuming that the mobile phone and SIM card have been seized, the analyst can apply the proposed cryptanalysis technique to detect whether the name of the particular person exists in some form in the call history on the SIM card to aid determining the innocence or guilt of the suspect.

The use of techniques for detecting confidential information leakage in a digital forensics context is a new concept for generating investigative support for information confidentiality. Since we are dealing with analysis after-the-fact, performance is not a major consideration when developing detection mechanisms and cryptanalysis techniques. We simply need the analysis to be done in a reasonable amount of time. The importance and necessity for this type of application seems to be growing day by day as the need for protecting ever-growing amounts of private and sensitive information continues to increase.

8. Conclusion and Future Work

We presented an application of the technique for detecting confidential information leakage proposed in Part I of this series of papers[2] in the area of cryptanalysis and digital forensics which is part of the development of tools for investigative support for information confidentiality. The presented cryptanalysis technique uses the relational representation of information and the tests to verify the existence of an abstraction relation and computations to find the abstraction relation if it exists from the technique for detecting confidential information leakage in conjunction with notions from known-plaintext attacks to aid in uncovering plaintext messages from a given ciphertext. We also presented the automation of the proposed cryptanalysis technique using a short case study and our prototype tool.

The application of the technique for detecting confidential information leakage in the area of cryptanalysis and digital forensics reveals that the technique has the potential to be useful beyond detecting confidential information leakage and in areas other than covert channel analysis. More research into the applicability of the technique in other application domains, particularly, those concerning the confidentiality of information, is needed. This research would involve more exploration into developing alternative tools and mechanisms for investigative support for information confidentiality. Furthermore, continual progress in the development of a more sophisticated and configurable automation system to handle large scale examples for both covert channel analysis and cryptanalysis applications is needed.

Acknowledgements

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Grant Number RGPIN227806-09.

References

1. Schneier, B.. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* Wiley; second ed.; 1996.
2. Jaskolka, J., Khedri, R., Sabri, K.. Investigative support for information confidentiality part I: Detecting confidential information leakage via protocol-based covert channels. *Procedia Computer Science (FNC-2014 & MobiSPC-2014)* 2014;.
3. Jaskolka, J., Khedri, R., Sabri, K.. A formal test for detecting information leakage via covert channels. In: *Proceedings of the 7th Annual Cyber Security and Information Intelligence Research Workshop*; CSIIRW7. Oak Ridge, TN, USA; 2011, p. 1–4.
4. Bishop, M.. *Computer Security: Art and Science.* Boston, MA: Addison Wesley; 2002.
5. Ponemon Institute, . 2013 cost of cyber crime study: United states. Tech. Rep.; 2013.
6. Schmidt, G., Ströhlein, T.. *Relations and Graphs: Discrete Mathematics for Computer Science.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag; 1993.
7. Jaskolka, J., Khedri, R.. Exploring covert channels. In: *Proceedings of the 44th Hawaii International Conference on System Sciences*; HICSS-44. Koloa, Kauai, HI, USA; 2011, p. 1–10.
8. Jaskolka, J., Khedri, R., Sabri, K.. Information leakage via protocol-based covert channels: Detection, automation, and applications. Tech. Rep. CAS-11-05-RK; McMaster University; Hamilton, Ontario, Canada; 2011. Available: `http://www.cas.mcmaster.ca/cas/0template1.php?601`.
9. Turing, A.. *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life, Plus the Secrets of Enigma.* Oxford University Press; 2004.
10. Zodiologists, . Analysis of the zodiac killer's three part cipher (Z 408). Available: `http://www.zodiologists.com/z408_cipher_analysis.html` (Accessed: May 23, 2014); 2009.
11. Deavours, C., Kruh, L.. The turing bombe: Was it enough? *Cryptologia* 1990;**14**(4):331 – 349.
12. Weierud, F.. The ENIGMA message. Spooks Newsletter. Fourth Edition of the N&O column. Available: `http://www.cvni.net/radio/nsnl/nsnl004/nsnl4msg.html` (Accessed: May 23, 2014); 1998.
13. Samuel, D.. Code breaking in law enforcement: A 400-year history. *Forensic Science Communications* 2006;**8**(2).
14. Williams, C.. Russian spy ring bust uncovers tech toolkit. The Register; 2010.
15. Srinivasan, S.. Security and privacy in the computer forensics context. In: *Proceedings of the 2006 International Conference on Communication Technology*. Piscataway, NJ, USA: IEEE Computer Society; 2006, p. 1–3.