Towards Discovering Criminal Communities from Textual Data

Rabeah Al-Zaidy Benjamin C. M. Fung Amr M. Youssef
Concordia Institute for Information Systems Engineering
Concordia University
Montreal, QC, Canada, H3G 1M8
{r_alzaid, fung, youssef}@ciise.concordia.ca

ABSTRACT

In many criminal cases, forensically collected data contain valuable information about a suspect's social networks. An investigator often has to manually extract information from the collected text documents and enter it into a police database for further investigation with criminal network analysis tools. In this paper, we propose a method to discover criminal communities, to analyze the closeness of the members in the communities, and to extract useful information for crime investigation directly from the text documents. The proposed method, together with the implemented software tool, has received positive feedbacks from the digital forensics team of a law enforcement unit in Canada.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.7.5 [**Document and Text Processing**]: Document Capture—*document analysis*; K.4.2 [**Computers and Society**]: Social Issues—*abuse and crime involving computers*

Keywords

Forensic analysis, data mining, community discovery, crime investigation

1. INTRODUCTION

The advancement of hardware technology has significantly reduced the cost of large-size storage devices; average computers users, including criminals, can now afford these devices to conduct their daily activities. After a suspect is arrested, his/her computer(s) are often seized for further investigation. With the current large sizes of storage media, analysis and examination of forensically collected data is a labour intensive task. In the United States, the FBI's Regional Computer Forensics Laboratories (RCFL) conducted over 6,000 examinations on behalf of 689 law enforcement agencies in one year. The amount of data they examined reached up to 2,334 Tera Bytes (TB) in 2009, double the size processed in 2007 [9]. Although performance has greatly improved over the years, there is a need for more resources and new software tools to aid forensics examiners to process collected data.

Inspection of files involves searching content for information that can be used as evidence or that can lead to other sources of information that may assist the investigation process and analysis of the retrieved information. It is typically up to the investigator on what and how to search for evidence, depending on the case. Our contribution in this paper is to bridge the gap between criminal network mining and unstructured text data. Specifically, we study the problem of mining criminal communities from a set of text files collected from a suspect's hard drive. "Text files" can be any text documents, such as e-mails, chat logs, blogs, webpages, or any textual data.

We derived the following notion of "frequent community" after extensive discussions with the forensics team of a law enforcement unit in Canada: If two or more names occur together frequently in the data set, this indicates they have a strong relationship; therefore, they are considered to be a *frequent community*. In many cases, an investigator may have very few clues on suspects or an organization at the early stage of investigation. In some other cases, an investigator may already know the members involved in an organization, but does not have concrete information on the relationships between them or how the organization operates as a whole. Thus, in addition to the discovery of frequent communities, we measure the closeness among the members in a community and extract useful information from the community for crime investigation.

Many methods and tools have emerged to assist investigators in data analysis for crime investigation. Some social network analysis tools can effectively discover criminal communities from a well-structured database. For example, Yang and Ng [11] present a method to extract criminal networks from web sites that provide blogging services by using a topic-specific exploration mechanism. In their approach, they identify the actors in the network by using web crawlers that search for blog subscribers who participated in a discussion related to some specific criminal topics. After the network is constructed, they use text classification techniques to analyze the content of the documents. Chen et al. [2] demonstrates a successful application of data mining techniques to extract criminal relations from a large volume of a police department's incident summaries. They use the co-occurrence frequency to determine the weight of re-

SAC'11 March 21-25, 2011, TaiChung, Taiwan.

lationships between *pairs* of criminals. However, in most forensically seized hard drives and storage media, the information of criminal communities is not stored in the form of a structured database but in text documents scattered across the hard drive. Our study focuses on unstructured textual data obtained from a suspect's hard drive, not from a well-structured police database. Furthermore, our method can discover criminal communities of any size, i.e., it is not limited to pairs of criminals.

A criminal network follows a social network paradigm [8]. Thus, the approaches used for social network analysis can be adopted in the case of criminal networks. Many studies have introduced various approaches to construct a social network from text documents. [6] propose a framework to extract social networks from text documents that are available on the web. [7] propose a method to rank companies based on the social networks extracted from web pages. These approaches rely mainly on web mining techniques to search for the actors in the social networks from web documents. Another direction of social network studies targets some specific type of text documents such as emails. [12] propose a probabilistic approach that not only identifies communities in email messages but also extracts the relationship information using semantics to label the relationships. However, the method is applicable to only emails and the actors in the network are limited to the authors and recipients of the emails.

In our proposed approach, we construct the social (or criminal) networks of a suspect from his file system by processing all the text documents, identifying the personal names, and analyzing their relationships. To efficiently identify all frequent communities, we propose a data mining approach using a frequent pattern mining algorithm. We then apply information extraction techniques to derive useful information about the communities' interactions.

The rest of the paper is organized as follows. Section 2 formally defines the problems of criminal community identification. Section 3 describes our proposed approach. Section 4 shows the performance analysis of our proposed method on a real-life dataset. Section 5 concludes the paper.

2. THE PROBLEM

The problem of criminal community discovery is to identify hidden communities from a set of text documents obtained from one (or multiple) suspect's file systems. In this paper, a text document is broadly defined to include e-mail messages, chat log sessions, webpages, blogs, and text files. Let D be a set of text documents. Let $U = \{p_1, \ldots, p_m\}$ denote the universe of all personal names in D. Each document $d \in D$ is represented as a set of names such that $d \subseteq U$. Let $C \subseteq U$ be a set of personal names called a *community*. A document d *contains* a community C if $C \subseteq d$. A community having k personal names is a *k-community*. For example, $C = \{p_2, p_3, p_7\}$ is a 3-community. The support of a community C is the number of documents in D that contain C. A community C is a *frequent community* in a set of documents D if the support of C is greater than or equal to some user-specified minimum support threshold.

DEFINITION 2.1 (FREQUENT COMMUNITY). Let D be a set of text documents. Let $support(C)$ be the number of documents in D that contain C, where $C \subseteq U$. A community C is *frequent* in D if $support(C) \geq min\_sup$, where the minimum support $min\_sup > 0$ is a user-specified integer threshold. ■

Since the input set of text documents is obtained from a suspect's file system, we consider that the identified frequent communities are related to the suspect. The problem of criminal community discovery is formally defined as follows:

DEFINITION 2.2 (CRIMINAL COMMUNITY DISCOVERY). Let D be a set of text documents. Let min_sup be user-specified minimum support threshold. The *problem of criminal community discovery* is to identify all frequent communities from D, i.e., all communities $\{C_1, \ldots, C_m\}$ that have $support(C_j) \geq min\_sup$ for $1 \leq j \leq m$, and to extract information about C_j that is useful for crime investigation. ■

3. MINING CRIMINAL COMMUNITIES

We provide an overview of the proposed criminal community discovery method. The first task is to read a collection of text documents and extract personal names from the text. The name extraction task is followed by a normalization process to eliminate duplicate names that refer to the same person. The next task is to extract the frequent communities from the document files. Then, we extract the information that is valuable to investigators, such as contact information and summary topics of the documents. Finally, we provide a visual representation of the frequent communities and their related information.

3.1 Criminal Community Discovery

To identify frequent communities from a given set of text documents, the first step is to identify the personal names. There are many Named Entity Recognition (NER) tools and methods available in the market to extract personal names. In our system, we adopt the Stanford Named Entity Tagger [3] to identify English names. Each document in the given data set is considered to be a transaction. For each document d, we apply the NER tagger, obtain a bag of entities R, and then remove any duplicates so that R contains a set of distinct names. Furthermore, variants of the same name are represented as one name. For instance, *Jason*, *J. Smith*, and *Jason Smith* are transformed into a common form: *Jason Smith*. In many cases, the full name of a person occurs in the same document with the initials of the same person. Since the full name is considered, discarding the initials in these cases will cause no information loss.

Our method allows the user to incorporate his/her domain knowledge. User interference to guide this step is crucial to improve the quality of the results in the subsequent communities extraction and information extraction steps. In some cases, an individual may have different names. For instance, suspects can have nicknames in a chat log and in the same session their real names are mentioned. According to the identification method, these will be considered as two individuals if there is no lexicographical resemblance between the two names. Our method allows the investigator to merge the two names into one prior to the analysis step. This will reduce redundancy in the identified communities, resulting in a more precise analysis. Additionally, the user may remove any false positives produced by the NER tagger.

Once the individuals are identified, the next step is to identify *all* frequent criminal communities. Two or more individuals interacting frequently indicates a strong linkage,

from our analysis point of view. Analyzing the strength of linkages is a key step for effective crime investigation. The strength of a linkage can be measured by either absolute strength or relevant strength. Absolute strength is measured by comparing the frequency of interaction between the individuals to a fixed threshold. A linkage is strong if the number of interactions passes a given threshold; otherwise, the linkage is weak or there is no linkage. Alternatively, one can measure the strength of a linkage by relevance, a more flexible method that requires no prior knowledge about the data set. This is achieved by setting a threshold as percentage of the total number of text documents instead of a fixed value. A group is considered to be a frequent community if its support is greater than or equal to a given integer threshold or a percentage threshold.

A naive approach to identifying all frequent communities is to enumerate all possible communities and identify the frequent ones by counting the support of each community in D. Yet, if the number of identified individuals $|U|$ is large, it is infeasible to enumerate all possible communities because there are $2^{|U|}-1$ possible communities. To efficiently extract all frequent communities from the set of identified individuals, we employ the Apriori algorithm [1], which was originally designed to extract frequent patterns from transaction data.

Let $U = \{p_1, \ldots, p_m\}$ denote the universe of all personal names in D. Each document $d \in D$ is represented as a set of names such that $d \subseteq U$. Apriori is a level-wise iterative search algorithm that uses the frequent k-communities to explore the frequent $(k+1)$-communities. First, the set of frequent 1-communities is found by scanning the documents D, accumulating the support count of each personal name, and collecting the personal name p that has $support(p) \geq min\_sup$. The resulting frequent 1-communities are then used to find the frequent 2-communities, which are then used to find frequent 3-communities, and so on, until no more frequent k-communities can be found. The generation of frequent $(k+1)$-communities from frequent k-communities is based on the following Apriori property.

PROPERTY 3.1 (APRIORI PROPERTY). All nonempty subsets of a frequent community must also be frequent because $support(C') \geq support(C)$ if $C' \subseteq C$. ■

By definition, a community C' is not frequent if $support(C') < min\_sup$. The above property implies that adding a personal name p to an infrequent community C' will never make it frequent. Thus, if a k-community C' is infrequent, then there is no need to generate $(k+1)$-community $C' \cup p$ because $C' \cup p$ must not be frequent. The strength of the linkages among the members in a frequent community C is indicated by $support(C)$. The presented algorithm can identify *all* frequent communities by efficiently pruning all communities that cannot be frequent based on the Apriori property.

Algorithm 1 summarizes the Frequent Community Discovery algorithm. The algorithm identifies the frequent k-communities from the frequent $(k-1)$-communities based on the Apriori property. The first step is to find the set of frequent 1-communities, denoted by L_1. This is achieved by scanning the data set once and counting the support count for each 1-community C_j. The support count for C_j, denoted by $support(C_j)$, is the number of documents containing C_j. L_1 contains all frequent 1-communities C_j with $support(C_j) \geq min\_sup$. The set of frequent 1-communities

Algorithm 1 Criminal Community Discovery

Input: A set of text documents D.
Input: User-specified minimum support min_sup.
Output: Sets of frequent communities $L_1, \ldots, L_k$ with $support(C_j)$ and $R(C_j)$.
Method:

```
1:  L_1 = all frequent 1-communities in D;
2:  for (k = 2; L_{k-1} ≠ ∅; k++) do
3:     Candidates_k = L_{k-1} ⋈ L_{k-1};
4:     for all community C_j ∈ Candidates_k do
5:        if ∃X ⊂ C_j such that X ∉ L_{k-1} then
6:           Candidates_k = Candidates_k − C_j;
7:        end if
8:     end for
9:     support(C_j) = 0 and R(C_j) = ∅ for every C_j ∈
       Candidates_k;
10:    for all document d_i ∈ D do
11:       for all C_j ∈ Candidates_k do
12:          if C_j ⊆ d_i then
13:             support(C_j) = support(C_j) + 1;
14:             R(C_j) ← d_i;
15:          end if
16:       end for
17:    end for
18:    L_k = {C_j ∈ Candidates_k | support(C_j) ≥ min_sup};
19: end for
20: return L_1, ..., L_k with support(C_j) and R(C_j);
```

is then used to identify the set of candidate 2-communities, denoted by $Candidates_2$. Then the algorithm scans the data set once to count the support of each candidate C_j in $Candidates_2$. All candidates C_j that satisfy $support(C_j) \geq min\_sup$ are frequent 2-communities, denoted by L_2. The algorithm repeats the process of generating L_k from L_{k-1} and stops if $Candidate_k$ is empty.

Two frequent $(k-1)$-communities can be joined together to form a candidate k-community only if their first $(k-2)$ personal names are identical and their last $(k-1)$ personal names are different. This operation is based on the Apriori property: A community C_k cannot be frequent if any of its subsets is not frequent. Thus, the only potential frequent communities of size k are those that are formulated by joining frequent $(k-1)$-communities. Lines 4-8 describe the procedure of removing candidates that contain at least one infrequent $(k-1)$-community. Lines 9-17 describe this procedure of scanning the database and obtaining the support count of each community C_j in $Candidates_k$. Each candidate community C_j is looked up in each document d_i in the document set. Once a match is found the value of $support(C_j)$ is incremented by 1 and the document d_i is added to the set $R(C_j)$. If $support(C_j)$ is larger than the user-specified minimum threshold min_sup, then C_j is added to L_k, the set of frequent k-communities with k members. The algorithm terminates when no more candidates can be generated or none of the candidate communities pass the min_sup threshold. The algorithm returns all frequent communities $L_1, \ldots, L_k$ with support counts and sets of associated documents for each frequent community.

EXAMPLE 1 (CRIMINAL COMMUNITY DISCOVERY). Consider Table 1 with $min\_sup = 2$. First, we scan the dataset to find all frequent 1-communities and their support

Document ID	Names in d_i
d_1	{Alan, John, Kim}
d_2	{Jenny, John, Mike}
d_3	{Alan, Jenny, John, Mike}
d_4	{Jenny, Mike}

Table 1: Personal names in documents

count. The set of frequent 1-communities is

$$L_1 = \{Alan, Jenny, John, Mike\}.$$

Next, we join L_1 with itself, i.e., $L_1 \bowtie L_1$, to generate the candidate set

$$Candidates_2 = \{\{Alan, Jenny\}, \{Alan, John\}, \{Alan, Mike\}, \{Jenny, John\}, \{Jenny, Mike\}, \{John, Mike\}\}$$

and scan the dataset once to obtain the support of every community in $Candidates_2$. Then, we identify the frequent 2-communities

$$L_2 = \{\{Alan, John\}, \{Jenny, John\}, \{Jenny, Mike\}, \{John, Mike\}\}.$$

Similarly, we perform $L_2 \bowtie L_2$ to generate

$$Candidates_3 = \{Jenny,\ John,\ Mike\}$$

and scan the dataset once to identify the frequent 3-communities

$$L_3 = \{Jenny,\ John,\ Mike\}.$$

The finding of each set of frequent k-communities requires one full scan of the dataset in Table 1. ■

3.2 Community Information Extraction

The next phase is to retrieve useful information for crime investigation, such as contact information, from the identified frequent communities (criminal communities). In the context of this paper, a group of people are considered to be in the same criminal community if their names appear together frequently in some minimum number of text documents. Thus, the topics of the set of documents containing their names are the "reasons" bringing them together. By analyzing the content of the text documents containing the names of the community members, a crime investigator may obtain valuable clues that are useful for further investigation, especially in the early stages of the investigation. For instance, if a set of community member names are all contained in the same chat sessions, then summarizing the topics of the discussion can help the investigator infer the type of relationship the community members share. To facilitate crime investigation, we extract the following types of information from the set of documents $R(C_j)$ for each frequent community C_j:

1. Key topics
2. Names of other people not in C_j
3. Locations and addresses
4. Phone numbers
5. E-mail addresses
6. Website URLs
7. Relationship duration

In some real-life cyber criminal cases, there could be thousands of identified individuals and hundreds of criminal communities. Even with data mining software, an investigator may still find it difficult to cope with such a large volume of information. The summarized key topics from $R(C_j)$ can provide an investigator with an overview of each community and the related topics. The extracted key topics can be the link labels when the communities are visualized on the screen. Some people names may appear only a few times in $R(C_j)$ but may not be frequent enough to be included as a member in C_j. Identifying these infrequent personal names may lead to some new clues for investigation. Locations, addresses, and contact information, such as phone numbers and e-mail addresses, are valuable information for crime investigation because they may reveal other potential channels of communications among the community members. To extract the key topics, we employ an Open Text Summarizer (OTS) [10]. The topic extraction involves four steps:

1. *Removing stop words:* Stop words are common words that do not help differentiate the semantic of the text. Examples of stop words in English are *a*, *the*, *he*, *them*, and *who*.
2. *Stemming*: The next step is to conflate words that have common stems into one term because all these words usually share the same semantic. For instance, the words *compute*, *computing* and *computer* are all stemmed to *comput*. Although *comput* is not a valid English word, it does capture the meaning of compute.
3. *Counting stemmed terms*: Each document in $R(C_j)$ is now represented as a vector of stemmed terms. The next step is to count the frequency of each of the stemmed terms.
4. *Identifying key topics*: The key topics of a document set $R(C_j)$ are the top n frequent terms in $R(C_j)$, where n is a user-specified threshold.

To extract city names, we search the documents for the cities in the GeoWorldMap database [5]. To extract other addresses, phone numbers, and e-mail addresses, we use regular expressions [4]. Other useful information may be extracted to further describe the relationship among the members of an identified frequent community, such as the duration of the relationship. The relationship duration is a key piece of information regarding the activity of members of a community; investigators are provided with a sense of a timeline for the relationships that the communities share. In order to determine the duration of relationships among a criminal community identified in a set of textual documents, we can use the metadata of these documents. The metadata of a file is the data linked to the file by the hosting system upon creation of the document. We can define the *duration* of a relationship as all or some of the values of: (1) the starting date of the relationship, (2) the ending date, (3) and the length of time the relationship lasted. We can identify the starting date of the relationship between members of a frequent community C_j by the oldest of the dates attached to the documents in $R(C_j)$. The end date of the relationship

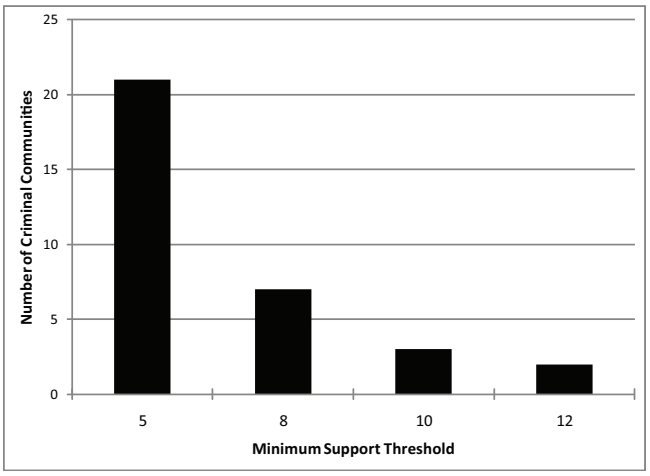

Figure 1: Number of Criminal Communities vs. Minimum Support Threshold

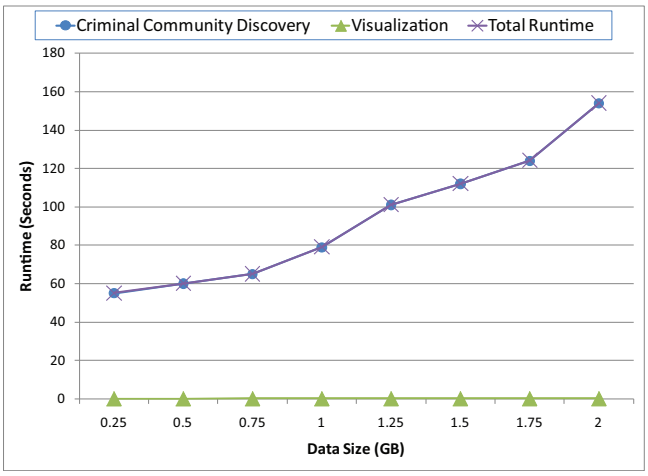

Figure 2: Scalability: Runtime vs. Data Size

is the most recent of the dates associated with these documents. The duration of the relationship is then calculated as the difference between the start and end dates.

We describe the information extraction procedure in detail. First, a document $d_i \in R(C_j)$ is represented as a set of tokens $\{tok_1, \ldots, tok_y\}$. Next, stop words are removed. The next step is to match the tokens against the regular expressions of phone numbers, e-mail addresses, locations, and website URLs. Once a match is found, the token is added to the corresponding set. Then, the next token in d_i is processed. However, if the token does not match any of the regular expressions, the token is passed to the summary topics extraction procedure. The topics are identified by first stemming the tokens, then counting their frequency. The top n most frequent tokens are selected as the document summary topics. It is important to note that words that are used as common criminal terminology or that are related by meaning to a crime topic are added to the *topics* list regardless of their frequency. This is due to their relevance to the domain in interest. A user-specified list of criminal terminologies is obtained to facilitate this step. The Open Text Summarizer method we apply for extracting the topics stems the tokens in five distinct steps: stemming punctuation prefixes, stemming punctuation suffixes, manual replacements, stemming inflection suffixes, and synonym replacements. The information extraction method returns the sets of phone numbers, email addresses, website URLs, and summary topics for each document $d_i \in R(C_j)$. The last step is to combine the sets of information for each document in $R(C_j)$ into one set for each information type.

4. PERFORMANCE ANALYSIS

We analyze the performance of the criminal community discovery method and evaluate the effectiveness of the information extraction procedure proposed in Section 3. We conducted the evaluation on the first author's machine, which included various types of files, ranging from PDF files to e-mail messages.

We evaluate the impact of the minimum support threshold on the number of discovered criminal communities. See Figure 1. As the minimum support threshold increases from 5 to 12, the number of criminal communities decreases from 21 to 2 because as the size of the community grows, the number of records containing the entire community drops quickly.

Next, we evaluate the scalability of our proposed methods by measuring the runtime of the criminal community discovery and visualization procedure. See Figure 2. To illustrate how the algorithm responds to the increase of dataset size, we incrementally increase the dataset size from 0.25GB to 2GB with $min\_sup = 3$. We intentionally set a low minimum-support threshold because a lower support threshold results in longer runtime. In general, the total runtime increases as the data size increases. For instance, the program takes 154.69 seconds to complete the two tasks for 2GB of data, excluding the time spent on reading the document files from the hard drive. The program takes 948 seconds to read from a hard drive consisting of 50,000 files with a total size of 2GB.

5. CONCLUSION

In this paper, we have proposed a method to extract criminal communities from a collection of text documents, to analyze the relationships among the members in the communities, and to visualize their interactions and associations. Previous studies on criminal network analysis mainly focus on analyzing links between criminals from some structured data. However, none of them aim to construct the social networks of a suspect from the documents on his file system. We present an efficient approach to identify all criminal communities that are related by co-occurrence. This approach is capable of identifying criminal communities as well as their interlinked subgroups. Thus, the structure of the network is identified at a more fine-grained level, a key requirement in criminal network analysis. The method also maintains a structure that facilitates an important feature applied in the visualization process. This feature allows for viewing the criminal communities with different levels of abstraction.

6. ACKNOWLEDGEMENTS

The research is supported in part by the National Cyber-Forensics and Training Alliance Canada (NCFTA Canada), Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT) new researchers start-up program, and the Concordia University Seed Funding Program.

7. REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22(2):207–216, 1993.

[2] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau. Crime data mining: a general framework and some examples. *Computer*, 37(4):50–56, 2004.

[3] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, 2005.

[4] J. E. F. Friedl. *Mastering Regular Expressions*. O'Reilly Media, 3 edition, August 2006.

[5] Geobytes Inc. Geoworldmap, 2003. http://www.geobytes.com/.

[6] T. Hope, T. Nishimura, and H. Takeda. An integrated method for social network extraction. In *Proc. of the 15th International Conference on World Wide Web (WWW)*, pages 845–846, 2006.

[7] Y. Jin, Y. Matsuo, and M. Ishizuka. Ranking companies on the web using social network mining. In I.-H. Ting and H.-J. Wu, editors, *Web Mining Applications in E-commerce and E-services*, volume 172 of *Studies in Computational Intelligence*, pages 137–152. Springer Berlin / Heidelberg, 2009.

[8] P. Klerks and E. Smeets. The network paradigm applied to criminal organizations: Theoretical nitpicking or a relevant doctrine for investigators? recent developments in the netherlands. *Connections*, 24:53–65, 2001.

[9] RCFL. Regional computer forensic laboratory annual report 2009. Technical report, Federal Bureau of Investigation, 2009. http://www.rcfl.gov/downloads/documents/RCFL_Nat_Annual09.pdf.

[10] N. Rotem. Open text summarizer, 2003. http://libots.sourceforge.net/.

[11] C. C. Yang and T. D. Ng. Terrorism and crime related weblog social network: Link, content analysis and information visualization. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 55–58, 2007.

[12] D. Zhou, R. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proc. of the 15th International Conference on World Wide Web (WWW)*, pages 173–182, 2006.

Rabeah Al-Zaidy received a M.A.Sc. degree in Information Systems Security from the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Canada, in 2010. Her current research interests include data mining, information security, and digital forensics.

Benjamin C. M. Fung received a Ph.D. degree in computing science from Simon Fraser University, Canada, in 2007. He is currently an assistant professor in the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, Canada, and a research scientist of the National Cyber-Forensics and Training Alliance Canada (NCFTA Canada). His current research interests include data mining, database, privacy preservation, information security, and digital forensics, as well as their interdisciplinary applications on current and emerging technologies. His research has been supported in part by the Discovery Grants and Strategic Project Grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) , Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT), and NCFTA Canada. Before pursuing his academic career, Dr. Fung worked at SAP Business Objects and designed reporting systems for various Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems. He is a licensed professional engineer in software engineering.

Amr Youssef is an associate professor of the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University. Before joining Concordia Institute for Information Systems Engineering at Concordia University, he worked for Nortel Networks, the Center for Applied Cryptographic Research at the university of Waterloo, IBM, and Cairo University. His main research interests are in the area of cryptology and network security. Dr. Youssef has more than 110 journal and conference publications in the area of cryptography and network security. He has co-chaired the workshop on Selected Areas in Cryptography (SAC) twice. Dr. Youssef is an active member of the National Cyber Forensics Training Alliance Canada (NCFTA Canada).