Using Relationship-Building in Event Profiling for Digital Forensic Investigations

Lynn M. Batten and Lei Pan

School of IT, Deakin University, Burwood, Victoria 3125, Australia
{lmbatten,l.pan}@deakin.edu.au

Abstract. In a forensic investigation, computer profiling is used to capture evidence and to examine events surrounding a crime. A rapid increase in the last few years in the volume of data needing examination has led to an urgent need for automation of profiling. In this paper, we present an efficient, automated event profiling approach to a forensic investigation for a computer system and its activity over a fixed time period. While research in this area has adopted a number of methods, we extend and adapt work of Marrington et al. based on a simple relational model. Our work differs from theirs in a number of ways: our object set (files, applications etc.) can be enlarged or diminished repeatedly during the analysis; the transitive relation between objects is used sparingly in our work as it tends to increase the set of objects requiring investigative attention; our objective is to reduce the volume of data to be analyzed rather than extending it. We present a substantial case study to illuminate the theory presented here. The case study also illustrates how a simple visual representation of the analysis could be used to assist a forensic team.

Keywords: digital forensics, relation, event profiling.

1 Introduction

Computer profiling, describing a computer system and its activity over a given period of time, is useful for a number of purposes. It may be used to determine how the load on the system varies, or whether it is dealing appropriately with attacks. In this paper, we describe a system and its activity for the purposes of a forensic investigation.

While there are many sophisticated, automated ways of determining system load [15] or resilience to attacks [13,16], forensic investigations have, to date, been largely reliant on a manual approach by investigators experienced in the field. Over the past few years, the rapid increase in the volume of data to be analyzed has spurred the need for automation in this area also. Additionally, there have been arguments that, in forensic investigations, inferences made from evidence are too subjective [8] and therefore automated methods of computer profiling have begun to appear [8,10]; such methods rely on logical and consistent analysis from which to draw conclusions.

X. Lai et al. (Eds.): E-Forensics 2010, LNICST 56, pp. 40–52, 2011.

There have been two basic approaches in the literature to computer profiling — one based on the raw data, captured as evidence on a hard drive for instance [3], the other examining the events surrounding the crime as in [11,12]. We refer to the latter as ***event profiling***.

In this paper, we develop an automated event profiling approach to a forensic investigation for a computer system and its activity over a fixed time period. While, in some respects, our approach is similar to that of Marrington et al. [11,12], our work both extends theirs and differs from it in fundamental ways described more fully in the next section.

In Sections 4 and 5, we present and analyze a case study to demonstrate the building of relationships between events which then lead to isolation of the most relevant events in the case. While we have not implemented it at this point, a computer graphics visualization of each stage of the investigation could assist in managing extremely large data sets.

In Section 2, we describe the relevant literature in this area. In Section 3, we develop our relational theory. Section 6 concludes the paper.

2 Background and Motivation

Models representing computer systems as finite state machines have been presented in the literature for the purposes of digital event reconstruction [3,5]. While such models are useful in understanding how a formal analysis leading to an automated approach can be established, the computational needs for carrying out an investigation based on a finite state representation are too large and complex to be practical.

The idea of linking data in large databases by means of some kind of relationship between the data goes back about twenty years to work in data mining. In [2], a set-theoretic approach is taken to formalize the notion that if certain data is involved in an event, then certain other data might also be involved in the same event. Confidence thresholds to represent the certainty of conclusions drawn are also considered. Abraham and de Vel [1] implement this idea in a computer forensic setting dealing with log data.

Since then, a number of inference models have been proposed. In [4], Garfinkel proposes cross-drive analysis which uses statistical techniques to analyze data sets from disk images. The method permits identification of data likely to be of relevance to the investigation and assigns it a high priority. While the author's approach is efficient and simple, at this stage, the work seems to apply specifically to data features found on computer drives.

In 2006, Hwang, Kim and Noh [7] proposed an inference process using Petri Nets. The principal contribution of this work is the addition of confidence levels to the inferences which accumulate throughout the investigation and the result is taken into consideration in the final drawing of conclusions. The work also permits inclusion of partial or damaged data as this can be accommodated by the confidence levels. However, the cost of analysis is high for very large data sets.

Bayesian methods were used by Kwan et al. [8] again to introduce confidence levels related to inferences. The probability that one event led to another is measured and taken into consideration as the investigation progresses. The investigative model follows that of a rooted tree where the root is a hypothesis being tested. The choice of root is critical to the model, and, if it is poorly chosen, can lead to many resource-consuming attempts to derive information.

Liu et al. [9] return to the finite state automata representation of [3,5] and introduce a transit process between states. They acknowledge that a manual check of all evidential statements is only possible when the number of intermediate states is small. Otherwise, independent event reconstruction algorithms are needed.

While methods in this area vary widely, in this paper, we follow the work of Marrington [12]. The relational device used in his work is simple and makes no restrictive assumptions. We believe, therefore, that it is one of the most efficient methods to implement.

Marrington begins by generating some information about a (computer) system based on embedded detection instruments such as log files. He then uses these initial 'relationships' to construct new information by using equivalence relations on objects which form part of a computer system's operation. These objects include hardware devices, applications, data files and also users [12, p. 69]. Marrington goes on to divide the set of all objects associated with a specific computer into four types: content, application, principal and system [12, p. 71]. A content item includes such things as documents, images, audio etc; an application includes such items as browsers, games, word processors; a principal includes users, groups and organizations; a system includes devices, drivers, registries and libraries.

In this paper, we begin with the same basic set-up as Marrington. However, our work differs in several essential ways. First, unlike Marrington, we do not assume global knowledge of the system: our set of 'objects' can be enlarged or reduced over the period of the investigation. Secondly, while Marrington uses relations to enlarge his information database, we use them primarily to reduce it; thus, we attempt to eliminate data from the investigation rather than add it. Finally, we do not assume, as in Marrington's case, that transitivity of a relation is inherently good in itself, rather, we analyze its usefulness from a theoretical perspective, and implement it when it brings useful information to the investigation.

The next section describes the relational setting.

3 Relational Theory

We begin with a set of objects $\mathbf{O}$ which is designed to be as comprehensive as possible in terms of the event under investigation. For example, for an incident in an office building, $\mathbf{O}$ would comprise all people and all equipment in the building at the time. It may also include all those off-site personnel who had access to the building's computer system at the time. In case the building has a website which interacts with clients, $\mathbf{O}$ may also include all clients in contact with the building at the time of the event.

Marrington defines two types of relationships possible between two elements of $\mathbf{O}$. One is a 'defined' relationship, such as 'Tom is related to document D because Tom is the author of D'. Another type of relationship is an 'inferred' relationship: suppose that 'document D is related to computer C' because D is stored in C and 'D is related to printer X' because X printed D. We can thus infer a relationship between C and X — for instance, that C is connected to X. Note that the precise relationship between elements of a pair here is not necessarily the same. The inferred relationship is one that must make sense between the two object types to which it refers.

In [12], the objective is to begin an investigation by establishing a set of objects and then determining the 'defined' relationships between them. Given those relationships, inferred relationships can then be constructed. In gaining new information by means of these inferred relationships, the transitivity property is crucial; it is the basis of inference. We define these concepts formally below.

In our context, $\mathbf{O}$ is the set of items perceived to be in the vicinity of, or connected to, a forensic investigation. The definitions below are standard definitions used in set theory or the theory of binary relations and can be found in [6].

Definition 1. *A* ***relation*** $\mathbb{R}$ *on* $\mathbf{O}$ *is a subset of ordered pairs of* $\mathbf{O} \times \mathbf{O}$*.*

Example 1. If $\mathbf{O}=\{a, b, c, d\}$, then the set of pairs $\{(a, c), (b, c)\}$ is a relation on $\mathbf{O}$.

Notation. If a pair (a, b) belongs to a relation $\mathbb{R}$, we also write $a\mathbb{R}b$.

Definition 2. *A relation* $\mathbb{R}$ *on* $\mathbf{O}$ *is* ***reflexive*** *if* $a\mathbb{R}a$ *for all* a *in* $\mathbf{O}$*.*

We can assume without any loss of generality that any relation on $\mathbf{O}$ in our context is reflexive since this property neither adds nor deletes information in a forensic investigative sense.

Definition 3. *A relation* $\mathbb{R}$ *on* $\mathbf{O}$ *is* ***symmetric*** *if* $a\mathbb{R}b$ *implies* $b\mathbb{R}a$ *for all objects* a *and* b *in* $\mathbf{O}$*.*

Again, without loss of generality, in our context we assume that any relation on $\mathbf{O}$ is symmetric. This assumption is based on an understanding of how objects in $\mathbf{O}$ are related. So for instance, a printer and PC are related bi-directionally in the sense that they are connected to each other.

Example 2. Let $\mathbf{O}$ be the set {printer, Joanne, laptop, memory stick, Akura}. Consider $\mathbb{R} = \{(a, a)$ for all $a \in \mathbf{O}\} \cup \{$(printer, laptop), (laptop, printer), (Akura, laptop), (laptop, Akura)$\}$. This relation is reflexive and also symmetric. The interpretation of the symmetric relation in practice is that the printer and laptop are physically connected to each other, and that the laptop belongs to Akura (and Akura to the laptop).

Definition 4. *Given a reflexive and symmetric relation* $\mathbb{R}$ *on* $\mathbf{O}$*, for each element* $a \in \mathbf{O}$*, we define a* ***relational class*** *for* a *by* $(a) = \{b \mid a\mathbb{R}b, b \in \mathbf{O}\}$*.*

In Example 2 above, (Akura) = {Akura, laptop}. Note that, because of reflexivity, a is always an element of the relational class (a).

Definition 5. *A relation $\mathbb{R}$ on $\mathbf{O}$ is* ***transitive*** *if $a\mathbb{R}b$ and $b\mathbb{R}c$ implies $a\mathbb{R}c$ for all a, b, c in $\mathbf{O}$.*

Example 3. The relation of Example 2 is easily seen not to be transitive. However, we can add some pairs to it in order to have the transitivity property satisfied: $\mathbb{R}' = \{(a, a)$ for all $a \in \mathbf{O}\} \cup \{$(printer, laptop), (laptop, printer), (Akura, laptop), (laptop, Akura), (Akura, printer), (printer, Akura)$\}$. This example now satisfies all three properties of reflexive, symmetric and transitive.

Example 3 demonstrates the crux of Marrington's work [12] and how he builds on known relationships between objects to determine new relationships between them. The facts that Akura owns the laptop and that the laptop is connected to the printer may be used to infer that Akura prints to the printer, or at least has the potential to do so. Any relation on a finite set of objects which is both reflexive and symmetric can be developed into a transitive relation by adding the necessary relationships. This is known as *transitive closure* [14] and may involve several steps before it is achieved. We formalize this statement in the following (well-known) result:

Theorem 1. *Let $\mathbb{R}$ be a reflexive and symmetric relation on a finite set $\mathbf{O}$. Then the transitive closure of $\mathbb{R}$ exists.*

We note that for infinite sets, Theorem 1 can be false [14, p. 388, 389].

Definition 6. *A relation on a set $\mathbf{O}$ is an* ***equivalence relation*** *if it is reflexive, symmetric and transitive.*

Lemma 1. *If $\mathbb{R}$ is an equivalence relation on a set $\mathbf{O}$, then for all a and b in $\mathbf{O}$, either $(a) = (b)$ or $(a) \cap (b) = \emptyset$.*

Proof. Suppose that there is an element x in $(a) \cap (b)$. So $a\mathbb{R}x$ and $x\mathbb{R}b$ results in $a\mathbb{R}b$. Then for any y such that $a\mathbb{R}y$, we obtain $b\mathbb{R}y$, and for any z such that $b\mathbb{R}z$, we obtain $a\mathbb{R}z$. Thus $(a) = (b)$. □

Lemma 2. *Let $\mathbb{R}$ be both reflexive and symmetric on a finite set $\mathbf{O}$. Then the transitive closure of $\mathbb{R}$ is an equivalence relation on $\mathbf{O}$.*

Proof. It is only necessary to show that as transitive closure is implemented, symmetry is not lost. We use induction on the number of stages used to achieve the transitive closure. Since $\mathbf{O}$ is finite, this number of steps must be finite.

In the first step, suppose that a new relational pair $a\mathbb{R}c$ is introduced. Then this pair came from two pairs, $a\mathbb{R}b$ and $b\mathbb{R}c$ for some b. Moreover, these pairs belonged to the original symmetric relation and so $b\mathbb{R}a$ and $c\mathbb{R}b$ hold; now $c\mathbb{R}b$ and $b\mathbb{R}a$ produce $c\mathbb{R}a$ by transitive closure, and so the relation is still symmetric.

Inductively, suppose that to step $k-1$, the relation achieved is still symmetric. Suppose also that at step k, the new relational pair $a\mathbb{R}c$ is introduced. Then this pair came from two pairs, $a\mathbb{R}b$ and $b\mathbb{R}c$ in step $k - 1$ for some b. Because of symmetry in step $k-1$, the pairs $b\mathbb{R}a$ and $c\mathbb{R}b$ hold. Thus, $c\mathbb{R}b$ and $b\mathbb{R}a$ produce $c\mathbb{R}a$ by transitive closure, and so the relation remains symmetric at step k. This completes the proof. □

Equivalence relations have an interesting impact on the set **O**. They partition it into equivalence classes — every element of **O** belongs to exactly one of these classes [6]. We illustrate this partition on the set **O** of Example 2 above in Figure 1.

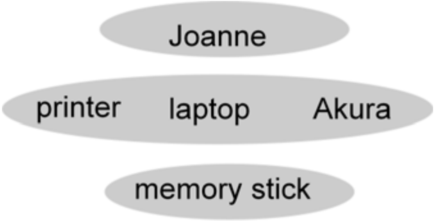

Fig. 1. A Partition Induced by an Equivalence Relation

The transitive property is the crux of the inference of relations between objects in **O**. However, we argue that one of the drawbacks is that, in taking the transitive closure, it may be the case that eventually all objects become related to each other and this provides no information about the investigation. This is illustrated in the following example.

Example 4. Xun has a laptop L and PC1, both of which are connected to a server S. PC1 is also connected to a printer P. Elaine has PC2 which is also connected to S and P. Thus, the relation on the object set **O** = {Xun, Elaine, PC1, PC2, L, S, P} is $\mathbb{R} = \{\{(a,a)$ for all $a \in \mathbf{O}\}, \{$(Xun, L), (L, Xun), (Xun, PC1), (PC1, Xun), (Xun, S), (S, Xun), (Xun, P), (P, Xun), (L, S), (S, L), (PC1, P), (P, PC1), (PC1, S), (S, PC1), (Elaine, PC2), (PC2, Elaine), (Elaine, S), (S, Elaine), (Elaine, P), (P, Elaine), (PC2, P), (P, PC2), (PC2, S), (S, PC2)$\}\}$. Figure 2 describes the impact of $\mathbb{R}$ on **O**.

Note that (S, P), (Elaine, PC1) and a number of other pairs are not part of $\mathbb{R}$.

We compute the transitive closure of $\mathbb{R}$ on **O** and so the induced equivalence relation. Since (S, PC1) and (PC1, P) hold, we deduce (S, P) and (P, S). Since (Elaine, S) and (S, PC1) hold, we deduce (Elaine, PC1) and (PC1, Elaine). Continuing in this way, we derive all possible pairs and so every object is related to every other object, giving a single equivalence class which is the entire object set **O**. We argue that this can be counter-productive in an investigation.

Our goal is in fact to isolate only those objects in **O** of specific investigative interest. We tackle this by re-interpreting the relationship on **O** in a different way from Marrington et al. [11] and by permitting the flexibility of the addition of elements to **O** as an investigation proceeds.

Below, we describe a staged approach to an investigation based on the relational method. We require that the forensic investigator set a maximal amount of time t_{max} to finish the investigation. The investigator will abort the procedure if it exceeds the pre-determined time limit or a fixed number of steps. Regarding each case, the investigator chooses the set $\mathbf{O}_1$ to be as comprehensive as possible

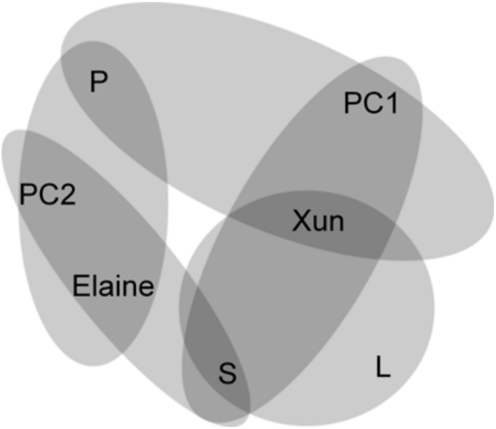

Fig. 2. The Relation $\mathbb{R}$ on the set $\mathbf{O}$ of Example 4

in the context of known information at a time relevant to the investigation and establishes a reflexive and symmetric relation $\mathbb{R}_1$ on $\mathbf{O}_1$. This should be based on relevant criteria. (See Example 4.)

We propose the following three-stage process.

Process input: A set $\mathbf{O}_1$ and a corresponding relation $\mathbb{R}_1$.

Process output: A set $\mathbf{O}_{i+1}$ and a corresponding relation $\mathbb{R}_{i+1}$.

STAGE 1. Based on the known information about the criminal activity and $\mathbb{R}_i$, investigate further relevant sources such as log files, e-mails, applications and individuals. Adjust $\mathbb{R}_i$ and $\mathbf{O}_i$ accordingly to (possibly new) sets $\mathbb{R}_i'$ and $\mathbf{O}_i'$. (If files are located hidden inside files in $\mathbf{O}_i$ these should be added to the object set; if objects not in $\mathbf{O}_i$ are now expected to be important to the investigation, these should be placed in $\mathbf{O}_i'$.)

STAGE 2. From $\mathbf{O}_i'$, determine the most relevant relational classes and discard the non-relevant ones. Call the resulting set of objects $\mathbf{O}_{i+1}$ and the corresponding relational class $\mathbb{R}_{i+1}$. (Note that $\mathbb{R}_{i+1}$ will still be reflexive and symmetric on $\mathbf{O}_{i+1}$.)

STAGE 3. If possible, draw conclusions at this stage. If further investigation is warranted and time $t < t_{max}$, return to STAGE 1 and repeat with $\mathbf{O}_{i+1}$ and $\mathbb{R}_{i+1}$. Otherwise, stop.

Note that transitivity is not used in our stages. This is to ensure that the investigator is able to focus on a small portion of the object set as the investigation develops. However, at some point, one of the $\mathbb{R}_i$ may well be an equivalence relation. This has no impact on our procedure.

Stage 1 can be viewed as a screening test which assists the investigator by establishing a baseline ($\mathbb{R}_i$ and $\mathbf{O}_i$) against which to compare other information. The baseline is then adjusted accordingly for the next stage (to $\mathbb{R}_i'$ and $\mathbf{O}_i'$). In Stage 2, this new baseline is examined to see if all objects in it are still relevant and all relations still valid. The investigator deletes any objects deemed to be unimportant and adjusts the relations accordingly. This process continues in several rounds until the investigator is satisfied that the resulting sets of objects and relations are the most relevant to the investigation. If necessary, a cut-off time can be used to establish the stopping point either for the entire process or for each of the rounds.

Our methodology can be used either alone, or as part of a multi-facets approach to an investigation with several team members. It provides good organization of the data leading to a focus on the area likely to be of most interest. It can be structured to meet an overall time target by adopting time limits to each stage. The diagrammatic approach used lends itself to a visualization of the data (as in Figures 1 and 2) which provides a simple overview of the relationships between objects, and which assists in the decision making process. We give a detailed case study in the next section.

4 Case Study

Joe operates a secret business to traffic illegal substances to several customers. One of his regular customers, Wong, sent Joe an email to request a phone conversation. The following events happened chronologically —

2009-05-01 07:30 Joe entered his office and switched on his laptop.
2009-05-01 07:31 Joe successfully connected to the Internet and started retrieving his emails.
2009-05-01 07:35 Joe read Wong's email and called Wong's land-line number.
2009-05-01 07:40 Joe started the conversation with Wong. Wong gave Joe a new private phone number and requested continuation of their business conversations through the new number.
2009-05-01 07:50 Joe saved Wong's new number in a text file named "Where.txt" on his laptop where his customers' contact numbers are stored.
2009-05-01 07:51 Joe saved Wong's name in a different text file called "Who.txt" which is a name list of his customers.
2009-05-01 08:00 Joe hid these two newly created text files in two graphic files ("1.gif" and "2.gif") respectively by using S-Tools with password protection.
2009-05-01 08:03 Joe compressed the two new GIF files into a ZIP archive file named "1.zip" which he also encrypted.
2009-05-01 08:04 Joe concatenated the ZIP file to a JPG file named "Cover.jpg".
2009-05-01 08:05 Joe used Window Washer[1] to erase 2 text files ("Who.txt" and "Where.txt"), 2 GIF files ("1.gif" and "2.gif") and 1 ZIP file ("1.zip"). (Joe did not remove the last generated file "Cover.jpg".)
2009-05-01 08:08 Joe rebooted the laptop so that all cached data in the RAM and free disk space were removed.

Four weeks later, Joe's laptop was seized by the police due to suspicion of drug possession. As part of a formal investigation procedure, police officers made a

[1] Window Washer, by Webroot, available at `http://www.webroot.com.au`

forensic image of the hard disk of Joe's laptop. Moti, a senior officer in the forensic team, is assigned the analysis task.

The next section describes Moti's analysis of the hard disk image.

5 Analysis

Moti firstly examines the forensic image file by using Forensic Toolkit[2] to filter out the files with known hash values. This leaves Moti with 250 emails, 50 text files, 100 GIF files, 90 JPG files and 10 application programs. Moti briefly browses through these files and finds no evidence against Joe. However, he notices that the program S-Tools[3] installed on the laptop is not a commonly used application and decides to investigate further.

To work more efficiently, Moti decides to use our method described in Section 3 and limits his investigation to 3 rounds. Moti includes all of the 500 items, all emails, all text files, all GIF and JPG files and all applications in a set $\mathbf{O}_1$. Because S-Tools operates on GIF files and text files, Moti establishes the relation $\mathbb{R}_1$ with the following two relational classes $\mathbb{R}_1 = \{\{$S-Tools program, 100 GIF files, 50 text files$\}, \{$250 emails, 90 JPG files, 9 programs$\}\}$. Now, Moti starts the investigation.

Round 1

Stage 1. Moti runs a data carving tool Scalpel[4] over the 500 items. He carves out 10 encrypted ZIP files, each of which is concatenated to a JPG file; Moti realizes that he has overlooked these 10 JPG files during the initial investigation. Adding the newly discovered files, Moti has $\mathbf{O}'_1 = \mathbf{O}_1 \cup \{$10 encrypted ZIP files$\}$ and defines $\mathbb{R}'_1$ based on three relational classes $\mathbb{R}'_1 = \{\{$10 ZIP files, WinZIP program$\}, \{$S-Tools program, 100 GIF files, 50 text files$\}, \{$250 emails, 90 JPG files, 8 programs$\}\}$.

Stage 2. Moti tries to extract the 10 ZIP files by using WinZIP[5]. But he is given the error messages indicating that each of the 10 ZIP files contains two GIF files all of which are password-protected. Moti suspects that these 20 GIF files contain important information and hence should be the focus of the next round. So he puts two installed programs, the 10 ZIP files and the 20 newly discovered GIF files in the set $\mathbf{O}_2 = \{$10 ZIP files, 20 compressed GIF files, 100 GIF files, 50 text files, WinZIP program, S-Tools program$\}$ and refines the relational classes $\mathbb{R}_2 = \{\{$10 ZIP files, 20 compressed GIF

[2] Forensic Toolkit (FTK), by AccessData, version 1.7, available at `http://www.accessdata.com`

[3] Steganography Tool (S-Tools), version 4.0, available at `http://www.jjtc.com/Security/stegtools.htm`

[4] Scalpel, by Golden G. Richard III, version 1.60, available at `http://www.digitalforensicssolutions.com/Scalpel/`

[5] WinZIP, by WinZip Computing, version 12, available at `http://www.winzip.com/index.htm`

files, WinZIP program}, {20 compressed GIF files, 100 GIF files, 50 text files, S-Tools program}}. (As shown in Figure 3.)

Stage 3. Moti cannot draw any conclusions to proceed with the investigation based on the current discoveries. He continues to the second round.

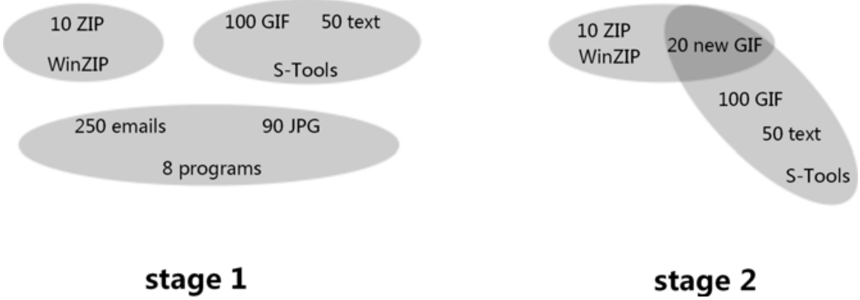

Fig. 3. Relational Classes in the Round 1 Investigation

Stage 1 of Round 1 indicates an equivalence relation on $\mathbf{O}_1$ as there is a partition of $\mathbf{O}_1$. However, in stage 2, the focus of the investigation becomes S-Tools, and so one of the relational (equivalence) classes is dropped and the new GIF files discovered are now placed in the intersection of two relational classes. Figure 3 emphasizes that there is no reason at this point to link the WinZIP program or the ZIP files with S-Tools or the other GIF and text files.

Round 2

Moti decides to explore the ten encrypted ZIP files.

Stage 1. Moti obtains the 20 compressed GIF files from the 10 ZIP files by using PRTK[6]. So, Moti redefines the set $\mathbf{O}'_2$ = {10 ZIP files, 20 new GIF files, 100 GIF files, 50 text files, WinZIP program, S-Tools program} and modifies the relational classes $\mathbb{R}'_2$ = {{10 ZIP files, 20 new GIF files, WinZIP program}, {20 new GIF files, 100 GIF files, 50 text files, S-Tools program}}.

Stage 2. Moti decides to focus on the newly discovered GIF files. Moti is confident he can remove the ZIP files from the set because he proves that every byte in the ZIP files has been successfully recovered. Moti modifies the set $\mathbf{O}'_2$ to $\mathbf{O}_3$ = {20 new GIF files, 100 GIF files, 50 text files, S-Tools program} and the relational classes $\mathbb{R}_3$ = {{20 new GIF files, 50 text files, S-Tools program}, {100 GIF files, 50 text files, S-Tools program}}. (As shown in Figure 4.)

Stage 3. Moti still cannot draw any conclusions based on the current discoveries. He wishes to extract some information in the last investigation round.

[6] Password Recovery Toolkit (PRTK), by AccessData, available at `http://www.accessdata.com`

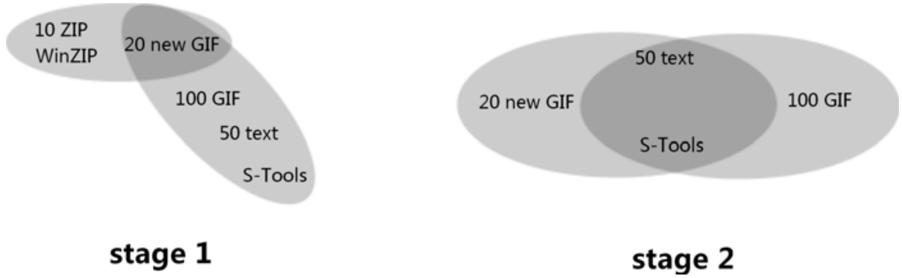

Fig. 4. Relational Classes in the Round 2 Investigation

In the first stage of Round 2, Moti recovers the GIF files identified in Round 1. In stage 2 of this round, he can now eliminate the WinZIP program and the ZIP files from the investigation, and focus on S-Tools and the GIF and text files.

Round 3

Moti tries to reveal hidden contents in the new GIF files by using the software program S-Tools found installed on Joe's laptop.

Stage 1. Since none of the password recovery tools in Moti's toolkit works with S-Tools, Moti decides to take a manual approach. As an experienced officer, Moti hypothesizes that Joe is very likely to use some of his personal details as passwords because people cannot easily remember random passwords for 20 items. So Moti connects to the police database and obtains a list of numbers and addresses related to Joe. After several trial and error attempts, Moti reveals two text files from the two GIF files extracted from one ZIP file by using Joe's medical card number. These two text files contain the name "Wong" and the mobile number 0409267531. So, Moti has the set $\mathbf{O}'_3 = \{$"Wong", "0409267531", 18 remaining new GIF files, 100 GIF files, 50 text files, S-Tools program$\}$ and the relational classes $\mathbb{R}'_3 = \{\{$"Wong", "0409267531"$\}, \{$18 remaining new GIF files, 50 text files, S-Tools program$\}, \{$100 GIF files, 50 text files, S-Tools program$\}\}$.

Stage 2. Moti thinks that the 20 new GIF files should have higher priority than the 100 GIF files and the 50 text files found in the file system because Joe might have tried to hide secrets in them. Therefore, Moti simplifies the set $\mathbf{O}'_3$ to $\mathbf{O}_4 = \{$"Wong", "0409267531", 18 remaining new GIF files, S-Tools program$\}$ and the relational classes $\mathbb{R}_4 = \{\{$"Wong", "0409267531"$\}, \{$18 remaining new GIF files, S-Tools$\}\}$. (As shown in Figure 5.)

Stage 3. Moti recommends that communications and financial transactions between Joe and Wong should be examined and further analysis is required to examine the remaining 18 new GIF files.

In the first stage of Round 3, Moti is able to eliminate two of the GIF files from the object set $\mathbf{O}_3$ as he has recovered new, apparently relevant data from them. The diagram in Figure 5 represents a non-transitive relation as there is still no

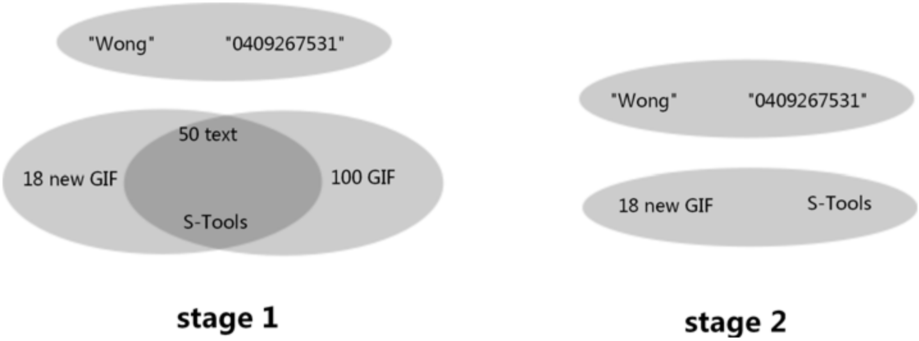

Fig. 5. Relational Classes in the Round 3 Investigation

clear connection between the 100 original GIF files and the newly discovered ones. In stage 2 of this round Moti then focuses only on the newly discovered GIF files along with S-Tools and the new information regarding "Wong". This is represented in Figure 3 by retaining one of the relational classes, completely eliminating a second and eliminating part of the third. These eliminations are possible in the relational context because we do not have transitivity.

In summary, Moti starts with a cohort of 500 digital items and ends up with two pieces of information regarding a person alongside 18 newly discovered GIF files. Moti finds useful information to advance the investigation within his limit of three rounds. Thus Moti uses three stages to sharpen the focus on the relevant evidence. This is opposite to the approach of Marrington et al. who expand the object set and relations at each stage.

6 Conclusions

We have presented relational theory designed to facilitate and automate forensic investigations into events surrounding a digital crime. This is a simple methodology which is easy to implement and which is capable of managing large volumes of data since it isolates data most likely to be of interest.

We demonstrated our theoretical model in a comprehensive case study and have indicated through this study how a visualization of the stages of the investigation can be established by means of Venn diagrams depicting relations between objects (*e.g.*, see Figures 3, 4 and 5). Future work by the authors will include development of a visualization tool to better manage data volume and speed up investigation analysis.

References

1. Abraham, T., de Vel, O.: Investigative Profiling with Computer Forensic Log Data and Association Rules. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 11–18 (2002)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)

3. Carrier, B.: File System Forensic Analysis. Upper Saddle River, Addison-Wesley (2005)
4. Garfinkel, S.L.: Forensic Feature Extraction and Cross-Drive Analysis. Digital Investigation 3, 71–81 (2006)
5. Gladyshev, P., Patel, A.: Finite State Machine Approach to Digital Event Reconstruction. Digital Investigation 1, 130–149 (2004)
6. Herstein, I.N.: Topics in Algebra, 2nd edn. Wiley, New York (1975)
7. Hwang, H.-U., Kim, M.-S., Noh, B.-N.: Expert System Using Fuzzy Petri Nets in Computer Forensics. In: Szczuka, M.S., Howard, D., Ślęzak, D., Kim, H.-k., Kim, T.-h., Ko, I.-s., Lee, G., Sloot, P.M.A. (eds.) ICHIT 2006. LNCS (LNAI), vol. 4413, pp. 312–322. Springer, Heidelberg (2007)
8. Kwan, M., Chow, K.-P., Law, F., Lai, P.: Reasoning about Evidence Using Bayesian Networks. In: Proceedings of IFIP International Federation for Information Processing. Advances in Digital Forensics IV, vol. 285, pp. 275–289. Springer, Heidelberg (2008)
9. Liu, Z., Wang, N., Zhang, H.: Inference Model of Digital Evidence based on cFSA. In: Proceedings IEEE International Conference on Multimedia Information Networking and Security, pp. 494–497 (2009)
10. Marrington, A., Mohay, G., Morarji, H., Clark, A.: Computer Profiling to Assist Computer Forensic Investigations. In: Proceedings of RNSA Recent Advances in Security Technology, pp. 287–301 (2006)
11. Marrington, A., Mohay, G., Morarji, H., Clark, A.: Event-based Computer Profiling for the Forensic Reconstruction of Computer Activity. In: Proceedings of AusCERT 2007, pp. 71–87 (2007)
12. Marrington, A.: Computer Profiling for Forensic Purposes. PhD thesis, QUT, Australia (2009)
13. Tian, R., Batten, L., Versteeg, S.: Function Length as a Tool for Malware Classification. In: Proceedings of 3rd International Conference on Malware 2008, pp. 79–86. IEEE Computer Society, Los Alamitos (2008)
14. Welsh, D.J.A.: Matroid Theory. Academic Press, London (1976)
15. Wolf, J., Bansal, N., Hildrum, K., Parekh, S., Rajan, D., Wagle, R., Wu, K.-L., Fleischer, L.K.: SODA: An Optimizing Scheduler for Large-Scale Stream-Based Distributed Computer Systems. In: Issarny, V., Schantz, R. (eds.) Middleware 2008. LNCS, vol. 5346, pp. 306–325. Springer, Heidelberg (2008)
16. Yu, S., Zhou, W., Doss, R.: Information Theory Based Detection against Network Behavior Mimicking DDoS Attacks. IEEE Communication Letters 12(4), 319–321 (2008)