

A Framework for Identifying Associations in Digital Evidence Using Metadata

A thesis by

Sriram Raghavan

Bachelor of Technology (Electronics), Anna University INDIA

Master of Science (Computer Science), IIT Madras INDIA

Submitted in accordance with the regulations

for the award of Degree of

Doctor of Philosophy

School of Electrical Engineering and Computer Science

Science & Engineering Faculty

Queensland University of Technology, Brisbane

May 2014

Keywords

Association Group, Digital Artifact, Evidence Composition, Metadata match, Metadata Associations Model, Provenance Information Model, Similarity Pocket, Similarity Group, Unified Forensic Analysis

Abstract

During a digital forensics investigation, it is often necessary to identify ‘related’ *files and log records* for analysis. While this is typically achieved through manual examination of content, recent advances in storage technologies pose two major challenges: the *growing volumes* of digital evidence; and the *technological diversity* in storage of data in different file formats and representations. Both these challenges call for a scalable approach to determine related files and logs from one or more sources of digital evidence. In this thesis, I address some of the challenges involved in identifying associations that are inherent among the sources of digital evidence, via their metadata.

Metadata pertains to information that describes the data stored in a source, be it a hard disk drive, file system, individual file, log record or a network packet. By definition, metadata as a concept is ubiquitous across multiple sources and hence presents an ideal vehicle to integrate heterogeneous sources and to identify related artifacts. I develop a metadata based model and define metadata-association based relationships to identify ‘related’ files, log records and network packets, using metadata value matches.

While there are many tools that allow the extraction of metadata for *examination* (using only a small fraction of the available metadata), the *analysis* step is left largely to the individual forensics investigator. In this thesis, I present a consolidated review of such tools and identify specific functionalities that are essential to integrate multiple sources for conducting automated analysis. Besides, I develop a framework and design the associated technology architecture to integrate these functionalities. In this framework, I define a metadata-based layer for automating the analysis – called the *metadata association model* which identifies the metadata value matches across arbitrary artifacts and organizes them based on the association semantics into meaningful groups. I have built a prototype toolkit of this model, called the AssocGEN analysis engine, for

identifying related artifacts from forensic disk images and log files. I evaluated this tool on heterogeneous collections of digital image files and word processing documents and successfully identified doctored files and determined the origin of artifacts downloaded from the Web.

Apart from associations in digital evidence that arise out of metadata value-matches, time-based sequencing is an essential part of forensic investigations. *Timestamps* are an important part of metadata that correspond to events that transpired. Sequencing these timestamps gives rise to a timeline. However, generating a timeline across heterogeneous sources poses several challenges; timestamp interpretation is one such. To address this issue, I develop a provenance information model and the associated technology architecture to incorporate timestamp interpretation while generating a unified timeline across multiple heterogeneous sources. The provenance information model can also validate time-based assertions by comparing *semantically related timestamps* to establish evidence consistency. I have built a prototype toolkit of this model, called UniTIME, to generate unified timelines from files, logs and packet captures and successfully evaluated it on datasets containing FAT file systems and ZIP file formats.

In summary, this research develops a framework for identifying associations in digital evidence using metadata for forensic analyses. I show that metadata based associations can help uncover the inherent relationships between heterogeneous digital artifacts which can aid reconstruction of past events. I also show that metadata association based analysis is amenable to automation by virtue of the ubiquitous nature of metadata across forensic disk images, files, system and application logs and network packet captures. The results obtained demonstrate that metadata based associations can be used to extract many meaningful relationships between digital artifacts, thus potentially benefitting real-life forensics investigations.

Table of Contents

Keywords	2
Abstract.....	3
List of Figures.....	10
List of Tables	12
List of Abbreviations	13
Acknowledgements.....	17
Declaration.....	18
Previously Published Material.....	19
1 Introduction.....	21
1.1 The Heterogeneous Nature of Digital Evidence	22
1.2 Motivation for Finding Associations in Digital Evidence	25
1.2.1 Concept of a Digital Artifact	27
1.2.2 Metadata in Digital Investigations.....	27
1.3 Using Metadata to Determine Associations in Digital Evidence.....	28
1.4 Objectives of this Thesis	28
1.5 Overview of the Research Method.....	30
1.6 Contributions from this Research.....	32
1.7 Chapter Summary	34
2 Related Work.....	37
2.1 Digital Forensics: A Multi-Staged Scientific Process.....	37
2.2 Related Research.....	39
2.2.1 Modeling the Digital Forensic Process.....	40
2.2.2 Evidence Acquisition and Representation.....	43
2.2.3 Evidence Examination & Discovery	45
2.2.4 Digital Forensic Analysis	48
2.3 Metadata ... in and as ... Digital Evidence.....	51
2.3.1 File Metadata.....	51
2.3.2 Use of File Metadata in Digital Forensics	53
2.3.3 Metadata for Grouping Files	54

2.3.4	Extending Metadata to Logs and Network Packet Captures	54
2.4	Timestamps as Metadata and Digital Time-lining	56
2.4.1	Timestamp Semantics and Interpretation	56
2.4.2	Causal Ordering of Events.....	57
2.4.3	Timestamp Representation Across Systems.....	58
2.5	Chapter Summary	59
3	Research Method.....	61
3.1	Research Methodology	61
3.2	Identifying the Research tasks for the Objectives.....	64
3.3	Evolution of a Metadata based Model	67
3.4	Experimental Evaluation of Model Prototype.....	69
3.4.1	Experimentation Environment.....	69
3.4.2	Experimentation Criteria	70
3.5	Chapter Summary	71
4	Determining Metadata based Associations in Digital Evidence.....	73
4.1	Review of Contemporary Forensic & Analysis Tools	74
4.1.1	Forensic Tools	74
4.1.2	Analysis Tools.....	75
4.1.3	Hypothesis Based Review	75
4.1.4	Classification and Grouping of Artifacts.....	80
4.1.5	Summary of the Review	81
4.2	<i>f</i> -FIA: Functional Forensic Integration Architecture	82
4.2.1	Digital Evidence Layer.....	83
4.2.2	Digital Artifact Traversal & Metadata Parser Layer	84
4.2.3	Evidence Composition Layer	84
4.3	Defining a Homogeneous Source of Digital Evidence	87
4.4	Method for Associating Metadata in Digital Evidence.....	91
4.5	Metadata Association Model.....	95
4.5.1	Types of Metadata Associations.....	97
4.5.2	Similarity Pockets, Similarity Groups and Association Groups.....	99
4.6	Nature of Forensic Analysis.....	107
4.7	Applying the Metadata Association Model in a Forensic Context	108
4.8	Identifying Metadata Families Relevant to Forensic Contexts	110
4.9	Deriving Digital Artifact Relationships from Metadata Associations	113
4.9.1	Existence Relationship	113

4.9.2	Source Relationship.....	113
4.9.3	Happens Before Relationship	114
4.9.4	Download Relationship	114
4.9.5	Parallel Occurrence Relationship	114
4.9.6	Structure Similarity Relationship	115
4.9.7	Unauthenticated Modification Relationship.....	115
4.9.8	Majority Relationship.....	115
4.10	Timestamp Interpretations across Heterogeneous Sources.....	116
4.10.1	Timestamps and Digital Events.....	117
4.10.2	Ambiguities in Timestamp Provenance.....	117
4.10.3	Interpreting Timestamps Using Forensic tools.....	118
4.10.4	The Timestamp Interpretation Problem.....	119
4.11	Provenance Information Model to Normalize Timestamp Interpretation	121
4.11.1	Structure for Provenance Information Model.....	122
4.11.2	Resilient Timestamps	122
4.11.3	Identifying and Validating Inconsistent Timestamps	123
4.12	Chapter Summary	124
5	Prototype Implementation.....	125
5.1	Prototype Development One: The AssocGEN Analysis Engine	125
5.1.1	Rationale for the Design	126
5.1.2	Digital Evidence Layer	128
5.1.3	Digital Artifact Traversal & Metadata Parsing Layer	128
5.1.4	Evidence Composition Layer	129
5.2	Prototype Development Two: UniTIME unified time-lining tool	142
5.2.1	Design Overview	142
5.2.2	UniTIME tool architecture	143
5.2.3	Dataflow in UniTIME	144
5.2.4	Maintaining Resilient Timestamps.....	145
5.3	Mapping Forensic Context into MAM experiments	146
5.3.1	Hypotheses for Experimentation	146
5.4	Forensic Analysis of Digital Images & Word Processing Documents	151
5.5	Chapter Summary	151
6	MAM based Analysis of Digital Images.....	153
6.1	Classification vs. Association	153
6.2	Conducting Forensic Analyses on Collections of Digital Images.....	155

6.3	MAM Based Analysis of Digital Image Collections	156
6.3.1	Criteria for Selecting Digital Image Collections	156
6.3.2	Metadata & Metadata Families in Digital Image files.....	158
6.4	Datasets	160
6.4.1	Digital Image Datasets	161
6.4.2	Dataset Characteristics	163
6.5	Conducting the Experiments	164
6.5.1	Determining the Provenance of Downloaded files.....	164
6.5.2	Image Analysis	173
6.6	Discussion	184
6.6.1	Association Index ai vs. Grouping Efficiency η	184
6.6.2	Digital Image relationships and analysis	185
6.7	Chapter Summary	185
7	MAM based Analysis of Word Processing Documents.....	187
7.1	Conducting Forensic Analysis on Collections of Word Processing Documents	188
7.2	MAM Evaluation Using Word Processing Document Collections.....	189
7.2.1	Criteria for Selecting Word Processing Document Collections	189
7.2.2	Metadata & Metadata Families in Word Processing Documents	190
7.3	Ascribing File Ownership Using Association Groups	192
7.3.1	File Ownership Problem.....	192
7.3.2	Discovering User A	193
7.3.3	Automatic Corroboration of Evidence Using Association Groups	194
7.3.4	Conclusions	194
7.4	Datasets	195
7.4.1	Word Processing Document Datasets.....	195
7.4.2	Metadata Availability in Document Datasets	198
7.4.3	Dataset Characteristics	201
7.5	Conducting Experiments.....	202
7.5.1	Identifying relevant documents with limited context	203
7.5.2	Document Analysis	205
7.6	Generating Unified Timelines Using PIM	209
7.6.1	Evaluation Criteria.....	209
7.6.2	Repeatability in Generating Unified Timelines – A Case Study	210
7.6.3	Validating Document Consistency Using Assertion Testing in PIM – A case study.....	213
7.7	Discussion	216

7.8	Chapter Summary	217
8	Conclusions & Future Work.....	219
8.1	Research Objectives & Contributions.....	220
8.1.1	Objectives of this Research	220
8.1.2	Contributions from this Research.....	220
8.2	Limitations & Future Directions	222
8.3	Conclusion	222
	References.....	223

Table of Figures

Figure 1.1 Traditional method for conducting forensic analysis on different sources.....	22
Figure 1.2 Some of the heterogeneity found in digital evidence	23
Figure 1.3 My approach to forensic analysis by identifying associations across different sources of digital evidence	26
Figure 1.4 Iterative Research Method.....	31
Figure 2.1 The various stages of the digital forensic process	38
Figure 2.2 Carrier's tool abstraction model.....	45
Figure 2.3 Taxonomy of digital forensic research literature.....	50
Figure 3.1 How my large research problem was broken down into smaller research objectives, each dealt with in sub-iterations.....	62
Figure 3.2 The research method applied in my research	63
Figure 3.3 Applying my methodology to derive digital artifact relations using metadata associations	71
Figure 4.1 Block schematic of the functional Forensic Integration Architecture (f-FIA)	83
Figure 4.2 Internal architecture of the Evidence Composition Layer	85
Figure 4.3 Example of source level hierarchy on a Microsoft Windows hard disk drive.....	88
Figure 4.4 Example of the log file data source hierarchy	90
Figure 4.5 Illustration of syntax and semantics associated with a metadata match	91
Figure 4.6 The metadata matching stage giving rise to similarity pockets	93
Figure 4.7 Grouping the overlapping similarity pockets into non-intersecting similarity groups	94
Figure 4.8 Grouping similarity groups across sources into association groups	95
Figure 4.9 Similarity pocket formed among five homogeneous documents on the value of the metadata index 'AUTHOR'.....	100
Figure 4.10 The set of all similarity pockets on some source.....	101
Figure 4.11 Conditions that govern membership to a similarity group	103
Figure 4.12 Conditions that govern membership to an association group	105
Figure 4.13 Intersecting similarity pockets across different metadata indices among a set of six documents	106
Figure 4.14 Metadata families pertinent to forensic analysis	110
Figure 4.15 Metadata tags for each metadata family across documents, logs and network packets	111
Figure 4.16 Generic timestamp structure	118
Figure 4.17 Timeline generation using traditional forensic tools	119
Figure 4.18 Differences in timestamp interpretation across timestamps with different timezone references	120
Figure 4.19 Intrinsic FAT32 timestamp interpreted with global timezone reference at a different location	121
Figure 4.20 Intrinsic NTFS timestamp interpreted on a FAT32 file system.....	121

Figure 5.1 The AssocGEN architecture	126
Figure 5.2 AssocGEN User Interface to analyze collections of digital image files from Digital Evidence	133
Figure 5.3 AssocGEN User Interface to analyze word processing documents from Digital Evidence	133
Figure 5.4 Timestamp interpretation logic for the digital time-lining tool	143
Figure 5.5 UniTIME Architecture based on f-FIA	144
Figure 5.6 UniTIME Dataflow during Timestamp analysis	145
Figure 5.7 Example of file grouping on digital image files	150
Figure 6.1 Illustrating the differences in Image classification vs. association.....	154
Figure 6.2 Digital image metadata tags of interest in Digital Investigations	159
Figure 6.3 Different probable sources for digital images discovered in digital evidence.....	160
Figure 6.4 Snapshot of the user's file system containing some digital image files	165
Figure 6.5 Snapshot of the user's temporary internet files	165
Figure 6.6 AssocGEN analysis engine processing a user file	168
Figure 6.7 AssocGEN processing temporary Internet files	168
Figure 6.8 AssocGEN code logic.....	169
Figure 6.9 AssocGEN pairing of the image files with their respective web page origins	171
Figure 6.10 Analysis of Internet Explorer History - identifying the origin of download	172
Figure 6.11 Snapshot of the specified webpage corroborating the listed files in the user's computer	172
Figure 6.12 Corroborating the findings with network trace analysis.....	173
Figure 6.13 Possible metadata associations between the different lists	175
Figure 6.14 Metadata associations discovered among the digital images from across all the datasets	178
Figure 6.15 Snapshot of AssocGEN displaying the results of classifying digital image files based on source	180
Figure 7.1 Word processing document metadata tags of interest in Digital Investigations	191
Figure 7.2 The file ownership problem.....	192
Figure 7.3 Snapshot of the partial timeline obtained sing UniTIME after harmonizing the provenance information between the different sources of digital evidence	212
Figure 7.4 The synthetic User folder structure for detecting timestamp inconsistencies	214

Table of Tables

Table 2.1 Comparison of file system metadata across different file systems	52
Table 4.1 The respective functionalities of various forensic and analysis tools	80
Table 4.2 Tabulating the nature of grouping conducted across diverse sources and digital artifacts	112
Table 6.1 Image characteristics of the five datasets.....	164
Table 6.2 Summary of the evidence analyzed and their characteristics.....	167
Table 6.3 The discovered metadata based relationships in the evidence	169
Table 6.4 Image Classification based on source	175
Table 6.5 Results of Common Source Identification for Image Datasets	177
Table 6.6 Results of association grouping to Image Datasets.....	181
Table 7.1 Summarizing the Microsoft document metadata from the different datasets	198
Table 7.2 Preliminary statistics of relative metadata richness of different Microsoft Office document types	199
Table 7.3 Percentage occurrence of metadata tags from across all word processing documents	201
Table 7.4 Outcomes from determining metadata associations on keyword matches.....	204
Table 7.5 Results from determining dataset characteristics for the two datasets.....	207
Table 7.6 Output from temporal assertion testing.....	215

List of Abbreviations

1. ACPO	<i>Association of Chief Police Officers</i>
2. AEST	<i>Australian Eastern Standard Time</i>
3. AFF	<i>Advanced Forensic Format</i>
4. ASCII	<i>American Standard Code for Information Interchange</i>
5. BMP	<i>Bitmap</i>
6. CDESF-WG	<i>Common Digital Evidence Storage Format Working Group</i>
7. CONF	<i>Configuration file</i>
8. CMOS	<i>Complementary Metal-oxide Semiconductor</i>
9. CSS	<i>Cascading Style Sheets</i>
10. DCMI	<i>Dublin Core Metadata Initiative</i>
11. DF	<i>Digital Forensics</i>
12. DE	<i>Digital Evidence</i>
13. DEB	<i>Digital Evidence Bag</i>
14. DM	<i>Data Mining</i>
15. DNS	<i>Domain Naming System</i>
16. DOC	<i>Microsoft Word Document (1997-2003)</i>
17. DOCX	<i>Microsoft Word Document (2007+)</i>
18. DoJ	<i>Department of Justice</i>
19. DFF	<i>Digital Forensic Framework</i>
20. DFRWS	<i>Digital Forensics Research Workshop</i>
21. EWF	<i>Expert Witness Format</i>
22. EXIF	<i>Exchangeable Image File format</i>
23. EXT	<i>Extended File System</i>

24. FACE	<i>Forensic Automatic Correlation Engine</i>
25. FAT	<i>File Allocation Table</i>
26. FS	<i>File System</i>
27. FTK	<i>Forensic Toolkit</i>
28. GB	<i>Gigabyte</i>
29. GIF	<i>Graphics Interchange Format</i>
30. GMT	<i>Greenwich Mean Time</i>
31. GPS	<i>Global Positioning System</i>
32. GUI	<i>Graphic User Interface</i>
33. HFS	<i>Hierarchical File System</i>
34. HTML	<i>Hypertext Manipulation Language</i>
35. HTTP	<i>Hypertext Transfer Protocol</i>
36. INI	<i>Initialization File</i>
37. IO	<i>Input-Output</i>
38. IP	<i>Internet Protocol</i>
39. ISO	<i>International Organization for Standardization</i>
40. JPEG	<i>Joint Pictures Experts Group</i>
41. JS	<i>JavaScript</i>
42. KB	<i>Kilobyte</i>
43. KFF	<i>Known File filter</i>
44. LDA	<i>Linear Discriminant Analysis</i>
45. MAC	<i>Modified-Accessed-Created</i>
46. MB	<i>Megabyte</i>
47. MCDFD	<i>Microsoft Compound Document File format</i>
48. MD5	<i>Message Digest hashing algorithm</i>

49. MIME	<i>Multipurpose Internet Mail Extensions</i>
50. MP3	<i>MPEG Audio layer III</i>
51. MPEG	<i>Moving Picture Experts Group</i>
52. NIJ	<i>National Institute of Justice</i>
53. NISO	<i>National Information Standards Organization</i>
54. NIST	<i>National Institute of Standards and Technology</i>
55. NTFS	<i>New Technology File System</i>
56. OCFA	<i>Open Computer Forensic Architecture</i>
57. OOXML	<i>Open Office XML format</i>
58. OS	<i>Operating System</i>
59. PCA	<i>Principal Component Analysis</i>
60. PCAP	<i>Packet Capture format</i>
61. PDA	<i>Personal Digital Assistant</i>
62. PDF	<i>Portable Document Format</i>
63. PNG	<i>Portable Network Graphics</i>
64. POSIX	<i>Portable Operating System Interface</i>
65. POT	<i>Microsoft PowerPoint template</i>
66. PPT	<i>Microsoft PowerPoint (1997-2003)</i>
67. PPTX	<i>Microsoft PowerPoint (2007+)</i>
68. PyFLaG	<i>Python Forensic Log & GUI</i>
69. RTF	<i>Rich Text Format</i>
70. S-DEB	<i>Sealed – Digital Evidence Bags</i>
71. SHA1	<i>Secure Hash Algorithm</i>
72. SMS	<i>Short Messaging Service</i>
73. SWGDE	<i>Scientific Working Group on Digital Evidence</i>

74. TCP	<i>Transmission Control Protocol</i>
75. TCT	<i>The Coroner's Toolkit</i>
76. TIFF	<i>Tagged Interchange File format</i>
77. TS	<i>Timestamp</i>
78. TXT	<i>Text document</i>
79. UDP	<i>User Datagram Protocol</i>
80. URI	<i>Universal Resource Indicator</i>
81. URL	<i>Universal Resource Locator</i>
82. USB	<i>Universal Serial Bus</i>
83. UTC	<i>Universal Coordinated Time</i>
84. XIRAF	<i>XML-based Indexing and Querying for Digital Forensics</i>
85. XLS	<i>Microsoft Excel Spreadsheet (1997-2003)</i>
86. XLSX	<i>Microsoft Excel Spreadsheet (2007+)</i>
87. XML	<i>Extensible Markup Language</i>
88. ZIP	<i>Zipped archive format</i>

Acknowledgements

At the outset, I am deeply indebted to my research panel and advisors, Prof. Colin Fidge, A/Prof. Andrew Clark, Prof. George Mohay and Dr. Bradley Schatz for continually inspiring me along the way during my PhD journey. My research had reached fruition backed by your constant support. My sincere thanks to all my colleagues at the ISI for many a stimulating conversation we've shared over the years.

I thank Dr. Michael Cohen for his thought-provoking conversations and keen interest in my research. I also thank Prof. Simson Garfinkel for providing access to the Digital Corpora repository which has been integral in experimental chapters 5 and 6. In a similar vein I also thank Drew Noakes and Thomas Gloe for sharing their digital image repositories for my experimental work.

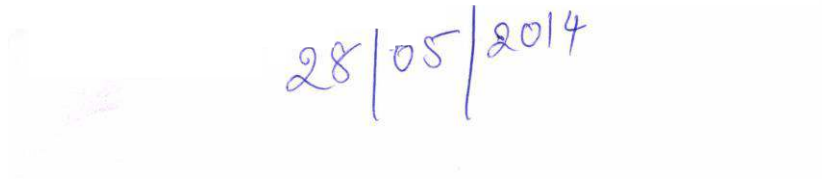
Formal education is often limited to the four walls, for a well-rounded education system, a good circle of friends and family is essential. To me personally, the Sai youth were indeed my friends and family in Brisbane, as home-away-from-home. My parents have always been a source of inspiration to me and have truly supported me in achieving my ambitions. My brother, his playful nature has always been a source of inspiration and provided me much-needed diversions in times of despair.

Declaration

The work contained in this thesis has not been previously submitted for a degree at any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed & Dated:

QUT Verified Signature

A rectangular box containing a handwritten signature in blue ink and the date "28/05/2014" written in blue ink. The signature is partially obscured by a faint pink watermark.

Previously Published Material

The following papers have been submitted, published or presented, and contain material based on the content of this thesis:

1. Raghavan S. and Raghavan S. V. (2013). Determining the Origin of Downloaded files Using Metadata Associations, *Journal of Communications*, Vol. 8(12), pp. 902-910, ISSN: 1796-2021, JCM 2013.
2. Raghavan S. (2013). Digital Forensic Research: Current State-of-the-art, *CSI Transactions on ICT*, March 2013, Volume 1(1), pp. 91-114, DOI: 10.1007/s40012-012-0008-7, ISSN (online): 2277-9086, Springer Publications 2013.
3. Raghavan S. and Raghavan S. V. (2013). AssocGEN: Engine for analyzing Metadata based associations in Digital Evidence, *In Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Accepted August 2013, Hong Kong, China, Nov 21-22, 2013, In Press.
4. Raghavan S. and Raghavan S. V., (2013). A study of Forensic & Analysis Tools, *In Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Accepted August 2013, Hong Kong, China, Nov 21-22, 2013, In Press.
5. Raghavan S. and Saran H. (2013). UniTIME: Timestamp Interpretation Engine for Generating Unified Timelines, *In Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Accepted August 2013, Hong Kong, China, Nov 21-22, 2013, In Press.
6. Raghavan S. and Raghavan S. V., (2009). Digital Evidence Composition in Fraud Detection, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2010, Volume 31(1),1-8, DOI: 10.1007/978-3-642-11534-9_1, Springer Publications 2009.

7. Raghavan S., Clark A J., and Mohay G. (2009). FIA: An Open Forensic Integration Architecture for Composing Digital Evidence., *Forensics in Telecommunications, Information and Multimedia, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2009, Volume 8(1), pp. 83-94, DOI: 10.1007/978-3-642-02312-5_10, Springer Publications 2009.
8. Raghavan S. and Raghavan S. V. (*Under Review*). Eliciting File Relationships Using Metadata Based Associations on File Collections for Digital Forensics, *Submitted to CSI Transactions on ICT*, Submitted Sept 2013.
9. Raghavan S. and Raghavan S. V. (2014). Methodology to Identify Metadata Associations in Digital Evidence, (*Manuscript under revision*).

Analysis, without associations, is incomplete.

- Anonymous

1. Introduction

Digital forensics, as a branch of science, involves the application of scientific principles to the interpretation of digital evidence during a criminal investigation. It spans acquisition, examination, analysis, documentation and presentation of digital evidence in a court of law. With the increasing use of computers and the Internet, the challenges associated with forensic investigations involving digital evidence have become formidable [67, 69]. As a consequence, issues facing the field today include the intrinsic technological *diversity*¹ (*heterogeneity*) and the increase in the number of sources of digital evidence (*volume*).

Traditionally in computing environments, hard disk drives were the dominant source of digital evidence and as a result analyses were largely confined to files. Today, however, in addition to hard disks, data is also found on volatile memory, log files and network packets, all of which are in different formats. As a consequence, system and application logs, volatile memory images and network packet traces have become equally important to investigators.

During a digital forensics investigation, each source of digital evidence is examined using one or more forensic tools to identify the artifacts contained in them which are then analyzed individually [50, 67]. When multiple heterogeneous sources of evidence are analyzed in this traditional manner, redundancy in processing the evidence becomes unavoidable, as illustrated in Figure 1.1. In fact, even among multiple sources of the same data type, redundancy results. Processing digital evidence in the traditional manner contains four parts; *source*, *process* (examination and analysis), *outcome*, and *consolidation*. For each one of the sources that require processing, the artifacts need to be examined and analyzed individually for generating relevant

¹ I henceforth use the terms *diversity* and *heterogeneity* interchangeably.

reports that are corroborated in the final step. The workflow described hitherto underlines the need for a cohesive approach to analyze diverse sources of digital evidence to arrive at a consolidated outcome.

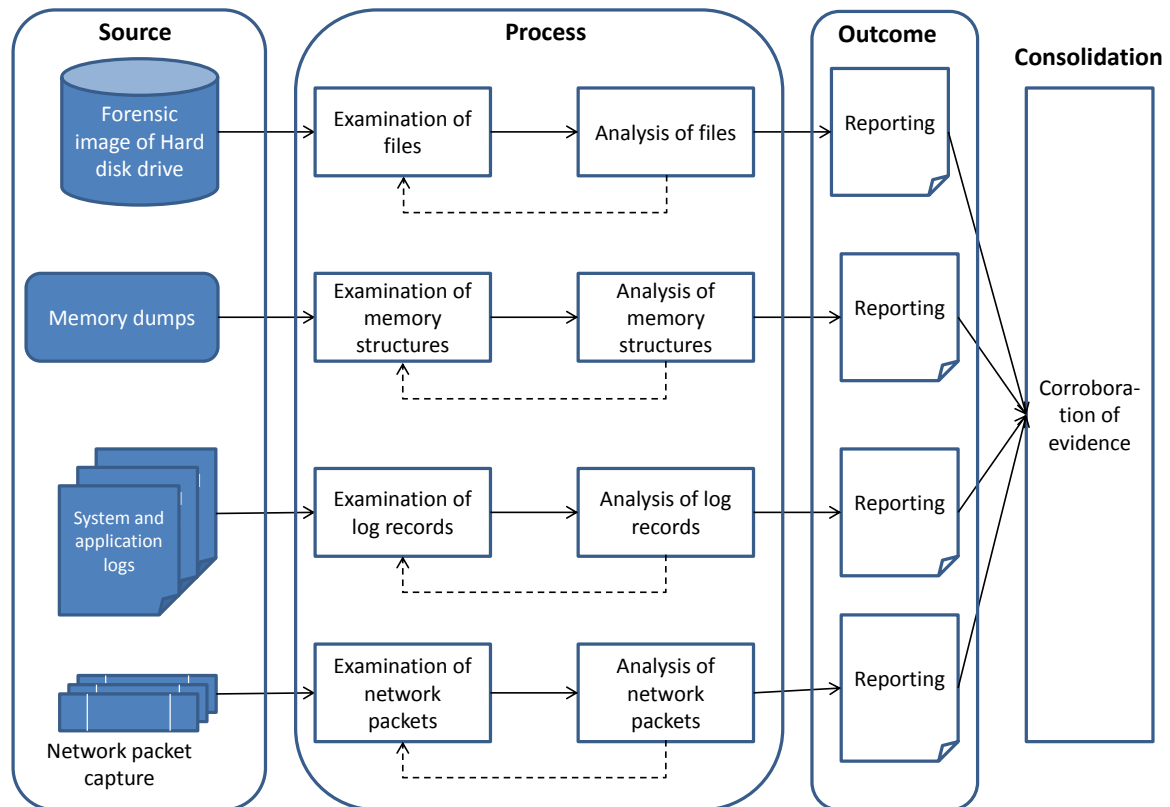


Figure 1.1 Traditional method for conducting forensic analysis on different sources

In this thesis, I posit that *it is beneficial to **group** the artifacts (available across sources of digital evidence, irrespective of their forms and formats) to enable an examiner to identify **relevant** evidence.* I achieve this goal using the metadata that is inherently present in digital evidence and identify *the associations²* between the artifacts. Interestingly, metadata based associations in digital evidence exist both at syntactic as well as semantic levels.

1.1 The Heterogeneous Nature of Digital Evidence

During a digital forensics investigation, evidence is acquired from four different types of sources [43, 44, 64], viz., hard disk drives, memory dumps, system and application logs, and network packet captures, each of which can also internally vary in formats. For example, in file systems alone there are dominant and often-used variations such as FAT, NTFS, EXTx and HFS+. The

² A connection or link between artifacts.

files, in turn, may be stored in different file formats and therefore, file analysis often requires multiple utilities to elicit evidence relevant to the investigation at hand. This is illustrated in Figure 1.2. This complexity is compounded by several usage scenarios, two typical examples of which are explained in the sequel.

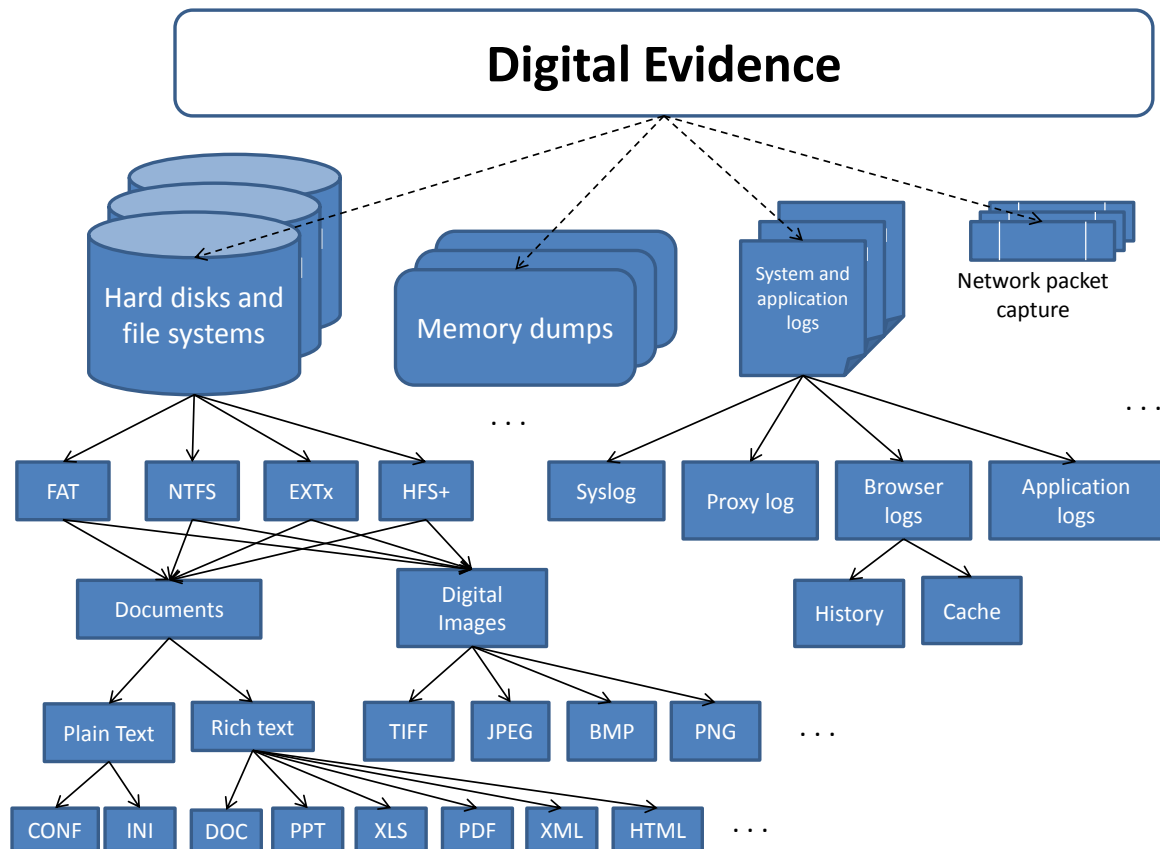


Figure 1.2 Some of the heterogeneity found in digital evidence

Scenario 1: When the same file is used across heterogeneous file systems

Today it is common for people to use multiple computers as well as multiple storage devices during their day-to-day work. Such storage devices may be formatted using the FAT32, NTFS, EXT2/3 or other such standards. If a user created a file on an NTFS file system and transferred the file to an EXT2 file system using a FAT32 formatted USB flash drive, then differently-formatted copies of the file may be present on all three file systems. Beyond this, if the user modified the file on the USB drive or the EXT2 file system, it is possible that *content similarity* or the *provenance* of the modified file may not be apparent. One can contend, however, that in such cases the metadata associated with each of the files, when analyzed in unison, can provide a better perspective and insight into the file’s origins, thus providing valuable evidence for a digital forensics

investigator. The research challenge is thus to identify such relationships across heterogeneous sources and devise a method to group them in order to aid a forensics examiner (assuming, of course, that the user does not meddle with the metadata).

Scenario 2: When an activity sequence relating to a given file spans heterogeneous sources

Consider another scenario where a user downloads a set of digital photographs from the Internet, edits them and markets them as originals. This would normally be construed as IP theft. Garfinkel notes that no existing tools can enable a forensic examiner to detect such related activities [69, 165]. While tools to detect whether or not an image has been edited are available, they fall short of detecting the activity sequence, which is necessary to create evidence linking copies to the original photographs while making a case. In order to do so, it is essential to find the original photographs from the user's computer and group them with the edited duplicate. One may also be faced with a situation where the user deleted the original files. While deleted files can be recovered using an approach known as data carving [50], trace-evidence of the user's online activity can also be obtained from the corresponding browser logs or network packet captures. If one were to use the metadata from files and the log attributes and determine related items, it can help relate these sources of digital evidence to establish *provenance*. The research challenge here is to identify log records that are directly related to the duplicates stored on a user's computer and to group the relevant pairs together to aid the forensics examiner.

The two scenarios described above highlight the challenges faced by a forensics examiner when dealing with heterogeneous data sources. In the first scenario, the challenge was to establish *file similarity* and *provenance*. In the second scenario, the challenge was to establish *provenance* and *authenticity verification*.

In both scenarios, the traditional method of analysis (based on the DFRWS report [50]) requires that the sources of digital evidence be analyzed individually and sequenced using timeline information. However, this requires that the conclusions cannot be arrived at until after all the sources are exhaustively analyzed. On the other hand, if one were to use the metadata that is present in all digital evidence sources, it has the potential to aid a forensic examiner to arrive at the same conclusion without having to exhaustively analyze all the files. Moreover, this approach can be designed to be technology-agnostic with ability to scale across heterogeneous sources.

In this thesis I present a model, methodology and a demonstrable toolkit to *automatically identify associations* in digital evidence at the syntactic and semantic levels *using metadata*. For example, if we consider my earlier description of Scenario 2, my approach can identify associations between different data sources based on metadata to elicit evidence, which could involve files that have been downloaded, evidence relating to the origins of the downloads and doctoring of digital photographs, and Internet browser logs and network packet traces. In the following section, I elaborate on this motivation further.

1.2 Motivation for Finding Associations in Digital Evidence

Digital evidence is ubiquitous in cyberspace today. As observed earlier, rapid advancements in digital technology over the past decade, the multiplicity in file formats and log formats of the artifacts have rendered forensic analysis a formidable challenge. Besides, applications also create multiple temporary files and logs hand-in-hand with regular files. In fact, in most computing applications, each and every stage of an activity is recorded at multiple levels, from the application down to the operating system level; these are often stored in different formats. Despite the differences in file formats, all files on a file system can be classified using common file system metadata like the filename, file size, MAC³ timestamps, etc. Still, files cannot always be (readily) associated with log records. Under these circumstances, in order to get a holistic perspective, during a forensics investigation, it thus becomes necessary to examine all these related files and logs along with the regular files. This is highly laborious and error prone, so a scalable method for analysis is needed.

In the literature, classification has mainly been used to group forensic artifacts belonging to the same source [17, 18, 109, 113]. The groups are then presented in some ordered form, e.g., alphabetical order or time-sorted, and analyzed for patterns. This approach seems to work well for homogeneous sources [27, 65, 69], however, when confronted with heterogeneous sources, even across file formats, classification requires an additional step by the examiner in “linking up” the groups so identified. In practice, one may have to classify the artifacts repeatedly, using different parameters, before a pattern emerges [17, 18].

To extract associations such as those illustrated in the scenarios described in the previous section, it is necessary to examine how the artifacts from heterogeneous sources are ‘connected’ in order to *corroborate a fact* [19, 130]. This can be achieved in two different ways: (i) using the actual

³ Refers to the Modified, Accessed and Created timestamps

content in the artifacts and identifying matches across artifacts; or (ii) using the **attributes** describing the artifacts, or metadata, and identifying matches in them. The former is computationally intensive and is often used in the literature whenever a deep file analysis is needed [98, 158]. On the other hand, the latter approach remains largely unexplored. I therefore focus on attributes and develop a framework to determine metadata based associations across heterogeneous sources. More specifically, I identify value matches that lead to associations that “link up” evidence. This naturally opens up a new regime of semantic understanding based on associations across artifacts. Figure 1.3 illustrates my approach.

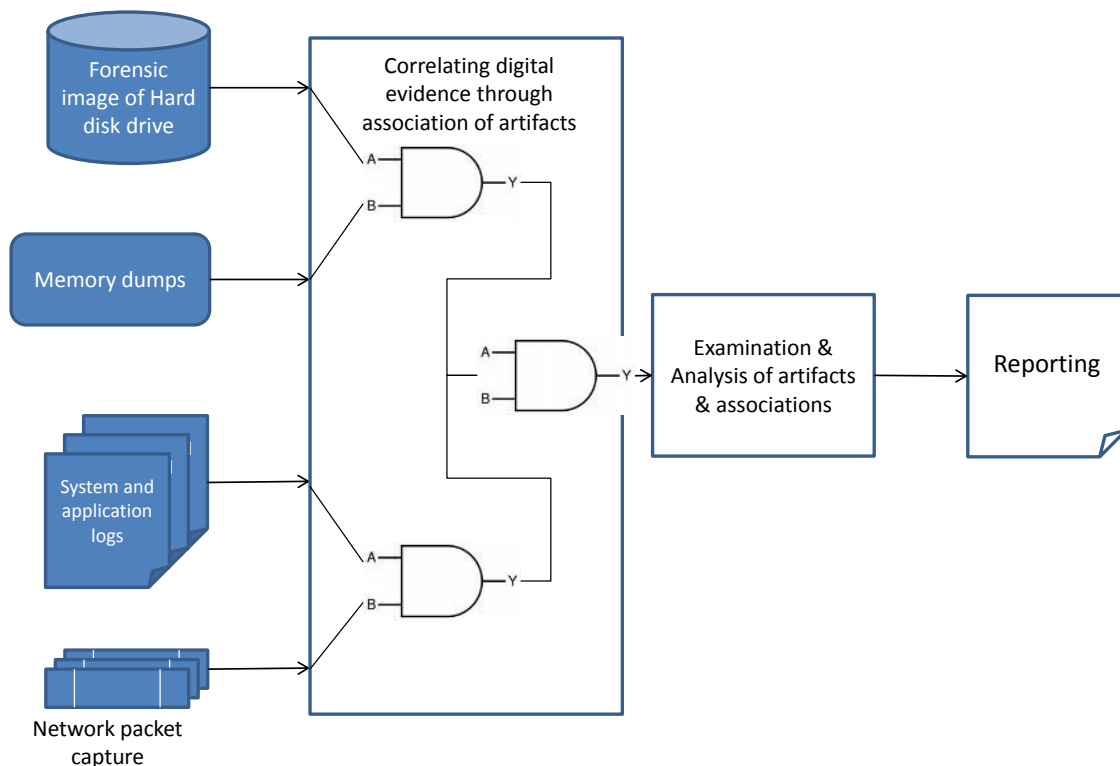


Figure 1.3 My approach to forensic analysis by identifying associations across different sources of digital evidence

In this thesis, I use metadata to identify those artifacts which contain overlapping contexts and group them to be analyzed together. As such, the grouping eliminates the repetitions which are an inevitable part of traditional analyses. In the sequel, I define a digital artifact and metadata. Further, I illustrate the role of metadata in digital investigations and describe the use of metadata to determine associations.

1.2.1 Concept of a Digital Artifact

An examiner is often confronted with the problem of *aligning* different sources to corroborate digital evidence by *correlating* the information between them. Since all types of digital evidence, notwithstanding their type, have abundant metadata [22, 67], it can act as the common medium to discover the inherent *relationships* that often exist in digital evidence. Carrier and Spafford [29] note that metadata can be treated as the characteristics of a digital object. Every digital object is evidence of at least one event and the metadata is a partial representation of the state of a *digital object*. In my work, I refer to the digital object with its associated metadata as a *digital artifact*. The metadata associated with these digital artifacts can correspond to events and thereby enable reconstruction of events and their sequence. For example, creation of a file on a file system is a type of file-event, accessing a file is another type of file-event and visiting a webpage is a type of Internet-event and so on. This abstraction of a digital artifact allows my work to focus on *not only syntactic value matches but also the semantics that links these matches*. Analyzing one or more digital artifacts can thus help in reconstructing the set of events that generated these artifacts.

1.2.2 Metadata in Digital Investigations

Metadata refers to *data about the data* that is stored in digital media. Metadata is the information about the data contained in a source, be it a file, folder, hard disk drive, logs or network traffic and *is independent of the content it describes*. For instance, metadata for a file contains information regarding the filename, location of the file, file size, content type, application type, ownership, access privileges, date and timestamps and so on. (Similar descriptions exist for log file related metadata.)

Metadata can be considered as sets of name-value pairs. It is common to all digital data stored in a digital storage medium, albeit in different forms. As metadata describes attributes regarding the data, and as these attributes can have values in common across similar digital artifacts, it is useful to group digital artifacts with the same values for attributes together in order to analyze them holistically.

Timestamps are one such kind of metadata that has been extensively used in the literature and timestamp analysis has played an important part in digital forensics so far. Timestamps are used to generate a timeline of activities relevant to an investigation. Sequencing timestamps generates a sequence of events, and this process is referred to as digital time-lining. Some of the challenges

pertaining to timestamp analysis are synchronization [184], clock skew and drift [23, 171], and timestamp interpretation [8, 34, 120-128].

Interestingly, metadata contains certain types of situational information as well. In other words, the information relating to *who*, *how* and *when* the digital artifacts were created, modified or accessed are present in their metadata. During forensic analysis, focusing on metadata enables us to understand the evolution of artifacts and their relationships with other artifacts.

1.3 Using Metadata to Determine Associations in Digital Evidence

In conventional systems of forensic analysis [50], content is analyzed for describing and understanding the artifacts. Such content analysis is carried out using “*searching*”. When searching a file or for a file, use of keywords is normally the norm. When the exact words are not known, one may use a regular expression search which supports searching for a set of keywords that fit a pattern. If a suitable search pattern too is unknown, a forensics examiner may have very little to go by during analysis and will need to resort to an exhaustive search. (A similar argument can be extended to log files as well.)

Metadata, on the other hand, contains information that can be used to achieve the same objective, but more efficiently. Metadata based search is amenable to automation by virtue of its ubiquitous nature. This property of metadata can potentially benefit digital forensic analyses, as there is always a need to identify all types of associations that exist between the digital artifacts. The research objectives of this thesis are to show how this can be done.

1.4 Objectives of this Thesis

In view of the challenges posed by heterogeneous sources and the growing volumes of digital evidence, I focus on the following three objectives in my research:

1. While there have been general advancements in the development of forensic tools, the abstraction of functionalities for an integrated analysis of heterogeneous sources of digital evidence is required. In my research, my first objective was to develop *a comprehensive understanding of the complementary functionalities of current forensic tools in order to integrate them for examining heterogeneous sources of digital evidence.*

Metadata is ubiquitous across heterogeneous sources of digital evidence. Naturally, metadata can become a vehicle for integrating the examination of different sources. The understanding gained from the first objective led to the development of an architecture which integrates the functionalities of complementary and specialized tools. To build a metadata based platform to integrate the examination and analysis of heterogeneous sources, I identify associations in digital artifacts that are present with the different sources based on metadata. I validate this understanding by developing a prototype toolkit for extracting metadata and identifying metadata associations across different sources.

When identifying metadata associations in digital evidence, I believe that it can be important to distinguish between a value match resulting in a syntactic association between two digital artifacts and the semantics of the association between them leading to a forensic context. I believe that the ability to identify syntactic associations can aid in automation, and the ability to identify semantic associations can aid in answering one or more of Casey's (six) forensic questions [32].

2. To cultivate such a distinction, my second objective was to *develop a model to represent a metadata association and a method for identifying (i) a syntactic metadata association between two or more artifacts, and (ii) a semantic metadata association between two or more artifacts.*

The understanding gained from the second objective led to the development of a framework to identify the semantics related to the associations of metadata in the context of forensic analyses and to group such related associations. This is motivated by the fact that metadata consists of several fields (called tags or names). In my work, I refer to these fields as metadata names.

When digital artifacts are associated based on metadata, it is possible that a single digital artifact can be associated with an artifact on a particular metadata name and at the same time be associated with yet another artifact on a different metadata name. In regards to forensic analysis, when the same artifact contains multiple associations, it requires consolidation.

3. Our third and last objective was to incorporate such consolidation by *grouping the related associations among digital artifacts.*

Based on the understanding gained from the second and third objectives, I developed a model to represent metadata associations both at the syntactic and semantic levels for artifacts of arbitrary

type. I validated this model by applying it to existing classes of artifacts and successfully inferred the existence of higher order relationships.

In the sequel, I present an overview of the research method used to achieve these research objectives.

1.5 Overview of the Research Method

In this research I adopted an iterative method [49, 61]. An iteration of the research method consisted of several steps. The first step was the definition of an overarching research goal. In my case this was the establishment of a framework for identifying metadata based associations across heterogeneous sources of digital evidence. The second step was the development of a framework to define the scope and solution design needed to address the research goal. The third step involved an implementation of my design which resulted in a research prototype. The fourth step was to conduct experiments using the prototype to test my hypotheses concerning the use of syntactic and semantic metadata associations in digital evidence. The fifth step involved an evaluation of the experimental outcome based on which I qualified the results and derived inferences. When the outcome addressed the research goal adequately as measured during the evaluation, I proceeded to the next research goal. If, however, the evaluation demonstrated that the solution was inadequate to address the research goal, I returned to the second step and refined the design to address the deviation in expected behavior. Thereafter, I proceeded with the method as outlined above. This process is shown in Figure 1.4.

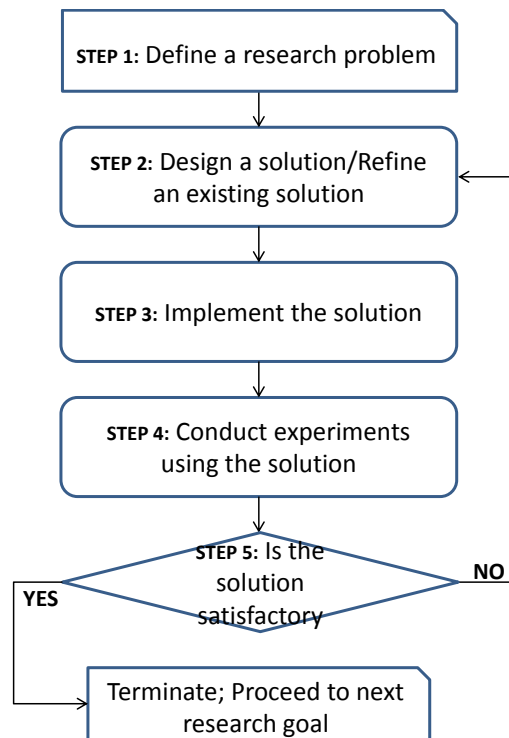


Figure 1.4 Iterative Research Method

The research method employed in this work is known as “*experimental computer science*” [49, 61, 130, 149, 186]. The output at the end of the second step resulted in a model that described the metadata associations empirically derived through experimentation. During evaluation, I tested this model to evaluate its adequacy. The third step creates the infrastructure via which the hypotheses pertaining to the use of metadata associations are tested, in the form of a research prototype for the design. The elaborated version of this research method as applied to my research is discussed in Chapter 3.

With regard to addressing the stated research objectives, my first objective involved a review of current forensic and analysis tools with regard to their treatment of metadata and the methods adopted in grouping the artifacts. To conduct such a review, I applied a survey method and identified a list of forensic and analysis tools that are used in the examination of different sources of digital evidence. The outcome of this survey was the identification of the levels of abstraction presented by these tools while handling digital evidence.

In order to achieve my second objective, it was essential to gain a “grounded” understanding of the nature of metadata matches that exist in (i) artifacts of the same type that result in syntactic metadata associations; and (ii) semantics of the metadata association interpreted in a forensic

context. Hence I adopted the grounded theory and applied metadata matches to files. In this study, I illustrated heterogeneity in file formats by identifying intra and inter metadata matches among application files such as *digital image files* and *word processing documents*. Such application files were frequently encountered during digital investigations that necessitate the grouping of related files to conduct forensic analysis. It was therefore necessary to understand how such files are ‘*interconnected*’ and assess their relevance to an investigation.

In order to achieve my third objective, it was essential to gain a “grounded” understanding of how to group the associated artifacts in a manner that can aid in answering forensically relevant questions. In this study, I group the associated digital image files and word processing documents based on ‘*source*’, ‘*ownership*’, ‘*timestamps*’ and structural application metadata concerning the file formats to determine answers to the questions posed by Casey [32].

Having outlined the research methodology in my work, I list the contributions from this research in the sequel.

1.6 Contributions from this Research

In the context of the research challenges outlined in this chapter, the following were the salient contributions from this research.

1. *An understanding of the functionalities of contemporary forensics and analysis tools* to examine digital evidence and to use the metadata for analysis. This is demonstrated through the development of a *functional forensic integration architecture* to identify associations across heterogeneous sources of digital evidence, using metadata.

This contribution addresses the first research objective. my work on this research objective has resulted in the following publications:

- i. Raghavan S. (2012)., *Digital Forensic Research: Current State-of-the-Art*, *CSI Transactions on ICT*, March 2013, Volume 1(1), pp. 91–114, Springer Publishers, Berlin Heidelberg.
- ii. Raghavan S. and Raghavan S. V. (2013)., *A Study of Forensic & Analysis Tools*, *In Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Hong Kong, China, Nov 21–22, 2013, Accepted Aug 2013, In Press.

- iii. Raghavan S., Clark A. and Mohay G. (2009)., FIA: An Open Forensic Integration Architecture for Composing Digital Evidence, In *Forensics in Telecommunications, Information and Multimedia, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2009, Volume 8(1), pp. 83–94, Springer Publishers, Berlin Heidelberg.
2. *An understanding of metadata associations* between digital artifacts when two or more artifacts exhibit (a) a syntactic metadata association resulting from a value match or similarity between corresponding metadata; and (b) semantics of a metadata association interpreted in a forensic context. This is demonstrated through the development of a metadata association model for identifying metadata-based associations among the artifacts in digital evidence. To validate this, I identified metadata associations across collections of digital images and word processing documents that were obtained from diverse sources using metadata matches.

This contribution addresses the second and third research objectives in a conceptual manner. My work on this research objective has resulted in the following publications:

 - i. Raghavan S. and Raghavan S. V. (2013)., AssocGEN: Engine for Analyzing Metadata Based Associations in Digital Evidence, In *Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Hong Kong, China, Nov 21–22, 2013, Accepted Aug 2013, In Press.
 - ii. Raghavan S. and Raghavan S. V. (2009)., Digital Evidence Composition in Fraud Detection, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2010, Volume 31(1), pp. 1–8, Springer Publishers, Berlin Heidelberg.
 3. *Development of a method to group the artifacts* in a manner that eliminates redundancy and organizes them into event-related groups for the purpose of conducting forensic analyses. This led to the identification of related digital artifacts from heterogeneous sources, whether a file, a log record or a network packet, to identify the higher-order associations or relationships via the metadata. I demonstrated this in my work by the grouping metadata associated digital image files and word processing documents that were obtained from diverse sources.

This contribution addresses the second and third research objectives in an experimental manner. my work on this research objective has resulted in the following publications:

- i. Raghavan S. and Raghavan S. V. (2013)., Determining the Origin of Downloaded Files Using Metadata Associations, *Journal of Communications*, Vol. 8(12), pp. 902-910, JCM ET Publishing 2013.
 - ii. Raghavan S. and Raghavan S. V., Eliciting File Relationships Using Metadata Based Associations on File Collections for Digital Forensics, (Under Review) *Submitted to CSI Transactions on ICT*, Springer Publications
4. *An understanding of timestamp related associations* in digital evidence and related challenges in timestamp interpretation. The analysis led to the development of a *provenance information model* to provide timestamp resilience in metadata for interpretation. I demonstrated these in my work using contemporary case studies involving FAT32 file systems and ZIP file formats.

This contribution addresses the second research objective for unifying heterogeneous events through the generation of a unified timeline. my work on this research objective has resulted in the following publication:

- i. Raghavan S. and Saran H. (2013)., UniTIME: Timestamp Interpretation Engine for Generating Unified Timelines, *In Proceedings of the Eighth International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE 2013)*, Hong Kong, China, Nov 21–22, Accepted Aug 2013, In Press.

1.7 Chapter Summary

In this chapter, I presented the need for identifying associations between the artifacts in digital evidence. Metadata, by virtue of its ubiquity, is the obvious choice for identifying these associations. My definition of association broadly conforms to the definition of the term ‘*association*’, according to Webster’s English dictionary [198a] which defines it as “*an identifier attached to an element in a system in order to indicate or permit connection with a thing or person*”. In my work, metadata is the identifier and an element is the artifact belonging to the source of digital evidence. The act of finding metadata associations refers to the identification of metadata between two or more digital artifacts that exhibit a value match.

I organize the rest of this thesis as follows:

In Chapter 2, I review related literature and motivate the need for identifying and analyzing metadata based associations in digital evidence during forensic analysis.

In Chapter 3, I present my methodology to develop a framework for identifying metadata associations among digital artifacts and for grouping them for analysis.

In Chapter 4, I develop a framework to identify metadata based associations among digital artifacts from heterogeneous sources of digital evidence. In particular, I introduce my *functional Forensic Integration Architecture* and define my *metadata association model* to determine associations using unconstrained combinations of metadata across heterogeneous sources of digital evidence. I also introduce my *provenance information model* to provide timestamp resilience in metadata for timestamp related associations and digital time-lining. This is the central theoretical contribution of the research.

In Chapter 5, I design experiments to evaluate my proposed model. I discuss the need to study the metadata found in digital image files and word processing documents and derive their respective metadata taxonomy to determine metadata associations for forensic purposes.

In Chapter 6, I present the results of the experiments that were designed in Chapter 5 to analyze digital images using the metadata association model.

In Chapter 7, I present the results of the experiments that were designed in Chapter 5 to analyze word processing documents using the metadata association model.

In Chapter 8, I summarize the contributions from my research and discuss some limitations and identify scope for future research in the area.

This page is intentionally left blank

“Learn from yesterday, live for today, hope for tomorrow. The important thing is to not stop questioning.”

- Albert Einstein

2. Related Work

The diversity and increasing volumes of digital evidence has generated a need to determine new approaches to unify multiple data sources for analysis. In this chapter, I review the digital forensics research literature, identifying significant contributions along the way and eliciting current challenges in the field. I motivate the need to determine associations in digital evidence and how metadata plays a role in deriving valuable relationships to aid forensic analysis.

2.1 Digital Forensics: A Multi-Staged Scientific Process

The Digital Forensic Research Workshop (DFRWS) 2001 report [50] has defined *digital forensic science* as follows:

“The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.”

Digital forensics is a multi-staged process starting with the identification of digital media from a scene as potential evidence until the time when the analysis results are presented in a court of law. The goal of a digital forensic investigation is the reconstruction of past events leading to an understanding of the incident being investigated. The sequence of activities [50] is illustrated at a high level in Figure 2.1.

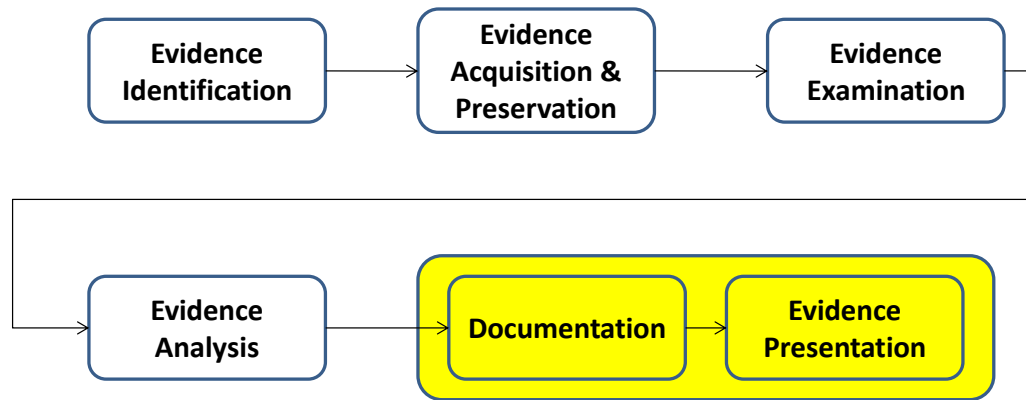


Figure 2.1 The various stages of the digital forensic process

Evidence Identification: The very first stage of the digital forensic process is the *identification* of relevant digital evidence. During this stage, one or more sources of digital data are identified as potential evidence. Examples of devices that can provide digital evidence include hard disks on computer systems, random access memory cards, USB storage devices, external sources of secondary storage, mobile phones, PDAs and so on.

Evidence Acquisition and Preservation: Once a data source is identified, its contents are forensically acquired and preserved. *Acquisition* refers to the process of obtaining a binary, bit-wise copy of the entire contents of a digital medium. The evidence is *preserved* using standard hash signatures like MD5 or SHA1 to verify its integrity. Besides such media, forensic examiners deal with digital records such as documents on a computer, telephone contact lists, lists of phone calls made, traces of signal strength from the base station of a mobile phone, voice and video files, email and SMS conversations, network traffic patterns, and virus intrusions and detections. The examiners use the actual *user data*, *metadata* associated with user data, *activity logs* and *system logs*. Each acquired source is duplicated to conduct forensic tests on read-only copies, lest an activity alters the data stored within the original sources [26, 33, 36].

Evidence Examination: The digital evidence is examined using one or more forensic tools which provide multiple file system abstractions and support schemas to enable examiners to interpret and understand raw binary data. This stage is called *evidence examination* where the sources are examined and indexed for conducting searches. Casey [32] defines forensic examination as the process of extracting information from digital evidence and making it available for analysis. In some cases, the examination of digital evidence may reveal some hidden or otherwise not-so-explicit information, which has to be extracted and subsequently analyzed. The act of identifying such information is termed *evidence discovery*.

Evidence Analysis: Evidence analysis begins when the evidence sources and the discovered data are analyzed to determine the sequence of events leading to the reported incident under investigation. Casey [32] defines forensic analysis as the application of scientific methods and critical thinking to address the fundamental questions in an investigation: what, who, why, how, when and where.

Documentation and Presentation: The individual stages are thoroughly documented and this *documentation* is presented in a court of law. Occasionally, the digital evidence may be presented in court by an expert witness.

2.2 Related Research

Current research in digital forensics can be classified into four major categories, viz. digital forensic process modeling, evidence acquisition and representation, evidence discovery and examination and digital forensic analysis.

Digital forensic process modeling deals with establishing theoretical models of the forensic process and the procedures and processes that must be in place to guarantee the integrity of evidence throughout an investigation [12, 26, 30, 50, 114]. The modeling process also defines fundamental forensic principles for the development of new tools in forensics examination, analysis and presentation.

Evidence acquisition and representation deals with that branch of digital forensics concerned with acquiring digital data in a forensically secure manner from a variety of digital devices and proposing models to represent the data contained for examination [9, 131, 134, 135]. This branch studies the forensic scope of data from different devices and presents new techniques and tools (both hardware and software) to acquire data from the field. The data so acquired is then carefully and securely imaged for examination and discovery.

Evidence examination and discovery deals with techniques to discover relevant data within the acquired sources and the software support needed to examine the contents using one or more forensic tools [26, 27, 39].

Digital forensic analysis deals with post-examination evidence study to attempt to recreate past events [26, 30, 53, 91, 114, 117]. This branch deals with the analysis of artifacts from one or more

sources of digital evidence to reconstruct the sequence of events and answer the questions pertinent to analysis.

2.2.1 Modeling the Digital Forensic Process

A digital forensic examiner typically has to contend with many different types of digital evidence during an investigation, i.e., forensic disk images, logical images of folders and files, logs, network packet traces and memory dumps. Owing to the diverse nature of digital evidence involved, there can also be a general lack of cohesiveness in the manner in which the evidence acquisition, examination and analysis are handled. The Digital Forensic Research Workshop (DFRWS) 2001 report [50] highlighted the challenges facing the field and called for new approaches to develop a better understanding of the digital forensic process.

Many digital forensic process models have been proposed in the literature. Primarily, these models deal with the definition of the general stages in a digital forensic investigation. McKemmish [114] identified four broad stages involved in a digital forensic investigation:

1. Identification of digital evidence;
2. Preservation of digital evidence;
3. Analysis of digital evidence; and
4. Presentation of digital evidence.

Among the digital forensic process models, the important ones are the physical investigation process model [26], the hierarchical objectives framework [12], the Hadley IO model [74], the computer history model [30] and the concept of digital evidence bags [87, 190]. Of these, the physical investigation model and the hierarchical objectives framework both model the entire process while the Hadley model and digital evidence bags emphasize digital evidence acquisition. The computer history model attempts to model the reconstruction process.

Carrier and Spafford [26] observed similarities in the digital investigation process with its physical twin; this work highlights the cross-applicability of many techniques used in the traditional form of physical forensics adopted into its digital sibling. Beebe and Clark [12] presented an objective based framework for digital forensic process, dividing it into six stages and proposing a 2-tier hierarchical objectives framework. The six stages defined by this work are

1. *preparation,*
2. *incident response,*
3. *data collection,*
4. *data analysis,*
5. *presentation of findings;* and
6. *incident closure.*

The framework further broke down the six stages into sub-stages (called sub-phases) and defined the objectives for these phases in typical investigations.

Activities on a computer can be treated as a series of input-output sequences. The layered Hadley model [74] for input and output defined computer-based IO as a sequence of translations followed by transport of data. This model is primarily a hardware computer model for the purposes of identifying all I/O sources of digital evidence on one computer. The Hadley model does not account for digital evidence generated from information flow on computer networks, external storage drives, logs and various other active digital devices such as mobile phones, PDAs, MP3 players and so on.

The computer history model [30] attempted to formalize digital forensics using a finite state automaton. However, it concluded that this approach is computationally infeasible owing to the size of the resulting state space. Hosmer [87] emphasized the importance of chain-of-custody equivalents in the digital world and called for auditing every operation conducted on digital evidence from digital devices. Since data on digital devices can be altered, copied or erased, Hosmer proposed the applying following principles,

- authentication,
- integrity,
- access control, and
- non-repudiation,

while handling digital evidence. The significance of this concept is reinforced by Turner's digital evidence bags [190]. Turner focused on these four aspects from the standpoint of forensic

acquisition and draws a parallel from physical investigations to define digital evidence bags to record provenance information.

Myers and Rogers [133] called for the need to standardize the forensic investigation process and presented an argument for achieving this through education and certification. Pollitt [151] presented an annotated bibliography of the different digital forensic models and examined the legal constraints of various forensic process models. Reith et al. [155] presented another independent examination of the digital forensic models and analyze its implications in the context of the challenges highlighted in the DRFWS 2001 report.

In 2003, Mocas [130] identified three main challenges that must be overcome to advance the field of digital forensics from a theoretical standpoint. These challenges are:

1. scaling forensics technology and the need to adapt scalable architectures;
2. the need to adopt uniform certification programs and courses in digital forensics; and
3. the need for changes in the digital evidence permissibility laws in courts.

In this thesis, I address the concept of the need for adaptable architectures and the need for forensics tools to scale with the technology under analysis by grounding the forensic analyses on identifying metadata matches across multiple sources of digital evidence.

Turner [190] stated that when devices become more specialized, forensic examiners require multiple tools to interpret the data contained. Existing digital forensic tools are typically designed to examine a few types of digital evidence. For instance, tools like Guidance EnCase or the AccessData Forensic Toolkit (FTK) primarily support hard disk images, albeit in different evidence formats. There are also several forensic tools in the open domain which perform specialized tasks like Sleuthkit [25], Volatility [195], Wireshark [42], etc.

The Common Digital Evidence Storage Format Working Group (CDESF-WG) [43] noted drawbacks with many current forensic tools not being able to cope with multiple forensic image formats exhaustively. CDESF-WG emphasized the need to introduce a common digital evidence storage format across multiple sources of evidence including hard disk images, network logs, proxy cache data and memory dumps.

2.2.2 Evidence Acquisition and Representation

Evidence acquisition deals with the identification of potential sources of digital evidence and how these sources may be acquired. Several national governmental agencies have recognized the increasing use of digital data and have participated in efforts to define guidelines for their use and handling in digital forensic investigations. The National Institute of Justice (NIJ) and the Department of Justice (DoJ) in the United States of America have laid down principles for first responders [134], e.g., where to search for evidence in a crime scene and how to go about acquiring data. The National Institute of Standards and Technology (NIST) have developed several tools and tool testing frameworks [138-140] for evidence acquisition. The Association of Chief Police Officers (ACPO) [9] in the United Kingdom has published the *Good Practice Guide for Computer Based Electronic Evidence* and Standards Australia [180] has laid down guidelines for the management of IT evidence in Australia.

Typically, when a digital source must be acquired, it is connected to the examiner's computer via a *write-blocker* and a binary image of the entire disk is taken. A write blocker is a hardware device or a software tool that allows read-only access to the source to avoid tampering with evidence and thus maintains data integrity [140]. Lyle described the functions of a hardware write blocker [112] and described the tool testing processes defined by NIST [138-141]. The development of a multitude of forensic acquisition tools necessitated the development of digital evidence representation models which could be processed while adhering to the requirements of a digital forensic investigation. The *digital evidence bag* and the *sealed-digital evidence bag* cater to this need [170, 190].

2.2.2.1 Digital Evidence Bags (DEB)

Turner [190] proposed the digital evidence bag (DEB) as a hierarchical digital evidence model that mirrors a physical piece of evidence. The model represents a source of digital evidence as data associated with tags (or metadata) that describe case information, evidence context, physical attributes of the source and so on. Once a source of digital evidence is tagged, it becomes immutable. But this posed a problem since during an examination when a new piece of evidence was discovered, since there was no place to record it. Therefore, DEB became monolithic and unwieldy during evidence examination and discovery [170].

2.2.2.2 Sealed Digital Evidence Bags (S-DEB)

To overcome immutability with DEB, Schatz and Clark [170] introduced the open DEB architecture called *sealed digital evidence bags* (SDEB). Each tag in the SDEB model was uniquely identified with an identifier and was made immutable. When the analysis of primary evidence resulted in secondary evidence, a new encapsulating evidence bag was created into which the new details were stored. Hence, the existing evidence bags were untouched and were secure from unintended modifications, thus guaranteeing the integrity of the digital evidence.

The models discussed here have demonstrated significant advances with regard to representing digital evidence and have provided us with different levels of abstraction to perceive digital evidence during the later stages of an investigation, such as examination and analysis. The development of such models instigated the development of different forensic image formats. There are different forensic imaging formats, viz., raw binary format, Encase image file format, advanced forensic format and so on. The *common digital evidence storage format working group* (CDESF-WG) presented a comparative study [44] of the different evidence storage formats and the forensic tools that support them.

2.2.2.3 RAW Forensic image

The raw image format is a binary image of the source; i.e., a bit-for-bit copy of the raw data of the source [44]. There is no metadata stored in raw Image Format files; however sometimes the metadata is stored in secondary files. The raw Image Format was originally native to UNIX operating system using the *dd* file copying utility, but presently it is supported by most computer forensic applications.

2.2.2.4 Encase Format (E0x)

The *Encase format* (E0x) is the basis of the image file format created by Guidance EnCase. The Encase image file format is used to store various types of digital evidence, e.g., disk image (physical bit stream of an acquired disk), volume image, memory and logical files. The Encase image file format is compressed but is a proprietary image format used by Encase forensic tools.

2.2.2.5 Advanced Forensic Format

Garfinkel [66] noted the need to maintain an open and extendable standard for forensic analysis and introduced the *advanced forensic format* (AFF) exclusively for hard disk drive images. The AFF was partitioned into two layers providing both abstraction and extended functionality. AFF's

lower *data storage layer* describes how a series of name/value pairs are stored in one or more disk files in a manner that is both operating system and byte-order independent. AFF's upper *disk presentation layer* defines a series of name/value pairs used for storing images and associated metadata. It presents an opportunity for an examiner to capture all the information related to a disk and also allows us to record case related metadata. The AFF can be processed by forensic tools like Sleuthkit and PyFlag. Garfinkel developed the *afflib*⁴ open source library to support the AFF format that has since been integrated with many open source forensic tools. Cohen et al. [41] proposed the AFF4 format by redesigning the AFF model to accommodate out-of-band information. AFF4 is a container format for multiple secondary storage devices, new data types (including network packets and memory images), extracted logical evidence, and forensic workflow.

Since initially recognizing the need to acquire digital data and use it in digital investigations, research has paved the way for many acquisition techniques and tools for evidence. Many forensic formats support varying levels of metadata information, but this introduced concerns regarding completeness of the acquired evidence [43, 44].

2.2.3 Evidence Examination & Discovery

Carrier's work on forensic tool abstraction layers [27] bridged the gap between the definition of a forensic process model and the development of associated forensic tools in aiding an investigation. Since raw data from digital evidence is often very difficult to understand, the data are translated through one or more layers of abstraction using forensic tools until they can be understood. The directory is an example of a file system abstraction while ASCII is a non-file system binary abstraction. The abstraction layer concept has been instrumental in the development of many forensic tools. The tool abstraction model proposed by Carrier is illustrated in Figure 2.2.

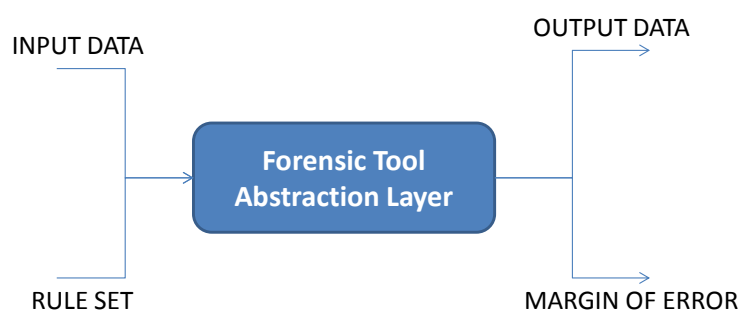


Figure 2.2 Carrier's tool abstraction model

⁴ <http://www.afflib.org/>

Carrier classified forensic tool abstraction layers as lossy or lossless. When a forensic tool processed a source of digital evidence, leaving the source intact after the processing, the tool was supposed to provide lossless abstraction. On the contrary, if a forensic tool processing a source affected the source such that the source was no longer intact, that tool was said to provide a lossy abstraction associated with a margin of error. The abstraction layers identified two types of errors introduced by forensic tools, namely, *tool implementation error* introduced by tool design errors and *abstraction error* introduced by the simplifications used to generate the tool. Pan and Batten [146] studied the reproducibility of digital evidence that builds on the abstraction layer concept.

During an evidence examination, digital evidence sources are interpreted using one or more forensic tools. Evidence discovery involves the process of *reliably*⁵ recovering encrypted, hidden, lost or deleted data from the acquired evidence for further examination. AccessData and Guidance introduced the AccessData FTK⁶ and Guidance EnCase⁷ forensic tool suites respectively for examining digital evidence. Carrier [25, 27] developed the SleuthKit⁸ framework based on the Coroner's (TCT) toolkit. Cohen [39] extended the Sleuthkit to develop the PyFlag network forensic architecture for examining forensic images of hard disks, memory dumps, network captures and logs.

The forensic community has also witnessed the advent of many other tools for examining digital evidence from hard disk images, logs, network packet captures, memory dumps, mobile phones and so on. Sleuthkit [25], Pyflag [39], Wireshark [42], log2timeline [111], tcpdump [185] and volatility [195] are a few examples⁹. Although tools such as Wireshark or tcpdump may have found their way into forensic investigations, it is interesting to note that they were not intended as forensic tools to examine and analyze digital evidence. Such tools are simply termed analysis tools. Sleuthkit and Pyflag excluded, many of the tools in the *opensourceforensics* website (refer to Footnote 9, p. 46) fall into this category, albeit for different sources.

During evidence examination, not all data may be readily available if efforts were made to conceal or eliminate data. One may need to identify and extract evidence from deleted or partial data, and recover hidden or encrypted data. The techniques associated with these methods are

⁵ This involves the process of obtaining data as it is represented in a digital evidence source, without having to manipulate or modify any information contained on that evidence source.

⁶ <http://accessdata.com/products/computer-forensics/ftk>

⁷ <http://www.guidancesoftware.com/forensic.htm>

⁸ <http://www.sleuthkit.org/>

⁹ More forensic tools can be found at <http://www2.opensourceforensics.org/tools>.

known as *data carving* and *steganography* respectively. After extraction, all the data in evidence is indexed to enable querying and searching.

2.2.3.1 Data Carving

Occasionally, evidence examination uncovers the presence of deleted or partial file data that could help an investigation. The process of uncovering such data gave rise to the new field called data carving. Data carving is the process of identifying file types using a string of bytes, called *magic numbers*, from a memory image and matching them with a database of known magic numbers to recover deleted or partially deleted files [50]. The magic number is a constant binary stream used to identify a file format and is hence unique to each format. Carving is done on a disk when the unallocated file system space is analysed to extract files because data cannot be identified due to missing allocation information, or on network captures where files are “carved” from the dumped traffic using the same techniques. One drawback of this process on disks or images is that file-carving tools typically produce many false positives [50]; hence tests must be done on each of the extracted files in order to check their consistency. A huge repository of such file types and headers are then incorporated into each forensic tool which then examines the section of data that need to be carved with the reference file signatures.

Garfinkel proposed a technique for controlling the state space explosion when carving from AFF images [64]. Richard and Roussev [157] described a high performance file carver called *Scalpel* for carving files from hard disk images.

2.2.3.2 Data Hiding and Steganography

Evidence examination is often accompanied by discovery of new information from within digital evidence and this is called evidence discovery. One such evidence discovery technique is the discovery of *steganographic content* or hidden information. Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. Digital steganography may include hiding information inside document files, image files, programs or protocols. Media files are ideal for steganographic transmission because of their large size. Hosmer and Hyde [86] discussed the challenges posed by steganography and proposed the *saturation view technique* to detect steganographic information from digital images. Lee et al [106] presented an approach for detecting image anomalies by combining computer graphics principles and AI reasoning. Image forgery has been classified into four categories, viz. deletion, insertion, photomontage and false

captioning. The approach segments a given image, computes the *importance map* on *regions of importance* and employs a rule based reasoning component to determine forgery status. Hargreaves et al. [82] described the Windows Vista format and examine the challenges it posed to forensics, while Park et al. [146] studied data concealment and detection in Microsoft Office 2007 files. Pal et al. [145] proposed a file fragmentation testing method using sequential hypothesis testing on raw forensic images to determine all sectors of a disk image where a file may have been stored.

2.2.3.3 Indexing and Querying Digital Evidence

Alink et al. [3] proposed XIRAF, XML based indexing and retrieval of stored digital evidence for querying. The XIRAF architecture indexed into raw disk images storing them in annotated XML format. The XIRAF framework consists of three subsystems; the *tool repository*, the *storage subsystem* and the *feature extraction manager*. The feature extraction manager handles the various feature extraction tools and integrates their outputs into XML which are then stored in the storage subsystem. A query engine called XQuery was used to query into the XML database for evidence related information.

In summary, over the years, researchers have devised new ways to examine digital evidence sources and discover potential sources of evidence using one or more forensic tools. However, it remains a largely manual and labour intensive process, and the growing volumes of digital evidence complicate this challenge. Garfinkel [67] noted that present-day forensic tools were designed to find new pieces of digital evidence but that the analysis continues to remain largely manual. There is a need to consolidate the research findings to provide a seamless transition from forensic examination to analysis, especially with multiple sources of digital evidence.

2.2.4 Digital Forensic Analysis

Digital forensic analysis involves the analysis of digital evidence (both direct and derived) using scientific methods to reconstruct the scenario or events. The solitary purpose of digital forensic analysis is the reconstruction of events by determining the answers for the six fundamental questions in an investigation. Studies of this nature have been carried out on different types of digital evidence, hard disks [26, 27, 64, 65], memory dumps [150, 167, 172, ,], the Microsoft Windows registry [54, , 116], log analysis [132] and time-lining from logs [90, 96, 111]. A generic approach to the event reconstruction problem involves the application of formal methods [76, 77, 91] while other techniques rely on file similarity matches in content [102, 115, 165].

2.2.4.1 Formal Methods for Event Reconstruction

A formal study to determine possible reconstruction scenarios involves the analysis of cause-and-effect sequences. In order to reconstruct the events, the digital artifacts from the evidence sources are sequenced using timestamps to set time windows within which the events could have occurred. Metadata are also utilized during analysis to determine who created or accessed the digital artifacts and how these were created or accessed. Gladyshev and Patel [77] proposed a finite state model approach for event reconstruction. However, they concluded that even a simple printer investigation problem has an exponential state space. Jeyaraman and Atallah [91] presented an empirical study of automatic reconstruction of events from logs in intrusion cases. Wang and Daniels [197] proposed an evidence graph approach to network forensic analysis and built a correlation graph using network packet captures. Garfinkel [65] studied forensic feature extraction using file carving across 750 hard disk images and determined cross drive correlation using personal identifiers such as email addresses, social security and telephone numbers. Case et al. [31] proposed the *FACE* framework for performing automatic correlations to determine static relations between network sockets in memory to TCP requests in packet captures.

2.2.4.2 File Content Similarity Detection

To identify similarities between different data files, Mead [115] explored unique file identification using hash signature mapping with NSRL database, and the Scientific Working Group on Digital Evidence (SWGDE) explored scope for digital evidence in Microsoft Windows operating systems [168, 169]. Garfinkel [65] developed a drive correlation technique to determine identical content across 750 secondary market hard disk drives. Kornblum [102] presented an approach for identifying similar files using piecewise hashing. Kornblum's aim was to automate detection of visual similarity between two files based on similarity in hash signatures. The approach combines a rolling hash with the *spamsun*¹⁰ algorithm to compare the resultant signature and determine if any similarity exists. My research also explores the concept of similarity albeit using metadata matches leading to associations among digital artifacts. Roussev et al. [164] explored hash-based similarity to retain enough information to allow binary data to be queried for similarity without additional pre-processing/indexing. Thumbnail images can be classified according to the National Library of Australia guidelines [135] to distinguish thumbnail image files from an image collection. The guidelines contain directives that state that all digital image files which range

¹⁰ <http://samba.org/~tridge/junkcode/spamsum/README>

under 160 pixels in height and/or width are to be treated as thumbnail icons. Identifying such image files during analysis would help an examiner to prune the thumbnail images.

We illustrate the various developments in digital forensic research literature through the taxonomy in Figure 2.3. Digital forensic analysis emerged as a key area of focus [50] and my review establishes a need for cohesive analysis of digital evidence by grouping and conducting holistic analysis, possibly across heterogeneous sources. Technological diversity of digital evidence and the volume in terms of the number of sources involved in a digital investigation continue to push the frontiers of research in this area. Hitherto, classification and filtering have been the techniques that examiners have relied on in identifying relevant evidence during analysis. However, in order to further an investigation, grouping related events is needed, and to achieve this, we need to be able to group related digital artifacts and derive inferences from them.

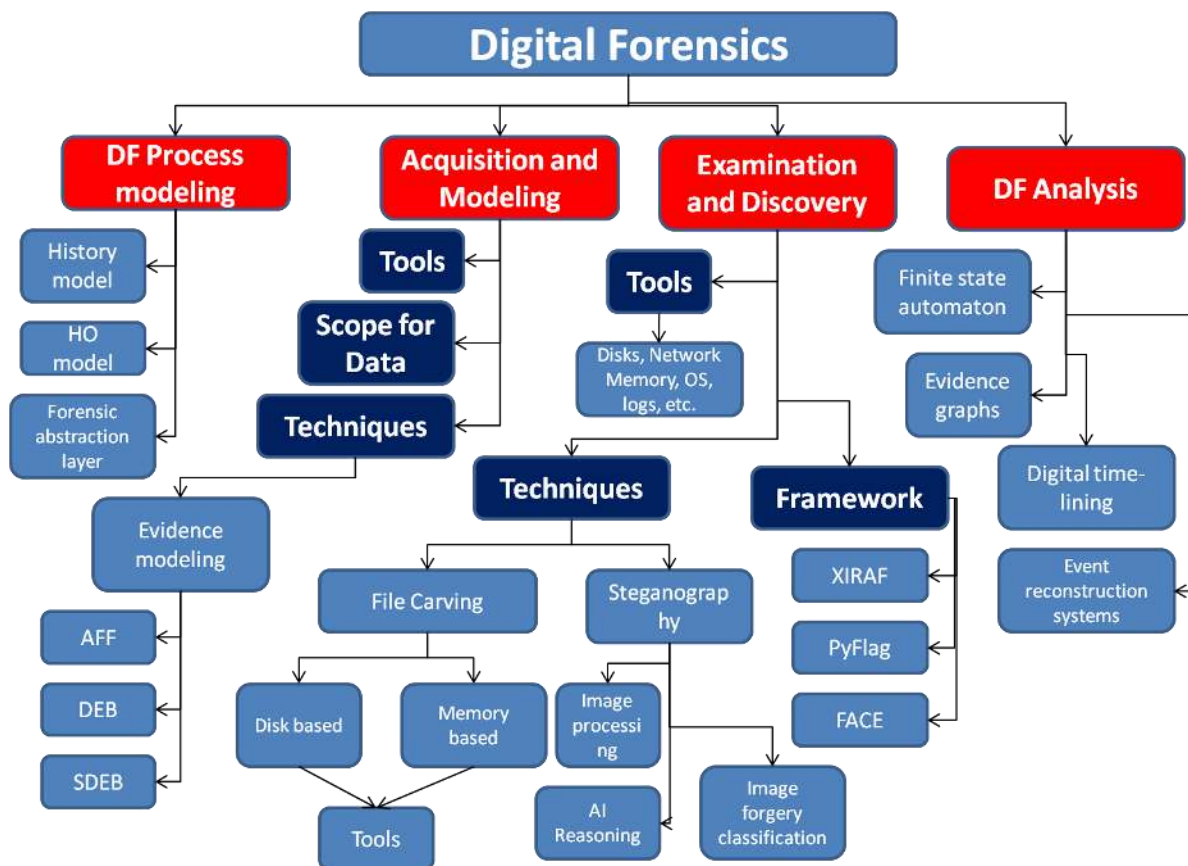


Figure 2.3 Taxonomy of digital forensic research literature

Therefore, my research focuses on grouping event-related digital artifacts together by identifying associations between the digital artifacts found in forensic evidence. I identify these associations using metadata that is inherent to all sources of digital evidence.

2.3 Metadata ... in and as ... Digital Evidence

Metadata contain information describing aspects pertaining to the digital objects or artifacts they are attributed to. Metadata provide context information that enables easy handling and management of the corresponding data or, in other words, for book-keeping purposes. There are many different types of metadata, such as system metadata, file system metadata, application metadata, document metadata, email metadata, business metadata, geographical metadata and many more. File system metadata describe attributes as recorded by a file system regarding the files, such as locations of files, MAC timestamps, file sizes, owners and access permissions. Application metadata describe attributes as recorded by the application handling the files such as file authors, file formats, content types, and encoding. Thus, the term *metadata* is an umbrella definition to encompass all such different types of metadata. According to the *Sedona Principles for Addressing Electronic Document Production*,

“metadata includes information about the document or file that is recorded by the computer (or digital device) to assist in storing and retrieving the document or file. The information may also be useful for system administration as it reflects data regarding the generation, handling, transfer and storage of the document or file within the computer (or digital device).” [175, 176]

Broadly, file system metadata and application metadata are also often referred to as *external* and *embedded* metadata [175] since file system metadata is stored external to the document or file it describes and application metadata is embedded into it.

2.3.1 File Metadata

Metadata, related to files, record the filename, location, file extension, size, MAC timestamps¹¹, author (group), and word count, etc. Some metadata may also provide additional attributes such as content length, total edit time, line count, last saved and printed timestamp, author group, last author, creator, publisher, etc. Two important types of file metadata are *file system metadata* or metadata generated by the file system regarding that file, and *application metadata* or metadata generated by specific applications about the content stored on such files.

¹¹ MAC timestamps indicate when a file was created (C), when it was last modified (M) and when it was last accessed (A).

2.3.1.1 File System Metadata

File system metadata record information that relates to the file system and helps it manage the file within that file system. Buchholz and Spafford [22] provide a qualitative treatment of file system metadata and their importance in digital forensics and briefly describe the different types of file system metadata found in different file systems. A comparison of some of the file system metadata across a few popular file systems is shown in Table 2.1.

File System	Stores File Owner	POSIX file permissions	Creation timestamp	Last Access timestamp	Last Modified timestamp	Last metadata change timestamp	Access Control lists	Extended attributes
FAT12	No	No	Yes	Yes	No	No	No	No
FAT16	No	No	Yes	Yes	Yes	No	No	No
FAT32	No	No	Yes	Yes	Yes	No	No	No
exFAT	No	No	Yes	Yes	Yes	Unknown	No	Unknown
HPFS	Yes	No	Yes	Yes	Yes	No	No	Yes
NTFS	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
HFS	No	No	Yes	No	Yes	No	No	Yes
HFS+	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
EXT2	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
EXT3	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
EXT4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2.1 Comparison of file system metadata across different file systems

(adapted from *Comparison of file system metadata*, Wikipedia¹²)

As Table 2.1 shows, the file system timestamps, i.e., the creation timestamp, the last modification timestamp and the last access timestamp are recorded by almost all file systems. Some file systems also record when the metadata was changed but this is not quite common. Some metadata like the file owner and access control are also recorded in some file systems which can come in handy during investigations. File systems that record access control lists are also used to record POSIX file permissions and extended attributes are recorded in all file systems introduced after FAT32. File system metadata has been critical in digital forensics and digital time-lining based on MAC timestamps is an integral part of a digital investigation [19, 22].

¹² http://en.wikipedia.org/wiki/Comparison_of_file_systems#Metadata

2.3.1.2 Application Metadata

Application metadata is a blanket name given to information that various applications store regarding the files they operate on describing their attributes. The National Information Standards Organization (NISO) [137] categorized application metadata into 3 categories, viz., descriptive, structural and administrative metadata. Application metadata are strongly reliant on the types of files they describe, i.e., the application metadata for a text file differs significantly from that of a Microsoft Office document or a JPEG image file. NISO presented an overview of the different metadata structuring that are prevalent and describe the *Dublin Core Metadata Initiative*¹³ (DCMI) [20] which is an international standard (ISO 15836) since 2003. Microsoft Office documents have imbibed this specification into their documents which resulted in the OOXML metadata.

2.3.2 Use of File Metadata in Digital Forensics

Buchholz and Spafford [22] examined the role of file system metadata in digital investigations and noted that despite the lack of quality and quantity of information stored in file system metadata, it played a crucial role in reconstructing events. Willassen [200] designed a method to compare the MAC timestamps in file system metadata to produce ways of antedating. Buchholz and Tjaden [23] proposed a clock model for translating MAC timestamps to address clock drift. Chow et al. [46] presented a discussion on the rules governing MAC timestamps to arrive at a systematic understanding of NTFS timestamps in file system metadata and Koen and Olivier [100] used these rules to validate files based on timestamp behavior for copy or move actions. Agarwal et al. [1] presented a study summarizing the extent to which file system metadata has grown in FAT32 and NTFS file systems over a five year period from 2000 to 2004. According to the study, they found significant temporal trends relating to the popularity of certain file types, the origin of file content, the way the namespace is used, and the degree of variation among file systems in size and capacities.

Alvarez [4] used EXIF metadata in digital photographs to verify the authenticity of a picture and determine whether or not it was altered. Kornblum [101] proposed a method to detect JPEG images that were processed by software based on analyzing the JPEG quantization tables. Huang and Fang [88] proposed a method that combines EXIF metadata with image error control codes to generate digital watermarks for copyright protection. Gloe and Bohme [78] described the Dresden Image dataset for benchmarking digital image forensics. The database aims to provide a uniform platform for researchers to test camera-based image forensic methods. It consists of over 8896 digital photographs taken with over 73 camera models and stored as JPEGs.

¹³ <http://dublincore.org/documents/dces/>

Castiglione et al. [35] highlighted the information that can be obtained from the Microsoft Compound Document File Format (MCDF) that can be relevant during digital investigations. They list some of the metadata in Microsoft documents that can potentially be useful in forensic investigations, most of which my own research incorporates for identifying metadata associations in word processing documents. Rowe and Garfinkel [165] developed a tool that used directory and file metadata to determine anomalous files on a large corpus. The tool used *fiwalk* [69] to traverse the corpus and compute statistical characteristics on the numerical metadata. The analysis generated multiple output files that were then analyzed to detect misnamed files and duplicate copies of files.

2.3.3 Metadata for Grouping Files

Boutell and Luo used EXIF metadata in digital photographs to classify camera types [17] and to perform scene classification [18]. Minack et al. [129] identified image-related metadata based searching as an effective solution for personal computers. In forensic investigations, examiners have to frequently deal with objects from personal computers and their work emphasizes the importance of metadata. Liu et al. [109] proposed a feature combination method to classify digital images that combined image content and EXIF metadata based on linear-discriminant-analysis (LDA) for digital photograph management.

Bohm and Rakow [16] discussed the different aspects of classifying multimedia documents based on document metadata. Multimedia documents can be classified into six orthogonal categories, viz., representation of media type, content description, content classification, document composition, document history and document location. Fathi et al. [59] classified web documents based on author and title in document metadata and Toyama et al. [189] built a system that utilizes geographic information in location metadata (or geotags) to classify digital photographs with same location information. Denecke et al. [47] developed a classification method using bibliographic metadata such as author and document title. Maly et al. [113] proposed a method to classify documents based on layout metadata. Lerman et al. [110] described a method to label web services and classify them based on metadata from the web services definition file (WSDF).

2.3.4 Extending Metadata to Logs and Network Packet Captures

In the traditional sense, metadata are native to files that reside on file systems. However, log records and network packets also have some associated information that can be attributed the term ‘*metadata*’. Although logs and network packet captures themselves reside as files in a file system,

the log entries and network packets they contain are discrete artifacts that correspond to specific events. For instance, an entry in the IE history log, *index.dat*, would correspond to visiting a web page characterized by a URI. The attributes describing such an entry contain the timestamp of web page visit, the domain, the host server IP address and so on. Similarly, an entry in a network packet capture corresponds to a network packet that was observed by the network capture sensor on a particular network belonging to a specific protocol containing a source and destination address. A network packet can be associated with a timestamp, source and destination IP addresses, and the protocol for transfer and payload size. Such information may be treated as metadata for a log record or a network packet, as the case may be.

Metadata for Grouping Log events & Network Packets

With regard to metadata in logs and network packet captures, timestamps are the most common metadata, used to generate timelines [19, 111, 199]. Often in network packet captures, the packets are organized according to the IP addresses and protocol in investigations involving network intrusion detection [202, 161]. Zander et al. [202] classified IP traffic based on statistical flow characteristics by filtering based on destination address and port. Roesch [161] introduced ‘snort’ intrusion detection tool that allows IP packets to be monitored and classified according to IP addresses. Jiang et al. [92] proposed a coloring scheme to identify a remotely accessible server or process to detect provenance aware self-propagating worm contaminations. This scheme associated a unique color as a system-wide identifier to each remote server or process and that is inherited by all spawned child processes.

In summary, metadata in digital forensics is kept to authentication and determination of hidden information. Metadata based classification is used for the identification of file classes and timestamp based correlation is used to discover antedating activities by comparing the MAC timestamps on the file. Metadata, by virtue of recording the partial state of a digital artifact, contain information of forensic value [29]. During an investigation where it is necessary to discover all higher order associations that exist between the digital artifacts, metadata can be used for inferring such associations. In my research, I conduct a systematic analysis of metadata association to extract higher-order associations and relationships across heterogeneous sources and group the related digital artifacts. My model to identify associations using metadata across heterogeneous digital artifacts is presented in Chapter 4 of this thesis.

When I discuss metadata associations, it is essential to take cognizance of a particular type of metadata which is extensively used to derive event sequences from different sources, viz.,

timestamps. A timestamp is the record of the time, according to some reference clock, of an associated event. Timestamps are an integral part of metadata associated with files, log records and network packets.

2.4 Timestamps as Metadata and Digital Time-lining

A timestamp has a physical realization and a temporal interpretation [55]. The physical realization is an encoding as a pattern of bits while the temporal interpretation stipulates the meaning of the bit pattern, the calendar date and time to which the pattern corresponds. Analysis of timestamps is the use of timestamps from digital evidence towards constructing a timeline of all events on a source of evidence and/or for reasoning with the sequence in which certain events are recorded on that source. Allen [2] discussed the different representations of timestamps adopted in the literature, including one where timestamps are logical timestamps only, merely a sequential numbering of events on a system. In this thesis, I adopt Dyreson and Snoddgrass's definition of a timestamp [55] and regard timestamps as a record of calendar time associated with an event recorded on a digital device. The major challenges associated with the use of timestamps in digital forensics can be limited to three broad areas, timestamp semantics and interpretation, timestamps for causal ordering of events, and timestamp representation across different systems. I review the related work in these three areas in the sequel.

2.4.1 Timestamp Semantics and Interpretation

Timestamp semantics and its interpretation are acknowledged as complex and challenging tasks [19, 23, 171, 199]. Weil [199] presented a method for correlating times and dates contained in application metadata to the file system's timestamps. The method attempts to standardize the apparent file MAC times to the actual time and concluded that increasing the number of independent sources enhances the reliability of the data and minimizes CMOS limitations. Boyd and Forster [19] described the timestamp interpretation challenges associated with Microsoft's Internet Explorer web browser and time zone translations between UTC and local time. In their paper, Boyd and Forster described a case study where examiners were wrongly accused of tampering with computer evidence based on misinterpreted timestamps.

When discussing the analysis of timestamps, it is important to acknowledge that not all system clocks are always accurate. Since system clocks are based on a low frequency CMOS transistor, the clock drifts over several charging and discharging cycles and a 1 second count no longer

remains at exactly 1 second. This can lead to two problems: *clock drift* and *clock skew*. Clock drift is when the system time continues to drift at an undetermined rate out-of-synchronization from a reference clock, due to the clock “ticking” either too quickly or slowly. Clock skew is the cumulative effect of clock drift at a singular instance in time, which results in the clock being offset from the reference time. Schatz et al. [171] and Buchholz and Tjaden [23] conducted independent studies on clock skew and clock drift to determine exact time from recorded system events. The conclusions from these works indicate that system clocks lose time non-linearly and no single, simple model can be applied to correct this. Clock skew and drift are beyond the scope of this thesis and will not be discussed further.

2.4.2 Causal Ordering of Events

Lamport [105] characterized causality in distributed systems as a “happened before” function on events (called the clock consistency condition) and presented a framework for reasoning about partial event ordering in distributed systems. Gladyshev and Patel [76] formulated the event time-bounding problem and proposed the sandwich algorithm for solving it when the causal order is known. The algorithm attempts to time bound an event between the smallest interval defined by predicate $T_{\min}^B \leq \text{timestamp}_{\text{event}} \leq T_{\max}^B$ when the event’s causal relationship is known with respect to other events whose timestamps are available. Willassen [200] proposed a similar method using hypothesis based testing on timestamps to detect antedating.

Stevens [184] proposed the unification of timestamps from different sources by accounting for factors affecting the behavior of system clocks with respect to a global clock. He proposed a global clock model to account for clock drift and skew and simulate the behaviour of each independent clock. The clock models were used to remove the predicted clock errors from timestamps to obtain a realistic indication of the actual time at which the corresponding events occurred. All the timestamps from different sources can then be unified using this global clock model into a single time-line. In order to be able to unify all the digital events, two sets of information are required. Firstly, one needs to identify all the different clocks that were used and which time stamps were produced by each clock. Secondly, one needs to know the complete behaviour of each clock over the relevant time period. It is also necessary to have a full understanding of how time stamps are generated and their semantics.

Stevens’ work identified the need for a global reference and addressed the problem of clock skew in unifying timestamps. Although Stevens’ model assumes a global reference for the timestamps,

the locality of timestamps is often lost if the design does not retain time zone information. Stevens validated the model on NTFS timestamps; however it is not readily applicable to the FAT32 file system or even ZIP file formats which do not record time zone information.

2.4.3 Timestamp Representation Across Systems

Microsoft [120, 121, 123, 124] documented the fact that the FAT and NTFS file systems have different time-references. Further, FAT file systems only record timestamps to the even second [120] while NTFS systems record up to nanosecond intervals. Consequently, when a file is copied across file systems, its timestamps undergo some changes [126, 127]. Koen and Olivier [100] discussed the information deficiency problem and the use of file timestamps from a UNIX file system in digital forensics. Chow et al. [46] proposed a method for systematic evaluation of timestamp behavior on the NTFS file system.

One of the challenges in using timestamps is their interpretation owing to varied semantics across different representations. For instance, timestamps are allocated 4 bytes on the FAT32 file system, 2 bytes each for DATE and TIME [201]. On other FAT file systems, such as FAT12 and FAT16, fewer bytes are allocated and hence, time is less precisely represented. On NTFS file systems, timestamps are represented as 64-bit values. In UNIX, timestamps were represented as 32-bit values but have recently been changed to 64-bit values in LINUX based systems.

When files from these file systems are archived into one of the many archiving formats, these formats dictate which timestamps get carried forward. For instance, the ZIP file format, which is a popular archiving format, stores timestamps as 2 + 2 bytes for DATE and TIME [96]. When multiple timestamps are recorded on a file system, such as creation, modification and last-access, the ZIP format only carries forward the modification timestamp forward dropping the remaining timestamps. As a result, timestamp precision often suffers, for example, when files from NTFS are transferred to FAT32 [126].

These diverse timestamps representations can have a significant bearing on the semantics and interpretation of their values which in turn will affect their sequencing to generate a causal ordering. This can often result in ambiguous or inconsistent timelines, particularly across heterogeneous sources of digital evidence. To address this problem, my work develops a provenance model to provide uniform interpretation across heterogeneous systems and develop a unified timeline during analysis. My model is presented in Chapter 4 of this thesis.

2.5 Chapter Summary

In this chapter, I reviewed the process of digital forensics and discussed related research and how the literature classifies the field into four main categories: digital forensic process modeling; evidence acquisition and representation; evidence examination and discovery; and digital forensic analysis. I discussed each of the categories highlighting relevant contributions and interesting challenges.

Based on my study of literature, I identified the following as the gap in technology that motivated my research:

1. There is general lack of cohesiveness in the use of forensic and analysis tools in the face of the heterogeneous nature and the growing volumes of digital evidence. This motivated my research on objective one stated in Chapter 1.
2. The results from the analysis of one or more tools do not yet lend to integrated analysis requiring significant manual effort to establish corroboration. It requires the identification of associations in digital evidence and grouping the associated elements in a manner that is forensically productive. This motivated my research on objectives two and three stated in Chapter 1.
3. The heterogeneity of digital evidence has significantly challenged the ability to generate unified timelines across multiple sources, often leading to ambiguous or inconsistent timelines. This motivated my research on objectives one and two stated in Chapter 1.

From my review, I elicit that metadata are an important part of digital artifacts and they lend a valuable hand during forensic investigations. The literature tells us that classification and filtering are two methods used to understand yet unknown data where the grouping can yield a determination of patterns based on value matches. However, in regard to forensics, it is necessary to discover and examine other higher level associations to answer the questions pertaining to an investigation. This necessitates the identification of all associations between the digital artifacts leading to the discovery of relationships and sequencing. While filtering and classification are single-stream, or single parameter based, metadata, which can be treated as sets of name-value pairs, they can provide the dynamics that I desire. Identifying metadata value matches and similarities can result in metadata associations that can inform us of the existence of such higher-order associations leading to the discovery of relationships in digital evidence.

Encouraged by this common thread, in my research I focused on developing a metadata association model based on matching metadata values across digital artifacts that result in association based aggregation. To investigate the implications of this approach, I apply my approach to analyzing two types of files, commonly examined during investigations, and study the relationships that emerge on multiple collections of digital images and word processing documents. The following chapter outlines the research method adopted in this thesis.

“You know my method. It is founded upon the observation of trifles.”

- Sherlock Holmes

3. Research Method

In this chapter, I describe the research method adopted in this thesis to achieve the research objectives set out in Chapter 1. My research objectives focus on the development of a model to identify association among digital artifacts from heterogeneous sources to elicit the higher-order relationships that may exist among them, via the metadata value matches.

3.1 Research Methodology

The overall approach to my research was to adopt the iterative method [49, 61] outlined in Section 1.5. The very first step in the *first iteration* of this method was the definition of the research problem, i.e., identification of metadata based associations and unconstrained combinations of metadata in digital artifacts from heterogeneous sources of digital evidence. The second step was the identification of sub-goals or specific research objectives which arise out of the research objectives stated in Section 1.4. Excepting the first research objective, which involved a survey of contemporary forensic and analysis tools and their treatment of metadata, the other two research objectives were achieved by applying the iterative method using a grounded theory approach [188], wherein I develop the model for identifying metadata associations from heterogeneous digital artifacts based on empirical data. Each research objective was broken down into multiple research tasks with well-defined goals. The completion of all the goals associated with each objective led to the successful completion of the overall objective. This strategy is shown in Figure 3.1, as an instantiation of the general research method presented in Figure 1.4.

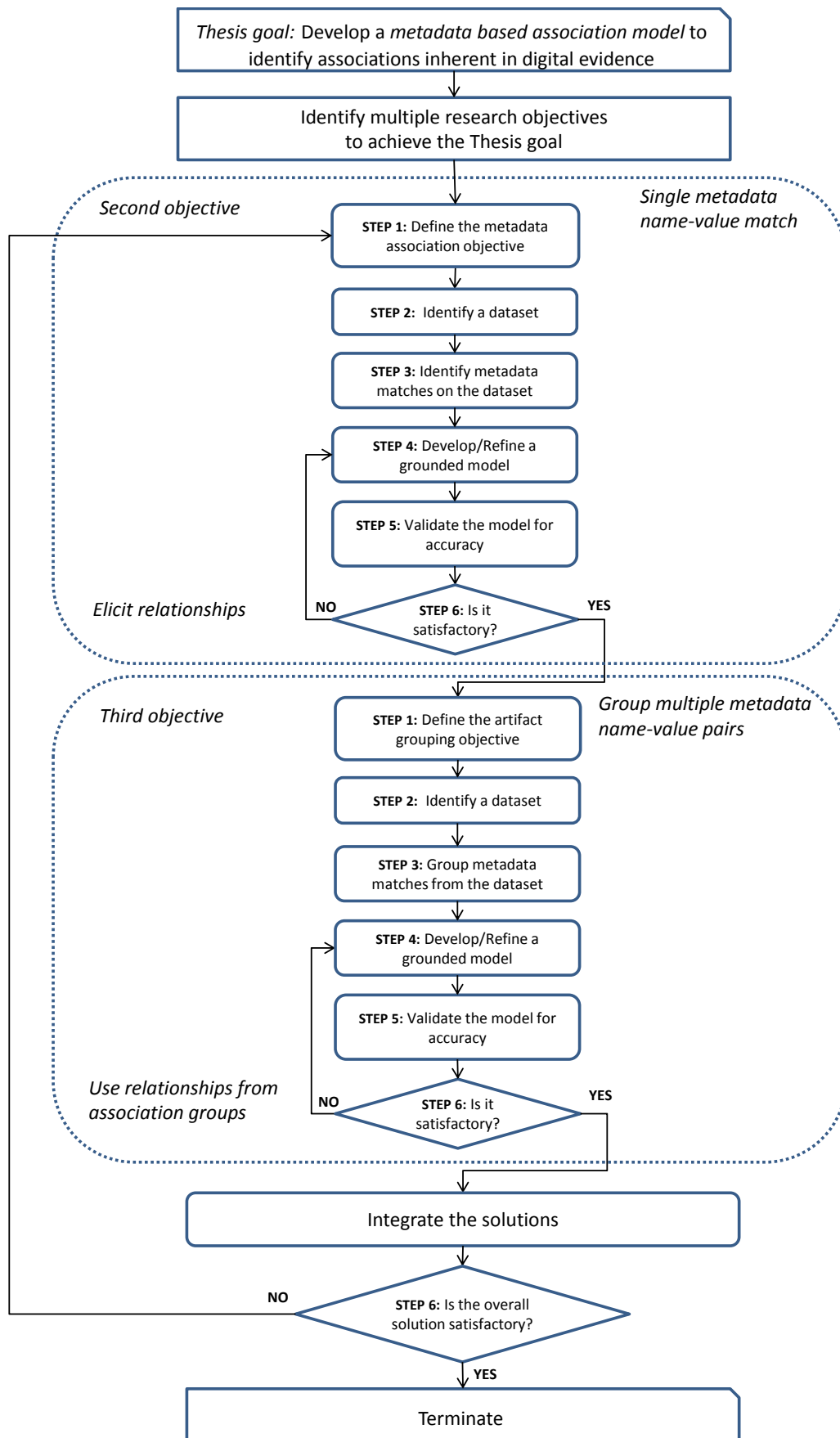


Figure 3.1 How my large research problem was broken down into smaller research objectives, each dealt with in sub-iterations

The first research objective involved an understanding of contemporary forensic and analysis tools in their treatment of metadata from digital evidence. It is common knowledge that file system metadata is extracted from forensic images of hard disk drives to validate the files and timeline the activities on that source of digital evidence [19, 22]. However, it was necessary to understand how the availability of other types of metadata, particularly the application metadata, can be used. Beside files, logs and network packet captures also contain attributes that provide valuable book-keeping information regarding a log record or a network packet, like timestamps, which are useful during analysis. It was necessary to identify the specific support extended by present-day tools in this regard. Besides, it was also necessary to gain an understanding regarding the functionalities of the forensic and analysis tools in regard to grouping the files. These requirements motivated us to use a survey method in undertaking this objective. I developed a hypothesis based survey; each hypothesis was tested in order to draw conclusions on the functionalities of the tools identified.

To define a metadata based association between two or more digital artifacts, I needed to take cognizance of the existing types of associations exhibited by digital artifacts. To develop this understanding, it was essential that I used those digital artifacts which contained a wide variety of metadata, which can be built into my model. In that regard, I identified files as the common ground for developing this model since files are well-understood in the literature [26] and contain metadata which can be directly attributed to forensic contexts [22]. Figure 3.2 illustrates the various stages in my research method.

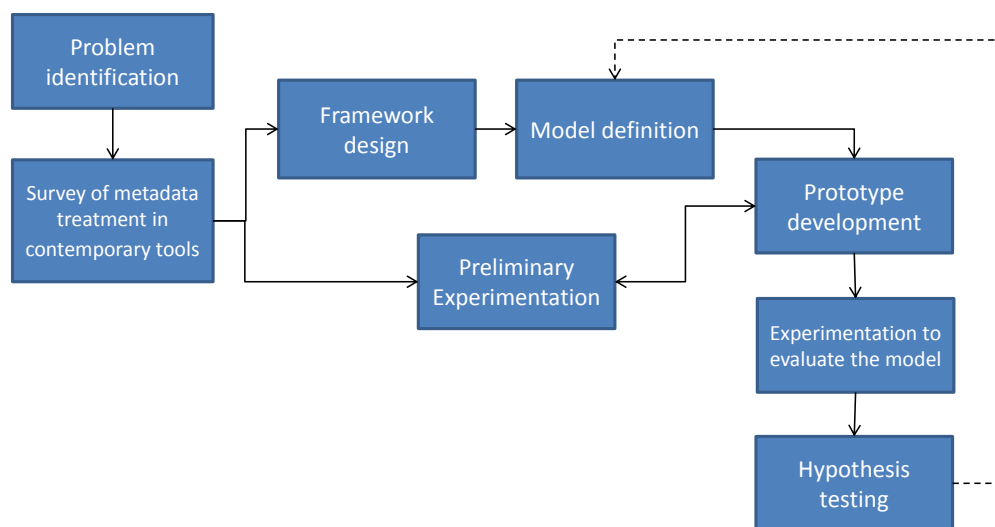


Figure 3.2 The research method applied in my research

Digital forensics, as a field largely built on explaining the nature of stored digital data and reconstructing the past, is founded in a grounded theory. Kessler [98a] argued for the need to develop grounded approaches to interpreting digital evidence. Jeyaraman & Atallah [91], Arasteh et al. [7] and Wang & Daniels [197] have also used grounded theory to reconstruct event sequences from logs and network packet traces. Besides this, the nature of forensic analysis can be qualitative [39, 32] which justifies the use of inductive argumentation to develop techniques that suit this cause [186]. As a result, I adopted the grounded theory [188] approach to identify existing associations and develop an association model that explained the observed associations. I outline the nature of the research objectives in regards to the methodology and identify the research tasks in the sequel.

3.2 Identifying the Research tasks for the Objectives

We have outlined the three major research objectives for this thesis in Chapter 1. my first research objective was to develop an understanding of the treatment of metadata in the digital artifacts and the grouping techniques used by contemporary forensic and analysis tools across heterogeneous sources of digital evidence. The nature of the task required a survey of existing forensic toolkits and architectures for handling different sources of digital evidence and access the metadata from their digital artifacts. I adopted a survey method and it involved the following research tasks:

- a. Identify a list of contemporary forensic toolkits, forensic examination tools and artifact analysis tools for the commonly occurring sources of digital evidence.
- b. Identify the methods used by the selected tools for extracting digital artifact metadata.
- c. Identify the methods used by the tools for grouping the digital artifacts based on the extracted metadata for the purpose of analysis.
- d. Determine if the tools can be configured to use multiple metadata for grouping the digital artifacts. If so, how? If so, is the configuration programmable?
- e. Develop generic abstractions for the functionalities of the tools in handling different sources of evidence at (i) binary data level; (ii) digital artifact abstraction level; (iii) metadata abstraction level; and (iv) digital artifact grouping level.

- f. Develop a functional architecture based on these abstractions to integrate these tools.

The nature of these research tasks was sequential that required one or more hypothesis based tests. Each chosen forensic or analysis tool was subjected to these tests using a test image identified for this task (refer to Chapter 4, Section 4.1). My research tested the tools using raw and logical forensic images, web browser log files and network packet captures. My hypotheses were predominantly binary in nature requiring answers of the type YES/NO, but often delving deeper using the same approach when a YES was determined. When a NO was determined, I proceeded to the next criterion or tool, as the case indicated. This is discussed further in Chapter 4.

To achieve my second research objective of developing an understanding for a metadata association between any two digital artifacts, it was necessary to distinguish the different types of associations that can exist depending on the artifacts involved. Between two digital artifacts, we may find

- a. a value match on corresponding metadata; and
- b. the semantics of the metadata as interpreted for digital artifacts that are deemed to be associated based on a metadata value match.

In order to develop this understanding further, I identified metadata value matches between two digital artifacts to ascertain the existence of higher-order associations relevant to an investigation (refer to Chapter 4, Section 4.4). I adopted a grounded approach to develop a solution and evaluate the nature of syntactic metadata associations between files on a file system. To this end, I identified two kinds of file types, viz., digital image files and word processing files, for conducting my experiments. These file types, in addition to containing a variety of metadata, are frequently encountered by forensic examiners during investigations. My approach to achieve the second objective involved the following research tasks:

- a. Identify a dataset on the file types of choice and identify all metadata value matches between the files.
- b. Develop a grounded model for syntactic metadata associations based on the matches identified for the files.
- c. Evaluate the model on a different dataset and measure its accuracy.

- d. Identify and define a set of metadata equivalences between files, logs and network packets.
- e. Develop a case study to justify the identification of metadata associations between files, logs and network packets.
- f. Develop a grounded model for semantic metadata associations based on the matches identified for the digital artifacts.
- g. Evaluate the model on a different dataset and measure its accuracy.
- h. Refine the models (repeat steps b-g) until the accuracy is at the required level.
- i. Develop a formal representation for metadata associations based on the models developed for any two digital artifacts.
- j. Evaluate the satisfaction criteria. If unsuccessful, identify the deviation and iterate to Step *b* and refine the model.

The nature of these research tasks was experimental and iterative. I identified metadata based matches, maintaining matches as the criterion for association between files, and grouped the files containing identical values for each metadata separately. Based on the groupings so formed, I developed a model to represent the associations among grouped files across different file types. I undertook similar approaches for log files and network packet captures as well and then integrated the models so generated in each case at the end of my experiments.

To achieve my third objective of grouping the associations using the digital artifacts, it was necessary to understand the requirements of forensic analysis. From its definition, digital forensics is the application of scientific methods to reconstruct the past. This activity involves discovering answers to questions that pertain to the creation, existence, modification and access of the digital artifacts in digital evidence. These questions can be succinctly listed as *who*, *what*, *when*, *where*, *how* and *why* [32]. In my experiments, this grouping based on metadata was used to find answers to the six forensic questions. For instance, questions pertaining to “the when” in digital evidence can be answered by grouping the artifacts based on value matches of those metadata that relate to timestamps. Therefore, achieving this objective involved achieving the following research tasks:

- a. Identify the categories of metadata relevant to a digital forensics investigation, taken from the six forensic questions.
- b. Organize the metadata chosen in step (a), as belonging to ‘source’, ‘ownership’, ‘timestamps’ and ‘structural application metadata’.
- c. Group the artifacts from a given source according to the metadata value matches.

The nature of these research tasks was experimental and iterative. Using the same dataset, I identified metadata associations using value matches. To eliminate repetition of files, I grouped the associated files on metadata (refer to Chapter 4, Section 4.4). I examined each group and mapped the associations in order to answer the forensic questions mentioned earlier. Based on this examination, I derived an algorithm which can be applied to files in my datasets. Then I validated against a different dataset to determine if the same questions can be answered. In situations where there were differences, I learnt from them and modified the algorithm. I applied my grouping strategies initially on files that I used as the dataset for the second objective and extended it to group the digital artifacts belonging to logs and network packet trace. The experiments were conducted for two reasons: firstly, to determine the categorization of metadata with regard to the forensic context as defined by the six forensic questions [32] and secondly, to evaluate my algorithm against a different dataset.

3.3 Evolution of a Metadata based Model

We adopted the scientific method [49, 50, 61, 186] and identified specific problems, where further research was required. I evolved this method by priming it with a solution based on preliminary studies (grounded research). In my work, I used metadata matches to generate associations between files and built an association model, which was experimentally tested on multiple datasets. I evaluated the success of the model by identifying the number of outliers (digital artifacts that should have been associated with other artifacts but were not associated) and refined the model to accurately represent the associations in the dataset.

Our research was completely driven by metadata, i.e., entirely depended on the availability of metadata in the digital artifacts that relate to the forensic questions to illustrate metadata matches which lead to metadata associations. Naturally, I required digital artifacts of the type that exhibit such characteristics wherein the metadata not only stores basic book-keeping information, but also

attributes which can help reveal information regarding its source, ownership and authorship, date and timestamps, content type, size and other attributes.

While there are many forensic disk images to conduct such tests, most datasets dispense with the creation of file metadata as it is not used by contemporary forensic tools (refer to Chapter 4). As a result, these datasets provide only file system metadata where the scope for developing associations is limited. Existing datasets provide limited sets of associations particularly as they focus purely on file system metadata. This includes the datasets available from *Digital Forensics Tool Testing Images*¹⁴ (*DFTT*), *Computer Forensic Reference Data Sets*¹⁵ (*CReDS*), *Digital Corpora disk images*¹⁶, *ForensicKB*¹⁷, *The Honeynet project*¹⁸, and so on. As a result, these datasets could not be readily used in my experiments to evaluate my model.

In the presence of additional metadata, the nature of associations can potentially provide insights into a user's activities discerned using the artifact relationships defined in this work. Incorporating such metadata into the existing datasets was found to trivialize the problem and did not suit my purpose either. As a result, I selected public data sources that provide such metadata and composed five datasets without modifying the data or the metadata to evaluate my metadata association model.

Since I required datasets which exhibit file system metadata as well as application metadata, which together can sufficiently represent the attributes of a file, I therefore developed a custom dataset based on files and used both file system metadata and application metadata in my evaluation. In order to evaluate the quality of the dataset, I designed a distance metric to quantify the effectiveness of the metadata associations using all the metadata that pertain to the four classes of source, ownership, timestamps and application and determined metadata value matches to establish associations. This distance metric is introduced in Chapter 5 of this thesis.

With regard to using file based datasets, I needed to understand the type of metadata that already exists in files and identify the subset of metadata that stores information pertaining to the source, ownership and authorship, date and timestamps, etc. Since my classification of the metadata was generic, it was sufficient to use a readily available dataset for this purpose. Any common workstation and personal computer together can be used provided they contained several files of

¹⁴ <http://dftt.sourceforge.net/>

¹⁵ <http://cfreds.nist.gov/>

¹⁶ <http://digitalcorpora.org/corpora/disk-images>

¹⁷ <http://www.forensickb.com/>

¹⁸ <http://www.honeynet.org/challenges>

each application type. Since the purpose of this study was to identify the subset of metadata that were relevant to forensic analysis, the metadata present in the files needed to be exhaustive and not the file contents themselves. Naturally, I identified such a dataset on which the grounded study was conducted.

In addition to file based datasets, to illustrate heterogeneity in data type, I required sources of evidence which recorded event attributes that can correlate activities on files. Log files readily exhibit these characteristics and their attributes can be mapped to many metadata in files. While there are many different types of log files, I decided to exploit web browser attributes. Activities such as web logins, checking web based emails, and uploading/downloading files to/from the Internet generate browser log records which are described by their attributes. To corroborate networking related events on browser logs, I acquired network traces which provide source and destination IP addresses among other attributes.

The nature of evaluation for metadata associations required that the datasets across sources can be related in an obvious manner so that extraction of these relationships from the metadata associations is straightforward. Most datasets in the literature have focused on digital artifacts of a specific type whereas I had to develop related activity logs on other kinds of sources. Besides this, as discussed earlier, files from existing datasets dispense with application metadata, thereby providing a restricted set of metadata to derive associations from. As a result, I used custom datasets and constructed browsing scenarios using case studies that included files, browser logs and network packet traces. The nature of the case studies included typical usage patterns that linked files with browser logs and network packet traces.

3.4 Experimental Evaluation of Model Prototype

This section presents my experimentation process and an overview of the environment. Besides, this section also outlines the criteria considered for conducting a successful experiment for evolving my research method and developing my prototype toolkit.

3.4.1 Experimentation Environment

Our experiments were conducted by selectively imaging file systems on workstations and isolating the relevant logs. A prototype implementation in software was used to validate my contributions. The software was multi-threaded and a separate thread was spawned to traverse a particular source of digital evidence. The log files and network packet captures were exported into

XML before it was analyzed using the prototype. The workstations I used had 2 hard disk partitions totaling 23.2 GB on the workstation used for constructed scenarios and 77.2 GB on the workstation with normal activity. Both computers were operating under Microsoft Windows 7 with Internet Explorer and Mozilla Firefox web browsers and common Microsoft Office applications installed. In addition, several documents belonging to these applications, and audio and video files were stored on the hard disks of each workstation.

3.4.2 Experimentation Criteria

Before an experiment was conducted, my success criteria were defined by the specific research objective being tested for evaluation. If these criteria were not satisfied by the outcome of the experiment, then the experiment was repeated after revisiting the implementation. The basic criteria to be satisfied for an experiment to be regarded successful are listed below:

1. Were all the digital artifacts and their metadata accounted for?
2. Did the prototype software generate metadata associations on the provided sources?
3. Did each association group contain all the digital artifacts it was expected to contain?

The expected answer to Criterion 1 is **yes**. If the prototype software was unable to generate the metadata associations on the given input, an investigation was conducted with the experimental setup to assess what went wrong in the implementation or the experiment itself. The most common error that was discovered was the prototype software running into a never-ending or infinite loop while trying to extract metadata and appropriate exit clauses had to be incorporated for such cases.

The expected answer for Criterion 2 is **yes**. To satisfy Criterion 2, I identified the actual digital artifacts contained in the sources provided as input to the experiment and enumerated the metadata and their values. This list was then compared against the output generated to determine the satisfaction status. If there were discrepancies, then the criterion could not be satisfied. The most common error that was discovered was the prototype software running into a never-ending or infinite loop while trying to extract metadata and appropriate exit clauses had to be incorporated for such cases.

The expected answer for Criterion 3 is **yes**, if the results generated by my prototype software did contain all the digital artifacts, enumerated as earlier. If there were discrepancies, the absence of the artifacts had to be explained. The most common programming error that resulted in such absences was insufficient rules to account for metadata equivalences across digital artifacts of heterogeneous types. This error is rectified by incorporating these metadata equivalences as rules for generating association groups in the prototype software code which was addressed in the subsequent iteration of the design before re-experimentation. If the subsequent re-experimentation then yielded the expected results, then this criterion is deemed to have been satisfied.

3.5 Chapter Summary

For a given (forensic) context, my methodology was applied to develop a design for experiments which outline experimental procedures. In my case, the experimental procedure applied the associations through metadata to determine related artifacts from multiple sources of digital evidence. The true nature of the associations (that were determined using my software) were ascertained using the specific metadata matches which help in revealing the digital artifact relationships that are latent in digital evidence. Extracting such latent relationships can benefit in the consolidation of multiple and possibly heterogeneous sources. Through the process of consolidation that is achieved through grouping the related artifacts, I also achieve a reduction in the volume of digital evidence. This is illustrated in Figure 3.3.

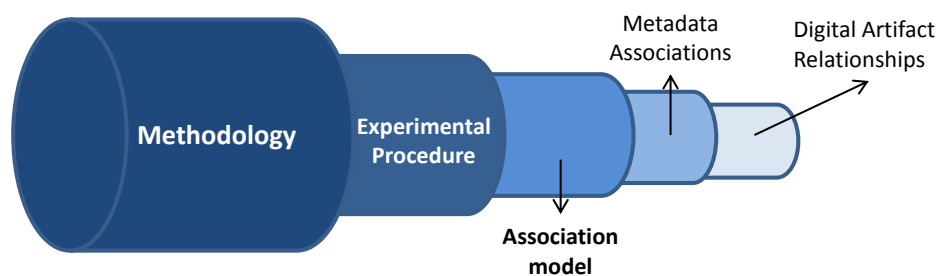


Figure 3.3 Applying my methodology to derive digital artifact relations using metadata associations

In this chapter, I presented my research methodology used to achieve the research objectives set out in Chapter 1. I analyzed the nature of these research objectives and identified a set of research tasks for each objective. I presented my approach to achieving these research objectives and highlighted its relevance to this thesis. I also presented an overview of the evaluation methods that I have adopted in this thesis to evaluate my contributions.

In the following chapter, I present my framework to identify associations among heterogeneous sources of digital evidence using metadata and develop the metadata association model.

"The world is full of obvious things which nobody by any chance ever observes. You see, but you do not observe. The distinction is clear."

- Sherlock Holmes

4. Determining Metadata based Associations in Digital Evidence

In this chapter I develop a new framework for identifying associations in digital evidence based on metadata. While there have been several forensic tools produced to examine digital evidence, the heterogeneous nature of the sources of digital evidence have significantly compartmentalized the use of such tools in analysis, as identified in Chapter 2. There is a general need for integrating the complementary functionalities of the tools used in this regard; to this end, I develop the *functional Forensic Integration Architecture (f-FIA)* and *introduce a new layer* to integrate the examination of heterogeneous sources of digital evidence and determine associations in digital evidence based on metadata. I develop this framework by abstracting the functionalities supported by existing tools and identifying new functionalities to integrate the examination of heterogeneous sources through a hypothesis based review (presented as an experiment in Section 4.1). In addition, I develop *the metadata association model (MAM)* grounded in metadata matches for identifying associations across digital artifacts both within and across sources of digital evidence and grouping them for analysis. I present a review of current forensic and analysis tools to examine digital evidence in the sequel.

4.1 Review of Contemporary Forensic & Analysis Tools

The goal of this review is to gain an understanding of the functionalities provided by existing tools for examining and analyzing digital evidence. In general, there are two classes of tools used to examine digital evidence, forensic tools and analysis tools. The two classes are discussed in the sequel.

4.1.1 Forensic Tools

All forensic tools take a forensic image of a data source such as a hard disk drive or a memory dump as input and provide binary abstractions of the raw data. This allows the entire source to be read as a binary stream of data. In my work, I refer to this functionality as the *binary abstraction*. The tools also distinguish the different files and their application formats on the file systems using standard file signatures [26]. A notable feature of this technology is the development of the *known file filter* (KFF) to omit system files during evidence examination. I refer to the functionality of recognizing files and automatically associating them with their application in order to help parse the file as *file system support*. These two functionalities address the complexity problem in digital evidence [27].

There are several software forensic tools both in the commercial and open domains. The commonly used forensic toolkits for analyzing file systems are Encase, FTK, X-Ways, Nuix, TCT, Sleuthkit, DFF, OCFA, Snorkel and LibForensics. Of these, Encase¹⁹, FTK²⁰ and X-Ways²¹ are commercial toolkits while TCT, Sleuthkit [25], DFF, OCFA, Snorkel and LibForensics are in the open domain²². Among these tools, most commercial varieties also support the examination of memory dumps and mobile device flash memories.

The tools extract file system metadata associated with each file including the location of the file, MAC timestamps, file ownership, file size and so on. Typically forensic tools do not rely on application metadata and consequently do not extract or parse them. To provide that functionality, one may resort to a special set of tools called analysis tools.

¹⁹ <http://www.guidancesoftware.com/encase-forensic.htm>

²⁰ <http://www.accessdata.com/products/digital-forensics/ftk#.UeTux6q6aM8>

²¹ <http://www.x-ways.net/forensics/index-m.html>

²² <http://www2.opensourceforensics.org/tools>

4.1.2 Analysis Tools

An analysis tool directly accesses a data source such as a log file or a packet capture and parses its contents as independent records while guaranteeing read-only access. Each record may contain several attributes which are parsed for analysis. This functionality can also be broadly classified under *schema support*, as part of the *file system support* layer. I term the ability to parse or extract metadata, including file system metadata, application metadata and all related attributes of an artifact in a non-intrusive manner as *metadata parsing*.

There are a wide range of analysis tools for examining files, memory dumps, log files and network packet captures. Some examples of such tools are Volatility for memory dumps, PyFlag²³ for log files and network packet captures, GrokEvt, libevt and Event Log Parser for Windows event logs, AWStats for web browser logs, RegRipper, python-registry, Forensic Registry Editor and Win32Registry for Windows Registry and Wireshark and tcpstat for network packet captures.

Log analyzers such as PyFlag, GrokEvt, libevt, and Event Log Parser parse the respective logs and their attributes. Typically, the attributes in such logs contain an event description, username associated with the event, event timestamp and so on. Wireshark and tcpstat parse corresponding attributes from network packet captures. Network packet attributes can include a packet sequence number, the protocol for communication, source and destination IP addresses, hosts' MAC addresses, hosts' operating systems and browser applications, and so on. Keyword based searching and filtering is used to conduct the actual analysis.

Based on this understanding of the two classes of tools, I conducted a review of such tools to develop an understanding of the support extended by contemporary forensic and analysis tools to examine multiple sources of digital evidence. This review is based on the hypothesis testing method developed by NIST [140, 141].

4.1.3 Hypothesis Based Review

Our approach to reviewing forensic and analysis tools was grounded on hypothesis testing. According to NIST [140], a forensic tool addresses one or more gaps in technology. Therefore, each technology solution is posed as a hypothesis which is then validated by conducting suitable experiments. In my review, the hypotheses concerning the capabilities of the different tools are as follows.

²³ It is notable that PyFlag has since integrated Sleuthkit and Volatility and in that way allows the examination of forensic disk images and the analysis of memory dumps.

1. If a source of digital evidence is provided as input, the tool will successfully load the source;
2. If a source of digital evidence can be successfully loaded, the tool can read binary data from the source;
3. If a source of digital evidence can be successfully loaded, the tool can interpret binary data on the source;
4. If the tool can interpret binary data from a source of digital evidence, the tool can recognize different file systems on the source;
5. If the tool can recognize all the file systems on a source of digital evidence, the tool can identify the individual digital artifacts on a source of digital evidence;
6. If the tool can identify all the digital artifacts on a file system on a source of digital evidence, the tool can extract/parse the metadata from the individual digital artifacts;
7. If the tool can extract and/or parse metadata from the individual digital artifacts that reside on a source of digital evidence, the tool can combine/group multiple digital artifacts based on metadata;
8. If the tool can extract and/or parse metadata from the individual digital artifacts that reside on a source of digital evidence, the tool can combine/group multiple digital artifacts using metadata in an unconstrained manner; or
9. If the tool can extract and/or parse metadata from the individual digital artifacts that reside on a source of digital evidence, the tool can interpret the semantics of the metadata linked to a digital artifact.

In order to test each tool's ability to satisfy these hypotheses, I conducted the following experiment, which was applied to each forensic or analysis tool in turn. I created a forensic (raw) image of a volume partition containing a FAT32 file system and an NTFS file system. The file system contained several files created to mimic regular user behavior on a workstation PC that contained different word processing files (Microsoft Word documents, MS PowerPoint, MS Excel, Rich Text Format files, Adobe PDF files, Text files, and digital images). The files contained both file system and application metadata and recorded events related to activity on the

files. The files mainly contained blank content or random text that weren't intended for use during the experiment. I also created a Windows XP SP2 raw memory dump, a browser history log from a web login session, and a network packet capture from the web login session on the same system used to create the files. My experiment involved the following tasks, each of which corresponds to one of the hypotheses above:

1. Load the source files on a tool; check completion status for each source.
2. Read the first X and last X bytes of the image and print them to the user interface.
3. Redisplay the displayed bytes in hexadecimal and printable text formats.
4. Identify and list the file systems on the image.
5. Identify and list all the digital artifacts on the image. Digital artifacts correspond to:
 - a. files on a file system;
 - b. process control blocks on memory dumps;
 - c. log records on a log file; and
 - d. network packets on packet captures.
6. On each digital artifact, parse/extract metadata from the system as well as the application. This corresponds to:
 - a. File system and application metadata on files;
 - b. Operating system memory maps and process attributes on memory dumps;
 - c. File system metadata of the log file and log record attributes on log files; and
 - d. File system metadata of the packet capture and network packet attributes on network packet captures.
7. Identify methods to group two or more artifacts on the image using metadata using the tool being assessed, if possible.

8. Identify combinations of metadata that are supported by the tool to group two or more artifacts, if possible.
9. Modify a selected file's metadata and replace the values for Author with the number 788755 and file size with the string "Jeffrey". Recreate the forensic image with the new file and load it into the tool. Now extract the metadata and note observations.

Since each of these tasks related to a specific hypothesis, their success corresponded to accepting the proposed hypothesis, and a failure corresponded to rejecting the hypothesis. For instance, if a tool was able to read and print the content, it supported binary data. In addition if the tool was also able to translate the content into hexadecimal and text, it supported interpretation. Listing the file system was applicable only to the forensic disk image, while on the other sources, success was implied by listing the digital artifacts (log records or network packets) on the source. For each digital artifact that was successfully traversed, the metadata was parsed for extraction.

In the literature, keyword filtering and classification are the most common methods to group artifacts on a data source. To evaluate the existing tools I conducted filtering using metadata both explicitly and implicitly; for the explicit method, I used the values assumed by the metadata as specific keywords and for the implicit method, I used the metadata label to conduct searches. I conducted multiple sequences of grouping and regroupings to determine which combinations of metadata were permitted by each tool. In regard to metadata semantics, if a tool flagged an error for replacing expected values on the metadata, then I interpreted the outcome as yes to Hypothesis 9. On the other hand, if the tool did not raise a flag, it implies that the tool is unable to detect inconsistencies in the value's type which indicated that the tool was syntactic by design. My findings are summarized in Table 4.1.

	Digital Evidence access	Digital Artifact Traversal & Examination					Metadata Parsing & Extraction	Evidence Composition using metadata	
	<i>Binary abstraction to DE²⁴</i>	<i>File system examination</i>	<i>Memory dump examination</i>	<i>Log examination</i>	<i>Packet capture examination</i>	<i>Text indexing and Search</i>		<i>Multiple sources of DE (examination and analysis)</i>	<i>Identify correlations</i>
Encase Forensic ²⁵	✓	✓	✓	✗	✗	✓	Only FS ²⁶ metadata	Can group artifacts using FS metadata, one at a time	✗
FTK ²	✓	✓	✓	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time	✗
X-Ways Forensics ²	✓	✓	✓	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time	✗
Nuix Investigator ²	✓	✓	✓	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, and keywords, configurable	Can correlate from specific keywords across content
Sleuthkit	✓	✓	✗	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time	✗
PyFlag	✓	✓	✓	✓	✓	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time, can classify using by combining timestamps across sources	✗
OCEFA	✓	✓	✗	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time	✗

²⁴ DE = digital evidence

²⁵ These are the respective commercial product names.

²⁶ FS = file system

DFE	✓	✓	✗	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, one at a time	✗
Snorkel	✓	✓	✗	✗	✗	✓	Only FS metadata	Can group artifacts using FS metadata, programmable prioritization	✗
Nirsoft log analysis	✗	✗	✗	✓	✗	✓	Log attributes	Can classify using one attribute at a time	✗
GrokEvt	✗	✗	✗	✓	✗	✓	Log attributes	Can classify using one attribute at a time	✗
Libevt	✗	✗	✗	✓	✗	✓	Log attributes	Can classify using one attribute at a time	✗
Reg-Ripper	✗	✗	✗	✓	✗	✓	Log attributes	Can classify using one attribute at a time	✗
Volatility	✓	✗	✓	✗	✗	✓	Memory attributes	Can classify using one attribute at a time	✗
Log2-timeline	✗	✗	✗	✓	✗	✗	Log attributes	Multiple timestamps can be combined for time-lining	✗
Wire-shark	✓	✗	✗	✗	✓	✓	Network packet attributes	Can filter using multiple attributes; classify using one at a time	✗

Table 4.1 The respective functionalities of various forensic and analysis tools

4.1.4 Classification and Grouping of Artifacts

Typically, forensic and analysis tools can classify artifacts using the file metadata or log or network attributes parsed, one attribute at a time [25, 26, 39, 69] as is the case with tools such as Encase, FTK, Sleuthkit or PyFlag. The file owner, username, last modified or event timestamp,

protocol, source or destination IP address are some of the attributes that are commonly used during analysis [19, 22, 23, 28, 32]. However, in order to determine answers to the six forensic questions of *what*, *when*, *where*, *how*, *who* and *why* [32], it may be necessary to conduct deep analysis. When deep analysis is required, the artifacts often may require to be classified multiple times using different attributes to determine the relevant set of artifacts and answer how they relate a particular investigation. This can often entail use of multiple different forensic tools (e.g. Encase, FTK, XWays, Sleuthkit, PyFlag, etc) on the same source of digital evidence and exporting their results for analysis using other tools (e.g. Volatility, GrokEvt, Wireshark, log2timeline, etc.). All commercial forensic tools are monolithic and exporting results from one tool to another can be cumbersome. Therefore, this is a laborious task; this disparity can be more pronounced when the sources of digital evidence span different file and log formats or source types.

4.1.5 Summary of the Review

From this review, I found that all digital forensics tools provide binary abstractions to forensic images to handle forensic images of hard disk drives or memory dumps. While the commercial toolkits may support both file system images as well as memory dumps, most open source forensic tools (e.g. Sleuthkit, DFF, OCFA, etc.) predominantly handle only file system images, albeit in different image formats as discussed in Chapter 2. File systems contain metadata associated with file activity which is independent of file content and forensic tools extract these metadata to identify the owner, MAC timestamps, access privileges and so on. However, forensic tools like Encase, FTK, Sleuthkit, PyFlag etc. do not usually extract or use application metadata from files. All forensic tools support text indexing and searching on an image and classify the artifacts on the image according to the file system metadata. While these tools support multiple forensic images, they do not provide the ability to correlate metadata values across files and alert an examiner when related metadata are discovered (for instance, an identical author name on documents). Besides this, log files which can also be found on many file systems are processed as files by these tools which have to be exported for analysis.

Most analysis tools, with the exception of Volatility or Wireshark, do not provide binary abstraction [41, 195]. These tools interpret the contents and process the data as independent entries while parsing the respective attributes for reporting. Analysis tools also support indexing and query base search; however, they can process only one source (e.g., search the records of a single log file) at a time. This does not permit examiners to conduct analysis to determine

identical or similar entries across sources which can be useful during an investigation. While analysis tools do support classification of the log entries based on the parsed attributes, they do not allow combining multiple attributes to derive semantic relationships. This functionality is becoming a necessity, particularly in the face of the diversity and volume challenges outlined in Chapter 1. Naturally, moving forward, tools should be able to support this functionality and there is a need for architecture that is inclusive by design.

Both forensic and analysis tools group their respective contents using two techniques, keyword filtering and attribute classification. Typically a digital forensic examiner may need to filter the contents based on different keywords or classify them based on different attributes during analysis to determine a pattern. In practice, these techniques are controlled by a human and unless the right combinations of keywords and attributes are specified, the pattern being sought is likely to be missed. Some attributes can also be combined during classification, even if sequentially. The most common way of combining attributes for classification as reported in the literature [17, 18, 113, 195] involves combining timestamps with the owner for forensic images, the *username* for log files and the *IP address* for network packet captures. This leaves the remaining metadata and attributes unused. There is therefore a need for a framework which can identify metadata based associations in an unconstrained manner both within a single data source, like files on a forensic disk image, and across sources, such as between forensic images and logs or logs and network packet captures, and so on.

Motivated by this understanding of contemporary forensic and analysis tools in examining and interpreting digital evidence, I design a functional architecture to integrate the different functionalities of existing tools identified from this review and define a new layer to provide the ability to combine artifacts using associations determined based on metadata.

4.2 f-FIA: Functional Forensic Integration Architecture

The *functional Forensic Integration Architecture (f-FIA)* is illustrated in Figure 4.1 and its layers are as follows:

1. *Digital Evidence Layer*
2. *Digital Artifact Traversal & Metadata Parser Layer; and*
3. *Evidence Composition Layer.*

f-FIA is component oriented and multi-layered (refer to subsections below) and is designed to integrate the functionalities provided by contemporary forensic and analysis tools to examine heterogeneous sources of digital evidence. Besides this, it is also designed to identify associations *within* and *across* sources of digital evidence to conduct analysis.

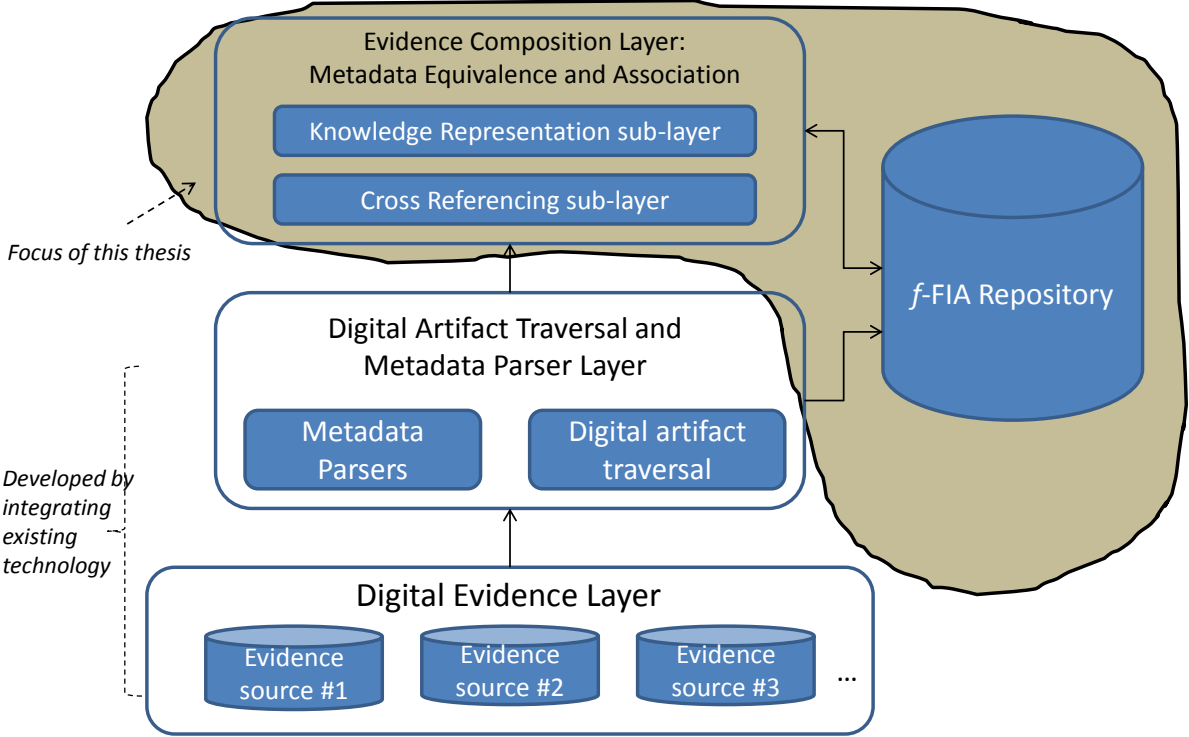


Figure 4.1 Block schematic of the functional Forensic Integration Architecture (f-FIA)

The architecture of *f*-FIA is consistent with forensic principles (maintaining data integrity and read-only access during the examination) and lends itself naturally to automation of forensic examination, while at the same time seamlessly integrating forensic examination with analysis. Its layered architecture is designed to allow scope for future extensions based on technological advances. As my work focuses on identifying associations amongst digital artifacts among sources of digital evidence, I focus on the Evidence Composition Layer and concentrate on methods to group related digital artifacts. I describe the different layers of the *f*-FIA in the sequel.

4.2.1 Digital Evidence Layer

The *Digital Evidence* layer provides binary abstractions of digital evidence sources that are part of an investigation. The media operated by this layer must comply with *read only* semantics to maintain integrity of data during an investigation. The functionality of this layer can be likened to

the binary (and possibly hexadecimal) data support extended by Encase, FTK and Sleuthkit to forensic images such as dd, EWF and AFF and so on.

4.2.2 Digital Artifact Traversal & Metadata Parser Layer

The *Digital Artifact Traversal and Metadata Parser* layer provides access to the artifacts in digital evidence and their metadata. The layer provides appropriate file system and/or schema support to the digital evidence sources for examination. For instance, in forensic file system images, the layer interprets the files, but in logs, the layer interprets the individual log records and in network packet captures, the layer interprets the individual packets. The functionalities of this layer are file system and schema support to examine the files in forensic disk images, processes in memory dumps, log records in log files and network packets in packet captures. Succinctly, this layer is responsible for providing suitable abstractions to the digital artifacts and their corresponding metadata present on each source as well as building indices for the same that can be utilized by the upper layer to determine associations in digital evidence.

In order to parse for metadata, the layer can determine an artifact's application type based on which suitable metadata can be extracted. For example, in hard disk images, files and metadata carry their usual meaning. In log file and packet capture sources, the records and packets take on their attributes as metadata in addition to inheriting the metadata of the log file or network capture file. The functionality of this layer also includes the development of source traversal algorithms and metadata parsers according to the source and the specific application types. The output of the metadata extraction and the indexing process feed into the repository which is then used by the upper layer during analysis.

4.2.3 Evidence Composition Layer

The *Evidence Composition* layer is responsible for integrating information from various sources of evidence and composing the components into consistent and comprehensive evidentiary material to present to a forensics examiner. I achieve evidence integration at two levels; at the first level, by determining related evidence artifacts based on value matches to group them together during analysis and at the second level, by validating the consistency of grouped artifacts to determine relevant evidence. Therefore, this layer is composed of two sub-layers, the *Cross Referencing* sub-layer and the *Knowledge Representation and Reasoning* sub-layer.

The cross referencing sub-layer correlates content and metadata from the digital artifacts in the repository. The repository is capable of supporting data from multiple sources of digital evidence

(digital artifacts, indexed content and metadata). The repository can also support information gathered from external sources (e.g., identity related databases such as social security, bank accounts, driver’s license database, etc.) that are deemed to be relevant to the investigation. The internal architecture of the evidence composition layer is illustrated in Figure 4.2.

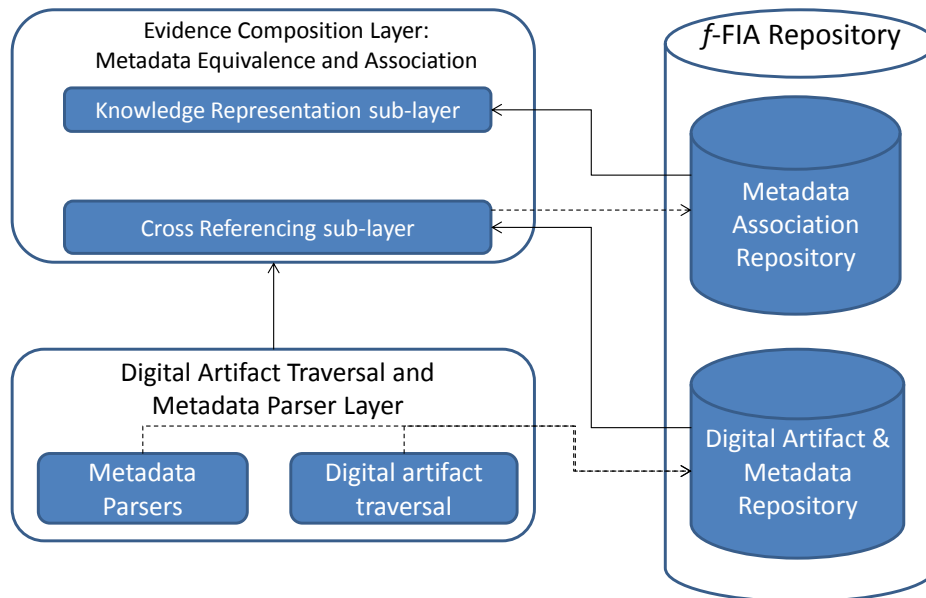


Figure 4.2 Internal architecture of the Evidence Composition Layer

4.2.3.1 Cross-Referencing Sub-layer

The *Cross Referencing* sub-layer is responsible for cross referencing content, including metadata, within and across digital evidence sources. It is the responsibility of this sub-layer to utilize the indices provided by the immediate lower layer to identify associations both on the same source as well as across multiple sources. Since the immediate lower layer abstracts each source by its artifacts and associated metadata, the cross-referencing sub-layer accesses each artifact through its metadata and determines value matches across artifacts, irrespective of the type of artifact. The functionality of this sub-layer is conceived in a technology-agnostic manner, in order to scale across arbitrary sets of digital evidence sources. The resulting sets of associated artifacts are stored in the repository for subsequent analysis. The cross referencing sub-layer can access data on the repository from multiple heterogeneous sources that can be deemed related to an investigation and consists of algorithms that aid in discovering the associations.

4.2.3.2 Knowledge Representation & Reasoning Sub-layer

The *Knowledge Representation and Reasoning* sub-layer is concerned with the logical validity of the digital evidence, based on the associations discovered. This sub-layer is responsible for

determining causal relationships between one or more assertions that can be established based on the artifacts that are associated. Establishing causal relationships between artifacts can help in the identification of relevant evidence for user activity reconstruction as part of the forensic analysis. For instance, consider the evidence that an email was sent by user X with a file attachment. This introduces three distinct predicates as listed below.

1. User X was logged into that system when the email was sent.
2. User X was logged into the email account when the email was sent.
3. The file existed on the system from which the email was sent.

While each predicate can be independently verified based on the evidence available, it may be necessary to identify the associations between the following.

1. The email and the file attachment.
2. The email and the email server logs for user X's email login.
3. The system and the system access logs for user X's system login.

This requires that evidence be considered across heterogeneous sources and associated to establish causation and relevance during an investigation. To achieve this, this sub-layer consolidates the syntactic metadata associations and metadata equivalence relationships across sources to derive semantic inferences on sets of associated artifacts. A few examples are listed below.

1. If digital image files in the evidence match on one or more of their technical metadata, like EXIF metadata, then one may infer that the images were digital photographs captured with the same make and model of digital still camera.
2. If web browser logs indicate a file download whose metadata matches against a file in the user's hard drive, one can infer that the file was not authored by the user.
3. If there are two records in a mail server for the same user at time instant T to indicate simultaneous logins from both Sydney and Melbourne, the information leads to two mutually-exclusive assertions "*The user was in Sydney at time T* " and "*The user was in Melbourne at time T* ".

In the last case, the login attempts in themselves cannot be treated as incriminating evidence. However, the assertions warrant further scrutiny since it is impossible that an individual was in both Sydney and Melbourne at the same time. Resolution of such conflicting assertions requires

that the examiner takes recourse to other (related) external databases – perhaps flight details and passenger manifest databases to determine if the individual travelled between the two cities immediately prior to time T . Correctly ordered timestamps become especially useful in such cases. Validating the correctness and accuracy of timestamps obtained from metadata on digital artifacts including related sources beyond evidence is an example of the reasoning process.

This sub-layer allows recording assertions and validating them by corroborating them within the scope of the data sources provided. Any evidence to the contrary is flagged and presented to the examiner. Besides assertion validation, often examiners need to repeatedly query the sources of digital evidence for information. Examples of such queries are “*list the set of files that were modified on June 10th 2008 between 2:00 PM and 6:00 PM*” or “*list all HTTP sessions on the network capture with IP address X as the source*”. This sub-layer’s architecture enables an examiner to *query* the digital evidence in this way *and determine* evidence associated with the query results *simultaneously*, without having to search for it.

To utilize such a framework, it is necessary to understand the implications of determining metadata based associations in digital evidence and develop a model to represent such associations for analysis. However, when heterogeneous sources need to be correlated for analysis, it is essential to characterize the homogeneity of a single source of digital evidence. This topic is addressed in the sequel.

4.3 Defining a Homogeneous Source of Digital Evidence

Typically, a source of digital evidence is known in terms of where it was acquired from, for example, a source of a forensic image could be a hard disk drive, memory dump, network packet capture file, and so on. These sources have well-defined semantics in the literature as discussed in Chapter 2. However, in this thesis, I require a refinement to this understanding since the digital artifacts and more particularly known by their metadata, which varies greatly across the data types and establishing a relationship between the digital artifacts is not a straightforward task. In this section I develop this understanding further and define what constitutes a homogeneous source.

Forensic images of hard disk drives are a known source of digital evidence. The hard disk drive may contain more than one volume partition containing one or more file systems. In that case, I refine my definition of a *source* to the file system level and distinguish each file system as a source of digital evidence. At this point, the forensic image level definition may be dispensed with

since the analysis is carried out with regard to the file systems. However, each file system, in turn, may contain files belonging to many applications, stored in different formats. Among common file types, the semantics of each metadata is well-understood, e.g., ‘Author’, ‘filename’, ‘file size’, etc; ‘Author’ refers to the author of a particular file; ‘filename’ refers to the name of a file on a particular file system and ‘filesize’ refers to the number of bytes occupied by a file on the file system. This can lead to metadata associations using syntactic value matches. Therefore, I collectively qualify all files belonging to the same application type (identified based on file formats) as a homogeneous source of digital evidence. For my purpose, this definition includes all files created either in temporary or permanent storage by that application. By this definition, all Microsoft Office Powerpoint presentation files, for instance, along with their temporary backup and cache files can be grouped as a single source during analysis. One can extend this argument to state that all files created by software X are collectively referred as a homogeneous source and the semantics is governed by software X. The definition ensures that when metadata based matches are identified between files on the same homogenous source, the corresponding metadata have identical semantics in reference to the respective files. Figure 4.3 diagrammatically illustrates the progression of a source hierarchy with an example of a Microsoft Windows based hard disk drive.

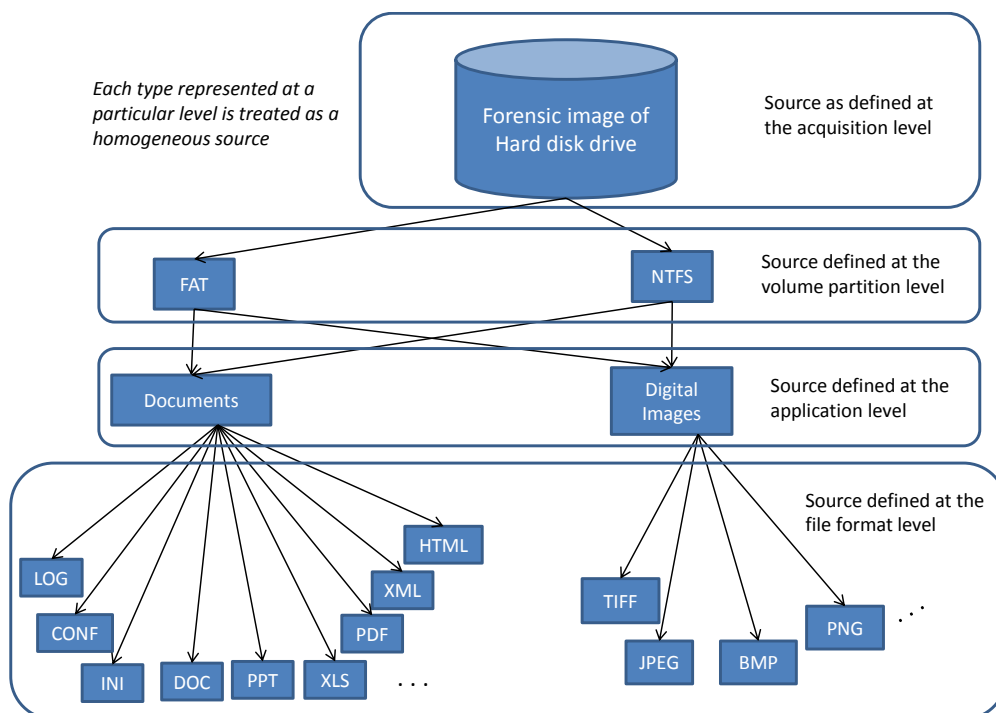


Figure 4.3 Example of source level hierarchy on a Microsoft Windows hard disk drive

In this example, as I refine a source definition at the logical levels, I draw distinctions in applications. Applications, typically record file metadata for file management and files belonging

to the same application tend to store similar metadata, e.g., Microsoft Office documents, PDF files, JPEG files, etc. Naturally, in such files, determining metadata matches can be simplified into a search task pivoting on the corresponding metadata index across all the files and searching for a matching value is trivial. Therefore, files whose metadata can be mapped one-to-one are regarded as homogeneous. Notwithstanding, file system metadata is common to all files and consequently a forensic file system image is homogenous with regard to file system metadata. Where files from the same application are found on different storage media, the choice of the level of abstraction (either at the storage media level as is traditional practice or at the application level to provide uniform semantics across the files) that is appropriate for analysis is left with the forensic examiner. The examiner can exercise their choice depending on the granularity of the analysis that an investigation necessitates.

Log files are of different types and while all log files can be treated as similar sources of forensic evidence, they are not identical. For instance, *syslog* and browser logs are not readily comparable, however all browser logs pertaining to a single browser application are comparable as their log records correspond to some form of browsing event. Naturally, logs describing specific events, such as logs pertaining to web browsing or mail server access, can be labeled homogeneous. Across homogeneous sources, metadata matches can be determined by identifying corresponding log attributes where the values match. The value match encapsulates the syntactic nature of the association across these logs while the activity that is represented by the log itself provides the semantics for the association, i.e., individual *X* who browsed website `www.domain.com` at time T_1 , also browsed website `www.abc.biz` at time T_2 , and so on.

When multiple (heterogeneous) logs have to be compared, attributes such as the *username* or *timestamps* in the respective logs are comparable. In order to do so, these attributes have to be equated and this establishes *attribute equivalence* between the logs. The concept of a value match, as envisioned in *f-FIA* to implement the cross-correlation sub-layer, captures the syntactic nature of the association across these logs while the activity that is represented by the log itself provides the semantics for the association, i.e., individual *X* who browsed website `www.domain.com` at time T_1 , also checked *X*'s email at time T_2 , and so on. Figure 4.4 diagrammatically illustrates the progression of a data source hierarchy using common log files.

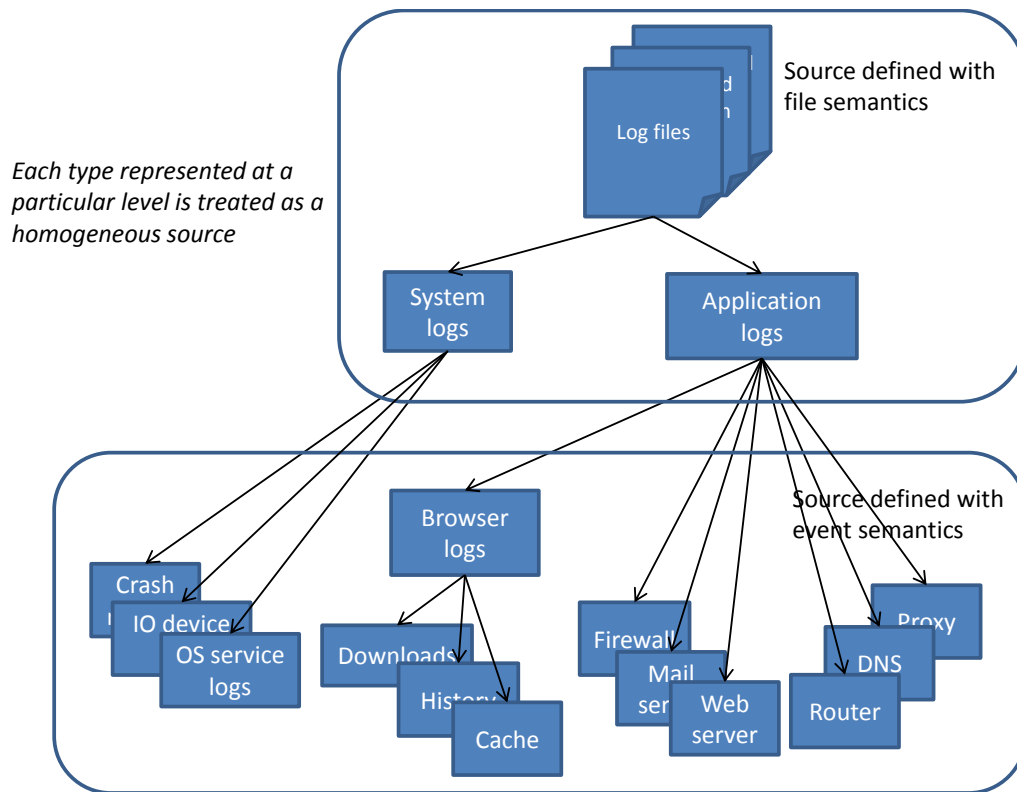


Figure 4.4 Example of the log file data source hierarchy

All network packet captures are regarded as a homogeneous source and where necessary, packets are distinguished based on the protocol used in communication; this distinguishes network packets that were exchanged based on the ARP or the ICMP from those exchanged using the TCP or UDP during a recorded session. These distinctions are necessary to differentiate the parties in communication and the nature of the communication that transpired.

From this, I make the following observation: A *homogeneous source* is one in which all digital artifacts have metadata of the same type (type is governed by the application responsible for creating the metadata) and determining syntactic value matches requires only a comparison operation. The semantics of the association on individual value matches is derived from the semantics of the metadata (elaborated in Section 4.4). Notwithstanding the fact that such digital artifacts may come from physically different acquired sources, in my work I treat such artifacts as a part of a homogeneous source. To develop this understanding, I present *an overview of my approach to discovering metadata based associations in digital evidence* through the identification of metadata matches in value.

4.4 Method for Associating Metadata in Digital Evidence

Metadata matches among artifacts correspond to matching partial states of the digital artifacts. Grouping such digital artifacts based on such a partial state can contribute to reconstructing the past events during an investigation [26, 28]. If such artifacts are grouped together, the “related” artifacts can also be holistically analyzed (e.g., to discover the existence of some higher-order relationships). To ground my research in basic metadata based associations and derive the design for a model to identify associations in digital evidence using metadata, I conducted several pen-and-paper exercises with different types of files and logs by matching the metadata of the files and log records respectively within the same source. An example of this exercise is illustrated in Figure 4.5.

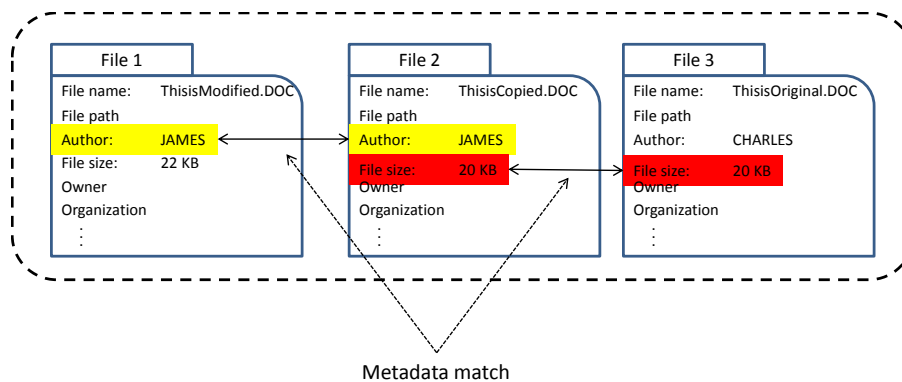


Figure 4.5 Illustration of syntax and semantics associated with a metadata match

Figure 4.5 illustrates how my experiments using metadata matches led to the identification of associations. This example is based on a value match and is hence syntactic in nature. If I consider the metadata shown for File 1 and File 2 in Figure 4.5, the files contain matching values for metadata field ‘Author’ leading to a syntactic association. A similar association exists between File 2 and File 3 for metadata ‘File size’. During my research, I discovered that when the digital artifacts are of the same type, the metadata indices have well-defined semantics between two or more artifacts belonging to the same application. Therefore, between File 1 and File 2, the semantics accompanying with metadata ‘Author’ is interpreted as ‘*James is the author of the file ThisisModified.DOC*’ and ‘*James is the author of the file ThisisCopied.DOC*’. This leads to the deduction “James wrote File 1 and File 2”. Similarly, the metadata accompanying the ‘File size’ is interpreted as ‘*ThisisCopied.DOC is of size 20 kilobytes*’ and ‘*ThisisOriginal.DOC is of size 20 kilobytes*’. This leads to the deduction “File 2 and File 3 have identical file sizes”. The deduction in the first case provides an answer to the *who* question concerning File 1 and File 2 while the deduction in the second case provides an answer to a *what* question concerning File 2 and File 3.

Thus the semantics of a metadata association has the potential to provide answers to questions pertinent during forensic analysis.

As illustrated, at the syntactic level, the representation of the association is based on each metadata and its value, and it concerned a set of digital artifacts (files) that satisfied the condition required to make the association. If a metadata value is changed, it will result in a different match when using the same association semantics. If the metadata's name is changed, then the semantics is changed. I conducted experiments on digital image files and word processing documents using different metadata to generate such metadata-value groups in the context of digital image provenance and identify files suspected of being doctored²⁷. These experiments and their results are discussed in Chapters 6 and 7 of this thesis.

We also conducted similar exercises on browser log files and network packet captures and analyzed the semantics for each syntactic metadata match between log records and network packets of the same type. I generalized the identification of syntactic metadata matches to derive a method for generating associations across artifacts. I am concerned with the identification of metadata associations through the identification of metadata matches, so in my design I explicitly represent the syntactic associations and make the semantics implicit. While combining the artifacts, I used the metadata semantics to guide the grouping and arrived at meaningful associations.

There are four stages of existence for digital evidence ($S_1 - S_4$) and three levels of transformation ($T_{12} - T_{34}$). In stage S_1 , the set U represents all sources of digital evidence, in their natural state, containing heterogeneous digital artifacts. The metadata, related to an artifact, visualized as a vector of metadata indices and its corresponding values, is used as the instrument in determining the transformations $T_{12} - T_{34}$. In stage S_1 , I apply a static filtering transformation T_{12} using metadata matches to group (similar) artifacts into entities called *similarity pockets* that correspond to the metadata tag where the match was determined. The set of all similarity pockets generated based on metadata matches is represented as SP in Figure 4.6.

²⁷ The concept of file doctoring extends the concept of image doctoring wherein the file in question is claiming to be original whereas in reality, it is derived by doctoring another file using one or more software.

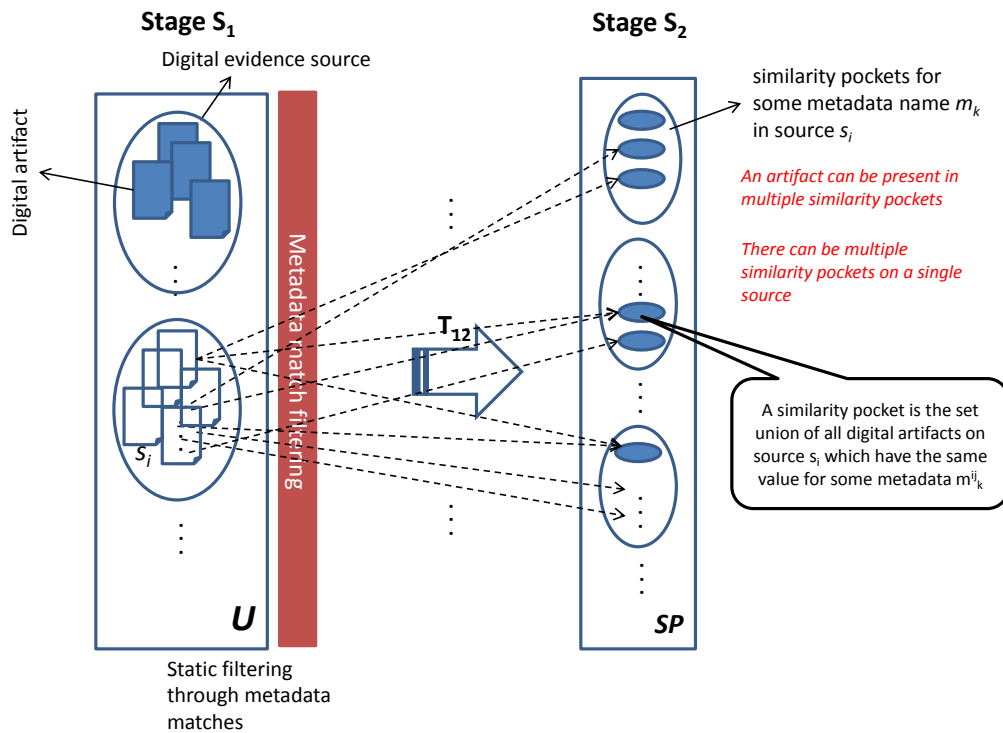


Figure 4.6 The metadata matching stage giving rise to similarity pockets

As a digital artifact can generate a metadata match based on more than one metadata tag, it can be present in multiple similarity pockets for multiple metadata matches in stage S_2 . In order to eliminate this redundancy, I apply a grouping transformation T_{23} to group the digital artifacts from component similarity pockets that overlap on at least one artifact to form *similarity groups*. The set of all similarity groups is represented by SG in Figure 4.7 and are a part of stage S_3 . Once the grouping process is completed, an artifact may occur in only one similarity group and the artifacts belonging to similarity groups on a single source are mutually exclusive.

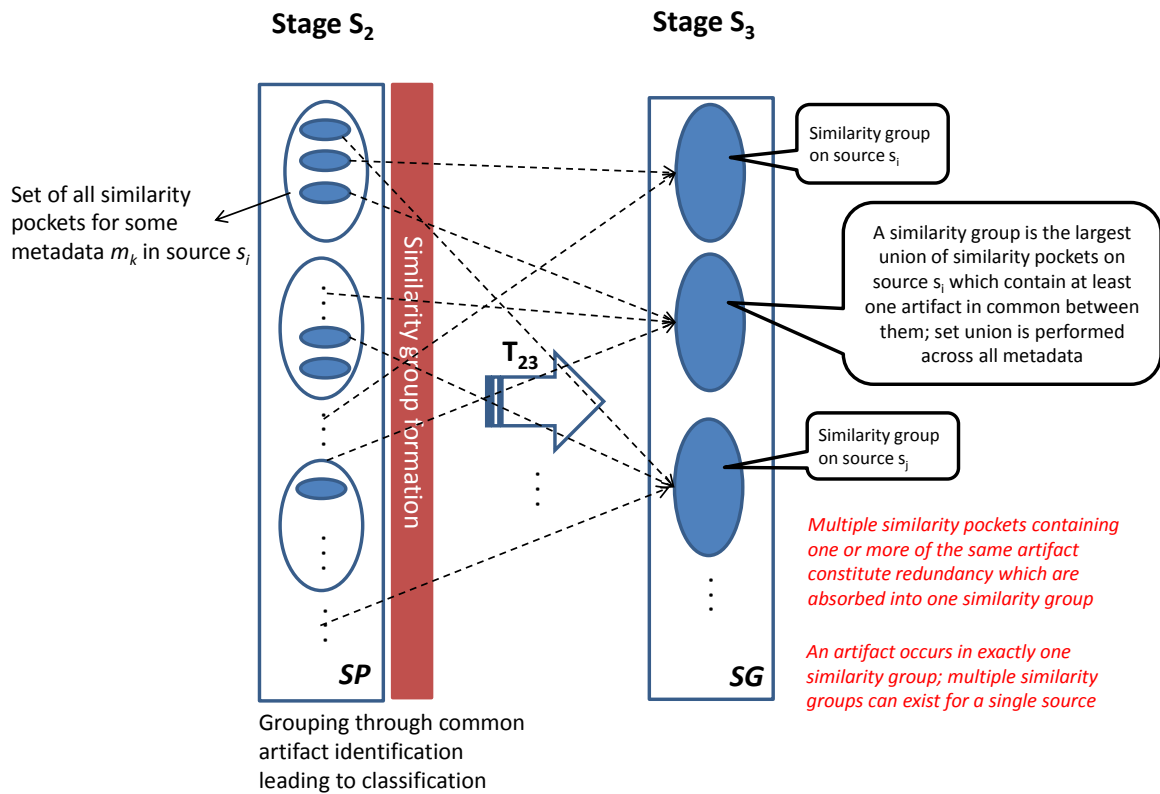


Figure 4.7 Grouping the overlapping similarity pockets into non-intersecting similarity groups

When we consider the metadata values taken by the artifacts belonging to similarity groups across sources, it is possible to establish metadata equivalence relationships on the metadata indices. A simple example of this concept is the equivalence relationships between the file system MAC timestamps and the packet timestamps on network packets from packet captures. A similar equivalence relationship can be established for ‘Author’ on file metadata and ‘usernames’ on log records. We can establish equivalence relationships between such metadata to group the artifacts from similarity groups across sources using transformation T_{34} to give rise to *association groups* in stage S_4 . The set of all association groups is represented by AG in Figure 4.8.

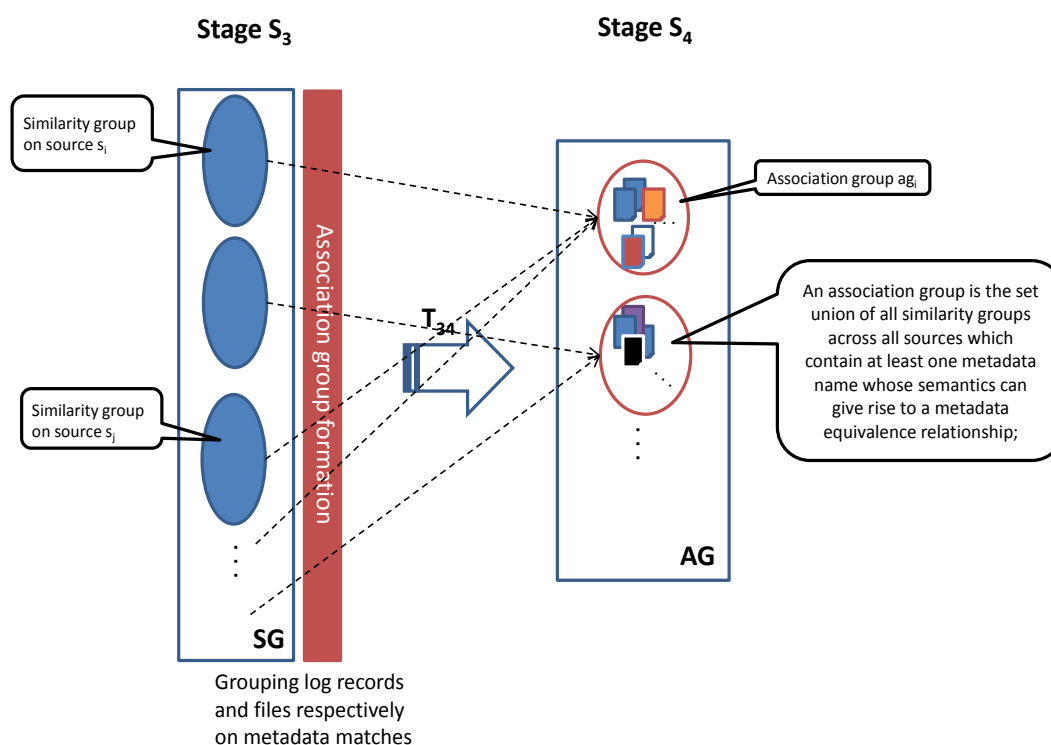


Figure 4.8 Grouping similarity groups across sources into association groups

The members of each association group may be presented to a forensic examiner to identify relevant evidence (as illustrated with the *user X activity example* presented for the Knowledge Representation and Reasoning sub-layer in Section 4.2). The model to identify associations in digital evidence based on metadata is presented in the sequel.

4.5 Metadata Association Model

In this section I define the concept of a metadata association and develop concepts to represent the relationships derived from metadata associations arising out of value matches in metadata. My *Metadata Association Model* (MAM) models the associations in digital artifacts identified through metadata matches. During an investigation, forensic investigators acquire one or more sources²⁸ of stored information which are collectively referred to as *digital evidence*. The digital artifacts that are contained in the collection of sources of digital evidence can be heterogeneous in their application type. If we were to impose a homogeneous view on this world of digital evidence that contain arbitrary types of digital artifacts, let there be S finite and distinct homogeneous sources of digital evidence. Each artifact of a homogeneous source is by definition, homogeneous, i.e., belong to the same application type and contain the same set of metadata, albeit with possibly

²⁸ These are forensic images of hard disk drives, memory dumps, system and application logs and so on.

different values. This method is consistent with my definition of a homogeneous source discussed in Section 4.3. If DE refers to digital evidence containing S homogeneous sources and s_i is the i^{th} homogeneous source acquired for analysis, then

$$DE = \{s_i \mid s_i \text{ is the } i^{\text{th}} \text{ homogeneous source of digital evidence, } i \in [1, S]\}. \quad \dots(1)$$

Each source s_i so acquired can contain one or more digital artifacts and the S sources have $N_1, N_2, N_3, \dots, N_S$ digital artifacts, respectively. In general, a source s_i has its digital artifacts numbered from 1 to N_i . Let each digital artifact in source s_i be referred to as $a^i_1, a^i_2, a^i_3, \dots$ and so on. Each digital artifact a^i_j has metadata associated with it, which I refer to as m^i_j , that describes the artifact. Thus I characterise a source of digital artifacts as

$$s_i = \{a^i_j \mid a^i_j \text{ is the } j^{\text{th}} \text{ digital artifact in source } s_i, j \in [1, N_i]\} \text{ where } i \in [1, S]. \quad \dots(2)$$

To represent a source s_i as the set of all metadata associated with the corresponding digital artifacts that belong to s_i , we can equivalently write Equation (2) in the following way, since there is a one-to-one correspondence between an artifact a^i_j and its metadata m^i_j .

$$s_i = \{m^i_j \mid m^i_j \text{ is the metadata corresponding to artifact } a^i_j \text{ in source } s_i, a^i_j \in s_i, \\ j \in [1, N_i]\} \text{ where } i \in [1, S] \quad \dots(3)$$

Each metadata m^i_j associated with an artifact a^i_j can be represented as an M-vector $(m^{ij}_1, m^{ij}_2, \dots, m^{ij}_M)$ of data values where m^{ij}_k is the k^{th} value for metadata m^i_j and a^i_j belongs to source s_i . Expressed formally,

$$m^i_j = (m^{ij}_1, m^{ij}_2, m^{ij}_3, \dots, m^{ij}_M) \text{ where } j \in [1, N_i], i \in [1, S]. \quad \dots(4)$$

For convenience, all such vectors are assumed to be of the same length, so where a digital artifact has metadata m^i_j such that $|m^i_j| < M$, I append null values to bring it up to size M. The value M is chosen such that it takes the cardinality of the largest metadata vector selected across all digital artifacts, across all sources in DE, i.e.,

$$M = \max_{i \in [1, S], j \in [1, N_i]} (|m^i_j|). \quad \dots(5)$$

Let the digital artifacts in each source $s_i \in DE$ be grouped according to their respective application types. In my research, I represent a source of digital evidence as a list of artifacts, each identified by a list of corresponding metadata obtained both from the file system and the respective

applications. Then, *a metadata match is indicated by matching values in corresponding metadata between two or more digital artifacts*. In this thesis, my model for metadata is a vector of name-value pairs, where the value can be either string or numeric type.

The metadata correspondence can be established in two ways: (i) directly and one-to-one where the artifacts involved in the match belong to the same application and thereby share the metadata semantics; or (ii) through a metadata equivalence established prior to identifying metadata matches between the equated metadata. Where there is a one-to-one correspondence between the domains of values taken by two distinct metadata indices p and q across different sources s_p and s_q , a metadata equivalence relationship can be established between the metadata p and q across these sources.

A syntactic association is assigned to the concerned artifacts when a value match or similarity is identified. The semantics of the association is derived from the semantics of the metadata where the match was found. The semantics of any single metadata associated with an artifact provides an ‘*of*’ relationship with the artifact. For instance, if metadata ‘Author’ for a file F is *James*, then the semantics is derived as ‘*the author of file F is James*’. When multiple files are syntactically associated for metadata ‘Author’, then the semantics of the association is derived as ‘*the author of files F, G and H is James*’. Another way of expressing this association is ‘*files F, G and H have the same author and the author is James*’. When identifying metadata associations, it is necessary to understand the different types of matches in metadata and define them unambiguously. I briefly discuss in the next section the types of syntactic associations that can be defined on metadata that can take string or numeric values.

4.5.1 Types of Metadata Associations

With regard to metadata values, there can be 4 basic types of associations based on value, viz., exact association, partial association, threshold association and date association. These are elaborated below:

Exact association: When a particular metadata value in one digital artifact matches exactly with the corresponding metadata on another artifact, irrespective of the type of value, an *exact association* is said to occur between the artifacts for that metadata.

Partial association: When a particular metadata value in one digital artifact matches partially with the corresponding metadata on another artifact, for a value of STRING type, a *partial association*

is said to occur between the artifacts for that metadata. Such a partial association can be of three different types.

Left sequence: For two strings s_1 and s_2 such that $s_1 \neq s_2$, if two or more characters from the left in s_1 match exactly with the corresponding characters in s_2 , that defines a *left sequence partial association* between s_1 and s_2 .

E.g. $s_1 = \underline{\text{SAMUEL}}$ $s_2 = \underline{\text{SAMSON}}$

Right sequence: For two strings s_1 and s_2 such that $s_1 \neq s_2$, if two or more characters from the right in s_1 match exactly with the corresponding characters in s_2 , that defines a *right sequence partial association* between s_1 and s_2 .

E.g. $s_1 = \text{WILLIAMSON} $s_2 = \text{ROBERTSON}$$

Anywhere in the middle: For two strings s_1 and s_2 such that $s_1 \neq s_2$, if two or more characters in s_1 match exactly with the corresponding characters in s_2 and do not match at either the left or right ends, that defines a *middle sequence partial association* between s_1 and s_2 .

E.g. $s_1 = \underline{\text{INTRIGUE}}$ $s_2 = \underline{\text{CONTRIVE}}$

Threshold association: When a particular metadata value in one digital artifact differs with the corresponding metadata on another artifact, for a value of NUMERIC type, such that the difference occurs within a pre-defined threshold δ , a *threshold association* is said to occur between the artifacts for that metadata. Such a threshold association may occur either with a value greater than or less than the specified threshold. As such, the nature of the difference in value is only relevant, if the artifact on which the comparison is pivoted, is identified.

Date association: When a particular metadata value in one digital artifact, for a value of DATE type, is matched against with the corresponding metadata on another artifact, it defines a *date association* between the said artifacts for that metadata. Such a date association can occur in 4 different types.

At time t: For two timestamps t_1 and t_2 , if their values match to the last degree of resolution that can be determined within technological constraints, then an *at t date association* is said to occur. The value is taken as reference time t .

Before time t: For two timestamps t_1 and t_2 such that $t_1 \neq t_2$, when it is determined that one timestamp is less than the other, then a *before t date association* is said to occur. In this case, the artifact corresponding to the larger timestamp value is taken as reference on which the comparison is pivoted and its value is taken as reference time t .

After time t: For two timestamps t_1 and t_2 such that $t_1 \neq t_2$, when it is determined one timestamp is greater than the other, then an *after t date association* is said to occur. In this case, the artifact corresponding to the smaller timestamp value is taken as reference on which the comparison is pivoted and its value is taken as reference time t .

Between time instants t' and t'': For two timestamps t_1 and t_2 , if we can determine pre-defined time instants t' and t'' such that $t' < t_1, t_2 < t''$, then a *between t' and t'' date association* is said to occur.

Based on the metadata associations outlined in this section, we can group the associated artifacts as discussed in the sequel.

4.5.2 Similarity Pockets, Similarity Groups and Association Groups

In Figures 4.9 and 4.10, nodes represent digital artifacts and the edges represent metadata with identical values. These artifacts, hence, have an *exact association* between them. Figure 4.9 represents such a collection of 5 artifacts associated (numbered counterclockwise) by a single name-value pair match. Since all the digital artifacts have the same metadata index-value pair, it is a *fully connected* graph. For brevity, I have only shown a connected graph. I term such a collection as a *similarity pocket* since there is exactly one metadata match that is shared by all the digital artifacts in that pocket. For instance, a set of 5 documents connected by the metadata ‘Author’ is a similarity pocket. I introduce the concept of a similarity pocket sp_t^{ik} as a set of digital artifacts within a source s_i which have the same metadata value m_k^j for the k^{th} metadata index. Each similarity pocket corresponds to a specific metadata index-value pair and hence, the similarity pockets generated by a particular metadata index k for different values v taken by the metadata index are mutually exclusive. The set of all such similarity pockets formed for a particular metadata index k and value v is tracked using the index t which belongs to the set of natural numbers \mathbf{N} , and each t corresponds to a unique value for the metadata index and value that has resulted in a similarity pocket.

Definition. A similarity pocket sp^{ik}_t is a subset of source s_i , such that there exists a metadata value v and metadata index $k \in [1, M]$, where, for each artifact $a^i_j \in sp^{ik}_t$, metadata value m^{ij}_k equals v . Formally,

$$sp^{ik}_t = \{a^i_j | j \in [1, N_i], m^{ij}_k = v\} \text{ for some } v \text{ where } k \in [1, M], t \in \mathbf{N}, i \in [1, S]. \quad \dots(6)$$

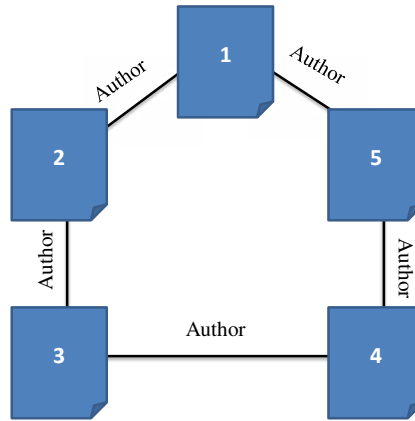


Figure 4.9 Similarity pocket formed among five homogeneous documents on the value of the metadata index 'AUTHOR'

The set of all similarity pockets across all values v for some metadata index $k \in [1, M]$ is denoted sp^i_k and the union of all such sets of similarity pockets across all metadata indices k in the range $[1, M]$ for a source s_i is denoted sp_i .

$$sp^i_k = \{sp^{ik}_t | t \in \mathbf{N}\} \quad \dots(7)$$

= set of all similarity pockets generated for metadata index $k \in [1, M]$.

$$sp_i = \bigcup_{k=1}^M sp^i_k \quad \dots(8)$$

= union over all sets of similarity pockets sp^i_k generated on source s_i across all metadata indices in $[1, M]$.

We refer to the union of all sp_i across all sources in DE as the set SP (illustrated in Figure 4.10), which is defined as:

$$SP = \bigcup_{i=1}^S sp^i \quad \dots(9)$$

$$= \bigcup_{i=1}^S \bigcup_{k=1}^M sp^i_k$$

$$= \bigcup_{i=1}^S \bigcup_{k=1}^M \{sp_{i,t}^{ik} \mid t \in \mathbf{N}\}$$

= set of all similarity pockets generated across all values v for all metadata indices $k \in [1, M]$ for all sources in DE .

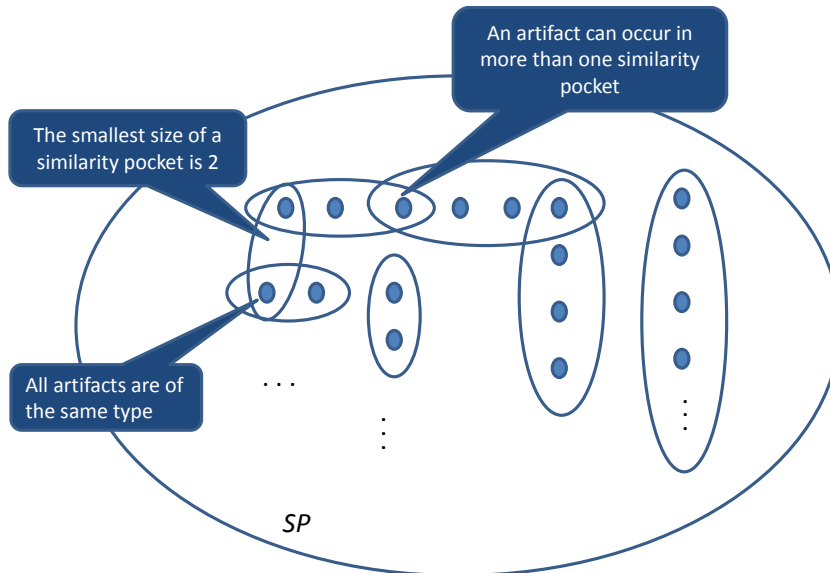


Figure 4.10 The set of all similarity pockets on some source

For the purposes of illustration, consider a scenario where a set of documents on a source are being analyzed. The set of all documents are logically treated as homogeneous artifacts belonging to that source. The abstraction applied in this context refers to the application level abstraction referred to in Figure 4.3. In Figure 4.13, the similarity pocket from Figure 4.9 is extended to a new artifact which is associated with documents 5 and 2 via different metadata. By virtue of the abstraction applied, document 6 is treated as a homogeneous artifact alongside documents 1 to 5. Since this collection is formed by combining multiple similarity pockets, it results in a *similarity group*. A similarity group is the largest combination of two or more similarity pockets within a given source of digital evidence where each digital artifact has a least one metadata match with one or more other artifacts. I introduce the notion of a similarity group sg_i^j , defined as a set of digital artifacts obtained from a union over similarity pockets from the set sp_i where, for each similarity pocket, there exists at least one other similarity pocket which has a non-empty overlap between them. To build a similarity group from the set of similarity pockets, I pick one similarity pocket at random as the seed pocket and iteratively check all remaining similarity pockets in sp_i to determine the similarity pockets that produce non-empty overlap with the seed pocket. The

process is continued until transitive closure is reached. I note that each similarity group that is formed from two or more similarity pockets has the characteristics described by Equation (10).

Characteristics of a similarity group. A similarity group sg_t^i is a subset of source s_i such that it is the largest union over similarity pockets in sp_i where for each similarity pocket $sp_q^{ik} \subset sg_t^i$, there exists another similarity pocket $sp_{q'}^{ik'} \subset sg_t^i$ such that $sp_q^{ik} \cap sp_{q'}^{ik'} \neq \emptyset$.

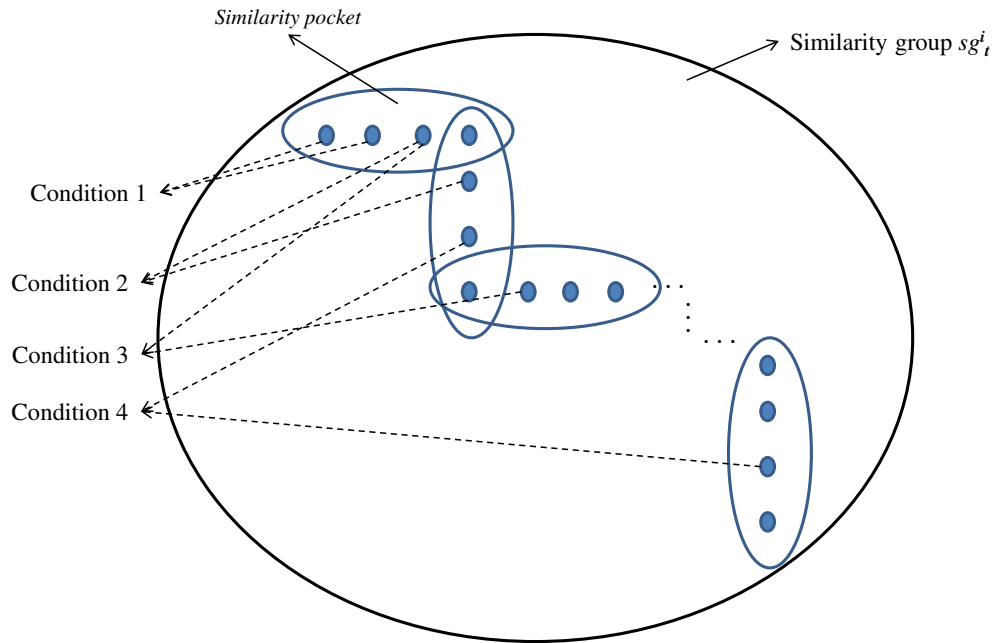
$$\begin{aligned}
 sg_t^i = \{a_j^i \mid k \in [1, M], \\
 \exists q, q': (q \in \mathbf{N} \wedge q' \in \mathbf{N} \wedge \\
 \exists sp_q^{ik}, sp_{q'}^{ik'}, k': (k' \in [1, M] \wedge a_j^i \in sp_q^{ik} \wedge sp_q^{ik} \subset sg_t^i \wedge \\
 sp_{q'}^{ik'} \subset sg_t^i \wedge sp_q^{ik} \neq sp_{q'}^{ik'} \wedge \\
 sp_q^{ik} \cap sp_{q'}^{ik'} \neq \emptyset))\} \text{ where } t \in \mathbf{N}, i \in [1, S]
 \end{aligned}
 \tag{10}$$

The set sg_t^i is illustrated in Figure 4.11 with a set of conditions that govern membership for any two artifacts a and b on source s_i .

$$\begin{aligned}
 sg_i &= \{sg_t^i \mid t \in \mathbf{N}\} \\
 &= \text{set of all similarity groups generated on source } s_i.
 \end{aligned}
 \tag{11}$$

We represent the set of all similarity groups across all sources as SG .

$$\begin{aligned}
 SG &= \bigcup_{i=1}^S sg_i \\
 &= \text{set of all similarity groups generated across all sources in } DE.
 \end{aligned}
 \tag{12}$$



1. artifacts a, b belong to the same similarity pocket.
2. for some artifact c , (a and c) and (b and c) belong to the two respective similarity pockets for metadata m and m' respectively.
3. for some similarity pocket sp in that similarity group, a, b belong to two different similarity pockets such that there are two other artifacts x, y in sp that are associated on different metadata with a and b .
4. for some arbitrary chain of similarity pockets sp_1, sp_2, \dots, sp_n , such that for all k , sp_k and sp_{k-1} exhibit the relationship described in point (2), there exist two other artifacts (x belonging to sp_1 , and y belonging to sp_n) that are associated on different metadata with a and b respectively.

Figure 4.11 Conditions that govern membership to a similarity group

Similarity groups are formed when I merge the similarity pockets across all metadata for all digital artifacts in a source. In order to associate these similarity groups across the sources and determine correlations based on metadata values, I group the sg_t^i where $i \in [1, S]$, $t \in \mathbf{N}$ by establishing metadata equivalence relationships between the metadata indices between two or more sources of digital evidence. The equivalence relationship is established by determining equivalence over the domain of values taken for the corresponding metadata indices between the respective sources.

If similarity groups are combined across multiple sources based on metadata matches established through metadata equivalence relationships, it results in an *association group*. An association group is the largest union of two or more similarity groups in SG where at least one digital artifact

from a similarity group sg_t^i has at least one metadata value match based on a metadata equivalence relationship with another digital artifact belonging to similarity group $sg_{t'}^{i'}$ where $i \neq i'$, $t \neq t'$. To build an association group from the set of similarity groups, I pick one similarity group at random as the seed group and iteratively check all remaining similarity groups in sg_i to determine the similarity groups that contain at least one artifact which has a metadata value match based on a metadata equivalence relationship for some artifact from the seed similarity group. The process is continued until transitive closure is reached. I note that each association group that is formed from two or more similarity groups has the characteristics described by Equation (13).

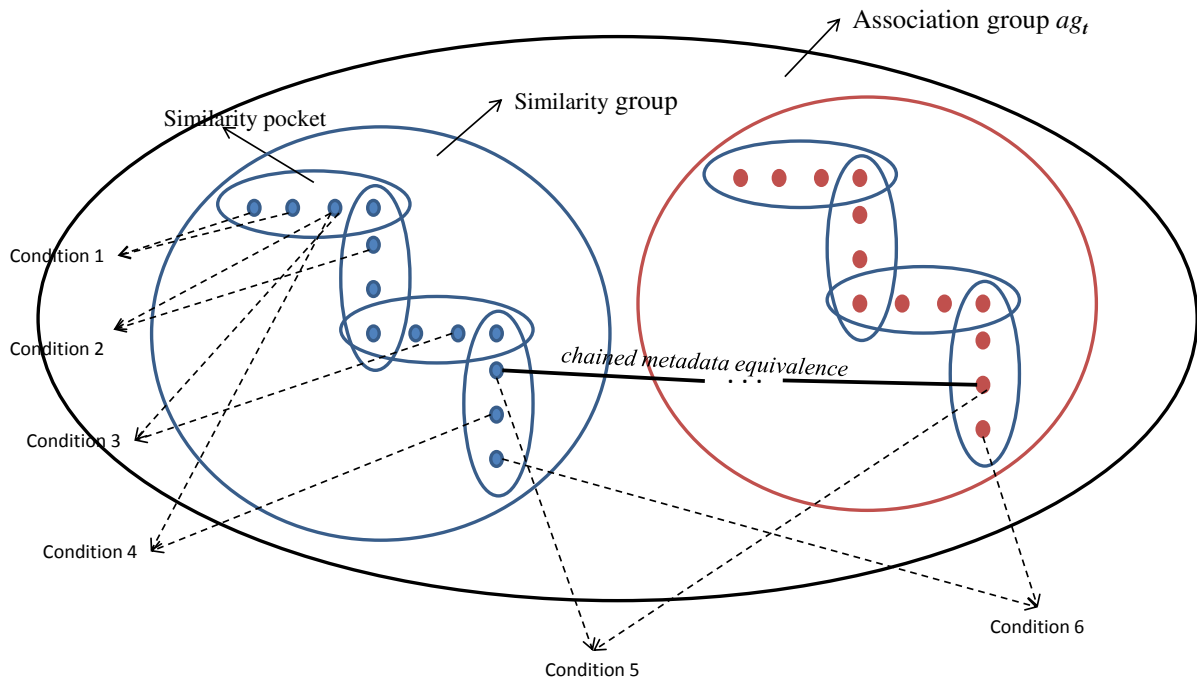
Characteristics of an association group. An association group ag_t is the largest union of similarity groups in SG across sources in DE where metadata equivalence reveals the presence of metadata matches. For each artifact in an association group a_j^i that belongs to some similarity group sg_t^i for some $i \in [1, S]$, $t \in \mathbf{N}$, there exists another similarity group $sg_{t'}^{i'}$ for $i' \in [1, S]$, $t' \in \mathbf{N}$ where $sg_q^i \neq sg_{q'}^{i'}$ and there is a metadata match between artifacts a_j^i and $a_{j'}^{i'}$ based on metadata equivalence.

$$\begin{aligned}
 ag_t = \{ & a_j^i \mid i \in [1, S], \\
 & \exists q, q', i': (i' \in [1, S] \wedge q \in \mathbf{N} \wedge q' \in \mathbf{N} \wedge \\
 & \exists sg_q^i, sg_{q'}^{i'}, a_j^i, a_{j'}^{i'}: (a_j^i \in sg_q^i \wedge sg_q^i \subset ag_t \wedge sg_{q'}^{i'} \subset ag_t \wedge \\
 & a_{j'}^{i'} \in sg_{q'}^{i'} \wedge sg_q^i \neq sg_{q'}^{i'} \wedge \text{artifacts } a_j^i \text{ and } a_{j'}^{i'} \\
 & \text{exhibit a value match based on metadata} \\
 & \text{equivalence})) \} \text{ where } t \in \mathbf{N} \quad \dots(13)
 \end{aligned}$$

The set ag_t is illustrated in Figure 4.12 with a set of conditions that govern membership for any two artifacts a and b .

Let AG refer to the set of all ag_t in DE , then,

$$\begin{aligned}
 AG & = \{ ag_t \mid t \in \mathbf{N} \} \quad \dots(14) \\
 & = \text{set of all association groups generated on } DE.
 \end{aligned}$$



1. artifacts a, b belong to the same similarity pocket.
2. artifacts a, b belong to the same similarity group.
3. for some metadata m and m' on ' a ' and ' b ' respectively, we can establish an equivalence relationship on their values such that metadata value of ' a ' **equivalent to** metadata value of ' b '
4. for some two artifacts x and y , (a and x) belong to similarity group 1 and (b and y) belong to similarity group 2 where x and y exhibit relationship identified in point (3).
5. for some similarity group sg in that association group, a, b belong to two different similarity groups such that there is at least one artifact in sg that exhibits relationship identified in point (4).
6. for some arbitrary chain of similarity groups sg_1, sg_2, \dots, sg_n , such that for all k , sg_k and sg_{k-1} exhibit the relationship described in point (3), there exist two other artifacts (x belonging to sg_1 , and y belonging to sg_n) that are associated differently with a and b respectively.

Figure 4.12 Conditions that govern membership to an association group

In the illustration in Figure 4.13, if the abstraction were drawn at the file format level, then logically document 6 would be regarded as an artifact of a different source and consequently, the grouping would result in an association group (where metadata equivalence is established a priori on the metadata 'filesize' and 'Title').

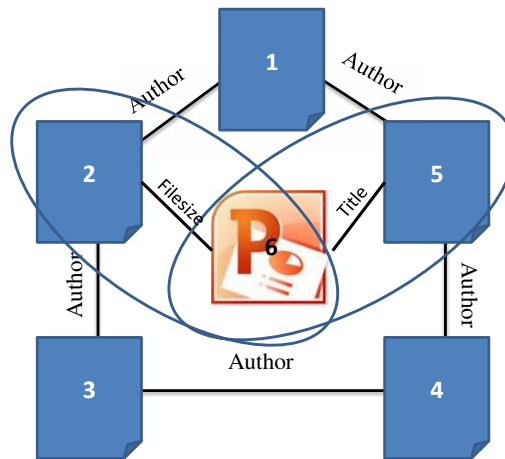


Figure 4.13 Intersecting similarity pockets across different metadata indices among a set of six documents

While documents 1 and 6 are not directly related, they are connected through documents 2 and 5. Thus a seemingly unrelated pair of documents (1 and 6) can be shown to be associated based on metadata. The transitive nature of metadata associations can be useful in scoping keyword searches during analysis as I demonstrate in Chapter 7 of this thesis where I apply my metadata association model to analyze collections of word processing documents. I also show how my model can be used on arbitrary collections of files to automatically identify and group the related files. It is then sufficient for an examiner to study the groups generated rather than examine the individual documents. This is discussed further in Chapters 6 and 7 of this thesis.

During forensic investigations, it is common to identify new sources of evidence and therefore, any model developed for evidence associations should be scalable, incrementally [130]. The metadata association model is so designed that a similarity pocket or association group can be easily extended to incorporate additional artifacts where new metadata matches are discovered from such incrementally discovered sources. When multiple similarity pockets are connected by discovering new metadata matches between them, these pockets are promoted to a similarity group if the metadata associations are limited to a single source or an association group, if they span across multiple sources.

When the digital artifacts from a *collection* (obtained from one or more sources of digital evidence) are “associated” using metadata, I can pose the following questions on the collection for analysis.

1. How many associated and unassociated digital artifacts are present in the collection?

2. How many digital artifacts are singly associated? What is most frequently occurring association on such singly associated digital artifacts?
3. How many digital artifacts are multiply associated? What is most frequently occurring association on such multiply associated digital artifacts?
4. What is the size of largest association group found in the collection?
5. What is the size of the largest multiply associated group found in the collection?
6. What is most number of distinct similarity pockets contained within a single association group?
7. What is the largest number of the associations generated by a single digital artifact?

The numbers determined are inherent to a particular collection. Such questions can be relevant when analyzing sets of digital image files or word processing documents as discussed in Chapters 6 and 7 in this thesis. In such collections of files, the file belonging to the largest association group or the file containing the most number of metadata associations are worthy of further examination.

The theory developed in this section and the definitions presented are revisited in Chapter 5 and is used to drive the implementation of artifact association algorithms that we've developed to determine specific relationships between two or more digital artifacts in digital evidence. I discuss the nature of forensic analysis to derive digital artifact relationships based on metadata associations in the sequel.

4.6 Nature of Forensic Analysis

The aim in the analysis of digital evidence is identification of the events leading to a reported incident, the nature of these events and their attribution to individual(s). For my discourse, an event refers to actions that are directly performed by an individual on any digital device. Examples of such events are creating a file, modifying a file, sending an email, logging into a server, visiting a website, downloading a file, etc. Each event can result in creating new digital artifacts, or accessing or modifying existing digital artifacts(s). Typically the following are observed when a new event occurs on the sources of digital evidence we've discussed in Chapter 1:

1. On a file system, an event can create a new file, or access or modify one or more aspects of an existing file.

2. In memory, an event can create a new process or modify an existing process.
3. On a log file, an event usually creates a new log record. Existing log records are preserved, untouched.
4. During a network packet capture session, an event captures a new network packet. Existing network packets are preserved, untouched.

If a new digital artifact is created as a result of an event, its occurrence is reflected in the metadata that are also created along with the digital artifact. If an existing artifact is modified as a result of an event, its occurrence is reflected in the change in values of the metadata linked to that artifact. Therefore, irrespective of the type of event, its effect can be perceived in the metadata linked to the metadata.

The analysis is concerned with finding answers to the forensic questions that relate to *who*, *what*, *when*, *where*, *how* and *why* [32]. Naturally the process of analysis is driven by methods intended to find these answers. In the previous chapter, I discussed the semantics associated with metadata associations. I explore this further here by extending the semantics to multiple metadata associations and identify metadata classes that naturally provide answers to these six questions.

The most common form of grouping metadata, as reported in the literature [17, 18, 98, 106], is timestamps with owners for files, usernames for logs or IP addresses for network packet traces. The motivation behind this grouping is evident since it helps one find answers to *who* and *when*. To determine answers to *what*, *where*, and *how*, the artifacts are individually analyzed with perhaps, keyword filtering. However, this can be an extended process and may require multiple back-and-forth activities to determine the exact nature of the events recorded in evidence.

When an event creates or modifies more than one digital artifact, identifying the metadata that pertain to the event across these artifacts will elicit the relationships that exist between them. Therefore, focusing on the appropriate metadata across the digital artifacts, one can reconstruct the event(s).

4.7 Applying the Metadata Association Model in a Forensic Context

Our approach parameterizes the artifacts (using metadata) in digital evidence and determines the associations that underline the artifact relationships both within and across sources. In general, the analysis raises several forensic questions, a few of which are listed below. The algorithms corresponding to the questions raised are presented in Section 5.3. Successful application of the

Metadata Association Model to the analysis of digital artifacts can answer the following questions:

1. *How do we apply the MAM to determine all individuals or devices from the digital artifacts in the sources of digital evidence? How many digital artifacts correspond to each identified individual or device?*
2. *How do we apply the MAM to determine the sequence in which the events that correspond to the identified individuals or devices occur?*
3. *How do we apply the MAM to identify those digital artifacts edited with a particular software product?*
4. *How do we apply the MAM to identify those digital artifacts downloaded from the Internet? How do we identify the sources (or URIs) of these digital artifacts?*
5. *How do we apply the MAM to identify digital artifacts that were created or accessed or modified at the same time instant as a given artifact?*
6. *How do we apply the MAM to identify digital artifacts that are identical or structurally equivalent to a given artifact?*

When a forensic examiner examines digital evidence, he/she is actually looking for one or more individuals or digital devices and their activities. Associating an individual or a device to an activity in evidence ascribes the ownership of that activity to that individual or device. Once the activities of an individual or device are identified, it is necessary to chronologically assemble them to study the sequence. Questions 1 and 2 pose this as MAM based problems.

When analyzing digital artifacts, it is necessary to determine those digital artifacts that were edited with software and those that were downloaded from the Internet, not necessarily exclusively. Having identified an artifact as a downloaded resource, it is necessary to determine the origin of that resource from the World Wide Web. Questions 3 and 4 pose this as MAM based problems.

An examiner is often likely to begin the analysis with a single digital artifact and then proceed with the analysis by identifying *related* artifacts based on the outcome from the first artifact. This relationship can exist in time or in structure. When artifacts related in time are sought, it is necessary to determine those artifacts that were affected at the same time instants. When artifacts

related in structure are sought, it is necessary to determine those artifacts that are stored with related structural information. Questions 5 and 6 pose this as MAM based problems.

To determine answers to these questions using the MAM, it is first necessary to identify the metadata in different digital artifacts that are likely to carry information pertinent to the forensic questions raised in this section. In the sequel, I identify metadata families across files, log records and network packets and link them to MAM based analysis of digital artifacts.

4.8 Identifying Metadata Families Relevant to Forensic Contexts

In my work, I am primarily concerned with analysis of files from file systems, log records from log files and network packets from network packet captures. Naturally, it is necessary to determine the classes of metadata from such artifacts that can provide specific answers to the questions raised during forensic analysis. In Figure 4.14, I classify metadata from different sources based on the questions concerning forensic analysis.

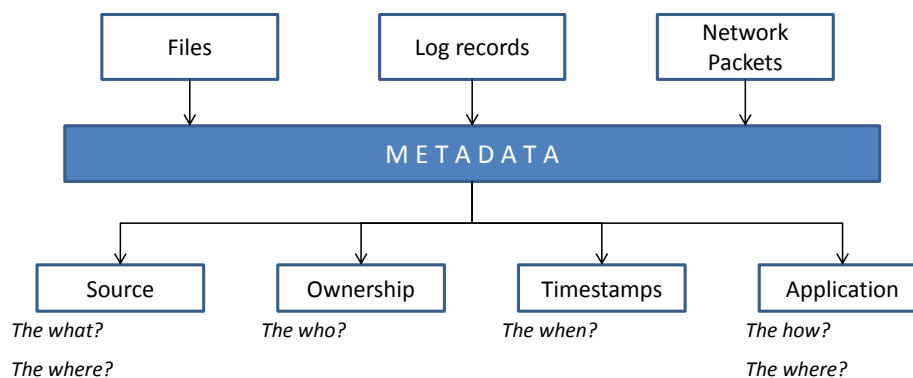


Figure 4.14 Metadata families pertinent to forensic analysis

Typically, questions of the type *what* or *where* relate to the source of the artifact and the metadata that identify such sources are potential candidates for finding the answers. The *who* question identifies an individual who is attributed to an artifact or a system that is attributed to an artifact. The *when* question relates to the time-related event(s) that affected an artifact and the timestamps in metadata can provide such answers. The *how* question pertains to describing other aspects pertaining to an artifact when an event affecting the artifact was observed. Therefore, metadata that identify such situational information are likely candidates. In Figure 4.15, I identify metadata from different artifacts, viz., files, log records and network packets, that belong to the four metadata families and pertain to specific questions in regard to forensic analysis. A mapping of

metadata indices²⁹ from documents, logs and network packets corresponding to each metadata family is illustrated in Figure 4.15.

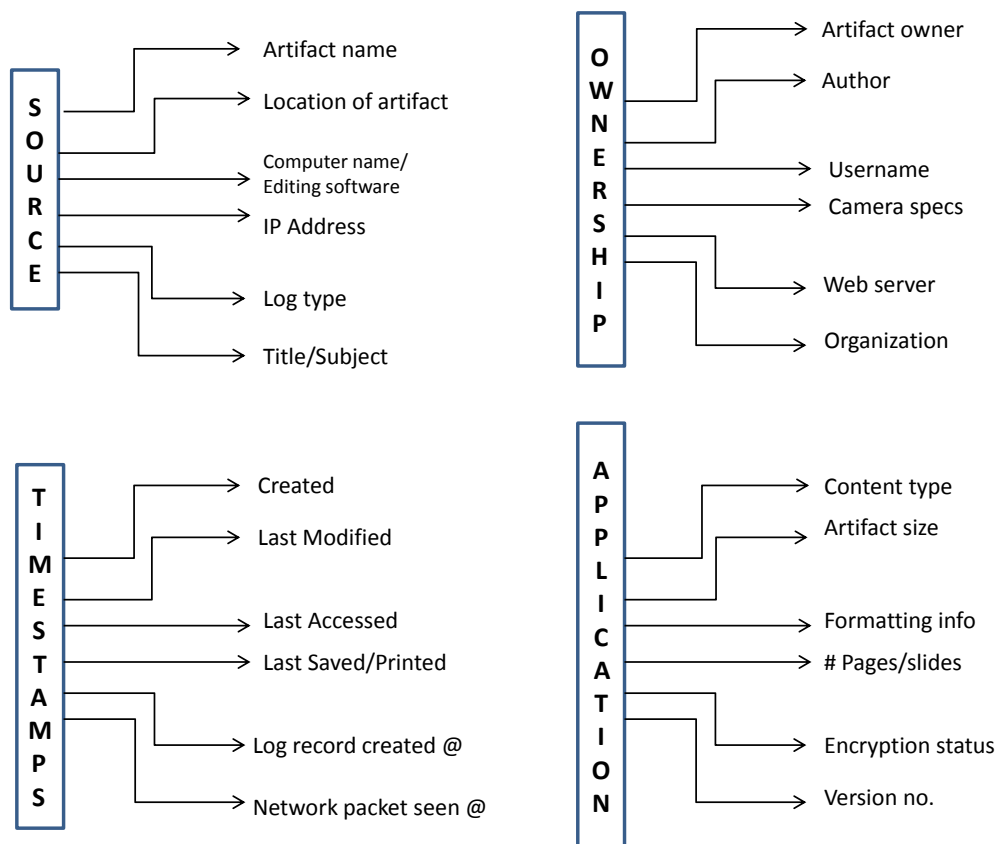


Figure 4.15 Metadata tags for each metadata family across documents, logs and network packets

Source typically corresponds to entities that exist within digital media, such as filename, file location, IP address and Creator/Publisher. *Ownership* corresponds to entities that have some form of physical presence in the real world, i.e., Author, Username, Organization, Digital camera specifications, Web server, etc. *Timestamps* correspond to time instants when digital events are recorded in the digital media and *Application* corresponds to features describing a particular artifact such as its application type, content size, number of pages or slides and formatting. While the metadata describe the characteristics of an artifact, my focus in this thesis is to utilize the semantics related to the four metadata families to elicit associations across digital artifacts.

Table 4.2 shows the nature of grouping conducted by identifying metadata associations in digital evidence. Since the primary method for determining associations is through metadata matches, irrespective of the nature of the sources of digital evidence, if all the digital artifacts are of the

²⁹ This illustration is not exhaustive, simply suggestive.

same type, i.e., homogeneous, then the process is similar to classification where groups are identified based on the values. For instance, all files in a file system, irrespective of the application type, contain some file system metadata. Naturally, when I apply the MAM using file system metadata, files containing identical values for file owner, last modified timestamp, file size, etc. are grouped together and this can also be achieved using a classification technique that pivots on those metadata. However, when a file and a log record are “associated” based on timestamp metadata, it conveys extra meaning in terms of the higher level actions of a user that can be discerned from this grouping. For instance, when a file *f* is grouped with a log record *r* belonging to application *A*, then we could infer that “*application A was used to modify file f*” which can be a valuable inference during analysis. By virtue of the difference in the native representation of these digital artifacts, classification may not be able to identify the inherent relationship³⁰. Therefore, the power of the metadata association model is best taken advantage of when there is inherent heterogeneity in the digital artifacts, irrespective of the number of sources of digital evidence from which they come.

<i>Source of Digital Evidence/Nature of Digital Artifacts</i>	Across multiple Homogeneous digital artifacts	Across multiple Heterogeneous digital artifacts
Across single or multiple Homogeneous source(s)	<i>Classification based grouping</i>	<i>Association based grouping</i>
Across multiple Heterogeneous sources	<i>Classification based grouping</i>	<i>Association based grouping</i>

Table 4.2 Tabulating the nature of grouping conducted across diverse sources and digital artifacts

When we group the associated digital artifacts for forensic analysis, it is necessary to define the semantics related to underlying metadata associations between digital artifacts that pertain to events of interest. In the sequel, I define digital artifact relationships based on the metadata associations for files and log records.

³⁰ This is particularly so if the timestamp in the log record and the file metadata are slightly different. While classification may ignore the chronology in the two artifacts, the MAM will associate them to identify the higher-level event that is being described.

4.9 Deriving Digital Artifact Relationships from Metadata Associations

When we determine metadata associations across artifacts, it underlines the relationship between the artifacts which can reveal the nature of activities recorded. In this section, I define eight types of artifact relationships based on metadata associations to conduct analysis.

4.9.1 Existence Relationship

When a metadata match occurs in the source metadata family for metadata *filename* or *Title/Subject* of the file between files f_1 and f_2 , where f_1 and f_2 reside on different homogeneous sources, I define an *existence relationship* between the files. The files themselves need not belong to the same application type, but only contain the metadata that leads to a metadata association, e.g., .DOC and .BAK or .TMP. The relationship is denoted by R_e and it may be expressed as $f_1 R_e f_2$. By definition this relationship is commutative and associative. The association groups containing such relationship pairs in evidence are referred to as existence association groups. Therefore,

$$1. f_1 R_e f_2 \Leftrightarrow f_2 R_e f_1 \quad \dots(15)$$

$$2. (f_1 R_e f_2) \wedge (f_2 R_e f_3) \Rightarrow (f_1 R_e f_3) \quad \dots(16)$$

When multiple such files ($f_1, f_2, f_3, \dots, f_n$) exhibit an identical association between each other, e.g., produce a metadata match for the same value of filename, I represent this relationship as $R_e (f_1, f_2, f_3, \dots, f_n)$.

4.9.2 Source Relationship

When a metadata match occurs in the source metadata family between files f_1 and f_2 , where f_1 and f_2 belong to the user file system, I define a *source relationship* between the files indicating that the files were likely to be created on the same source as identified the respective metadata. The relationship is denoted as R_s and is expressed as $f_1 R_s f_2$. By definition this relationship is commutative and associative. Therefore,

$$1. f_1 R_s f_2 \Leftrightarrow f_2 R_s f_1 \quad \dots(17)$$

$$2. (f_1 R_s f_2) \wedge (f_2 R_s f_3) \Rightarrow (f_1 R_s f_3) \quad \dots(18)$$

When multiple such files ($f_1, f_2, f_3, \dots, f_n$) exhibit an identical association between each other, e.g., produce a metadata match for the same value of computer name or software, I represent this relationship as $R_s (f_1, f_2, f_3, \dots, f_n)$.

4.9.3 Happens Before Relationship

When a metadata match occurs on the ownership metadata family of log files such as the log records of the web history and cache logs of a web browser, I define a *happens before relationship* indicating the occurrence of a web page visit prior to the download of the specified resource on the cache log. The relationship is denoted by R_h and expressed as $x R_h y$ where x is the digital artifact corresponding to the browser history log and y is the artifact corresponding to the browser cache log. In general, for two events x, y , $\text{time}(x) < \text{time}(y)$ indicating that x happened before y , and the interpretation of this relationship is that event y occurred after event x . This relationship is purely concerned with the associations between events that exist based on the metadata and those that can be practically detected using a deterministic algorithm. Whether these events were causally related or not cannot be decided based on the timestamps alone, and in general, such decisions are to a human forensic examiner. The discussion regarding the causality is beyond the scope of this thesis. By definition, the relationship is not commutative but is associative.

$$(x R_h y) \wedge (y R_h z) \Rightarrow (x R_h z) \quad \dots(19)$$

4.9.4 Download Relationship

When the filename of a file f on the user file system generates a source metadata family metadata match with a download resource r recorded in a browser cache log, I define a *download relationship* indicating the download of the resource r to the user file system. The relationship is denoted by R_d and expressed as $f R_d r$ indicating the creation of f implies the download of resource r .

4.9.5 Parallel Occurrence Relationship

When a metadata match occurs in the timestamp metadata family between two files f_1 and f_2 , where f_1 and f_2 belong to the user file system, I define a *parallel occurrence relationship* indicating that the two files f_1 and f_2 were accessed at the same time identified by the matching value of the timestamps in their metadata. This relationship is purely concerned with the associations between events that exist based on the metadata and those that can be practically detected using a deterministic algorithm. Whether the association leads to the determination of parallelism in the abstract sense is beyond the scope of this thesis and often left to the judgment of

a human forensic examiner. The relationship is denoted by R_{po} and expressed as $f_1 R_{po} f_2$. By definition, this relationship is commutative and associative. Therefore,

$$1. f_1 R_{po} f_2 \Leftrightarrow f_2 R_{po} f_1 \quad \dots(20)$$

$$2. (f_1 R_{po} f_2) \wedge (f_2 R_{po} f_3) \Rightarrow (f_1 R_{po} f_3) \quad \dots(21)$$

When multiple such files ($f_1, f_2, f_3, \dots, f_n$) exhibit an identical association between each other, e.g., produce a metadata match for at least one timestamp on the same value, I represent this relationship as $R_{po}(f_1, f_2, f_3, \dots, f_n)$.

4.9.6 Structure Similarity Relationship

When a metadata match occurs in the application metadata family between two files f_1 and f_2 , where f_1 and f_2 belong to the user file system, I define a *structure similarity relationship* indicating that the two files f_1 and f_2 have identical or equivalent attributes. The relationship is denoted by R_{ss} and expressed as $f_1 R_{ss} f_2$. By definition, this relationship is commutative and associative. Therefore,

$$1. f_1 R_{ss} f_2 \Leftrightarrow f_2 R_{ss} f_1 \quad \dots(22)$$

$$2. (f_1 R_{ss} f_2) \wedge (f_2 R_{ss} f_3) \Rightarrow (f_1 R_{ss} f_3) \quad \dots(23)$$

When multiple such files ($f_1, f_2, f_3, \dots, f_n$) exhibit an identical association between each other, e.g., produce a metadata match for the same value of content type or file size, I represent this relationship as $R_{ss}(f_1, f_2, f_3, \dots, f_n)$.

4.9.7 Unauthenticated Modification Relationship

When two files f_1 and f_2 differ in metadata only with respect to the structural composition of the files and the software exclusively present in only one of the files, it indicates an *unauthenticated modification relationship* denoted by R_{ua} and expressed as $f_1 R_{ua} f_2$. The relationship, by definition, is commutative.

4.9.8 Majority Relationship

When two files f_1 and f_2 have an unauthenticated modification relationship, in the presence of a third file f_3 which contains a source relationship with either f_1 or f_2 , then that pair of files is said to exert a *majority relationship*, denoted by R_m , over the other file. Therefore,

$$(f_1 R_{ua} f_2) \wedge (f_1 R_s f_3) \Rightarrow (f_1, f_3) R_m f_2. \quad \dots(24)$$

When such relationships are determined across digital artifacts on the same homogeneous source, it results in similarity pockets if exactly one metadata match is discovered or similarity groups in the case of multiple metadata matches. Across multiple sources, as in the case of the existence relationship, this would result in association groups. Selecting the right MAM group, one can determine either a single relationship or combinations of multiple relationships as the case may be. Using these eight relationships identified through metadata associations, I show how to discover evidence of files downloaded from the Internet, files likely to be downloaded from the Internet, the source of the download and doctored files. This is discussed in Chapters 6 and 7 in the context of digital image files and word processing documents respectively.

Once the association groups are generated after applying my model, the related digital artifacts can be sequenced on a timeline for analysis. This is achieved by sequencing the timestamps in metadata into a unified sequence. However, since association groups typically contain heterogeneous digital artifacts, possibly across different sources of digital evidence, the timestamp interpretations can be a challenge. In the sequel, I discuss this problem in detail.

4.10 Timestamp Interpretations across Heterogeneous Sources

To determine evidence relevant to an investigation, it is necessary to analyze the artifacts that were created, accessed or modified closer to the time of a reported incident that is being investigated. In such situations, one can set up time windows and analyze the artifacts that were created, accessed or modified within it. Such time windows are relevant only if the artifacts' timestamps are chronologically ordered or digitally time-lined.

Timelines are usually generated using the timestamps recorded in the artifacts' metadata. However, when associating timestamps, the syntactic associations as discussed in Section 4.4 cannot be applied readily. This is because timestamps have varied representations across different systems (refer to Chapter 2). Broadly speaking, there are three types of challenges that can arise when dealing with timestamps from across heterogeneous sources. Apart from basic syntax aspects, there are the following other two types of challenges:

1. Time zone reference and timestamp interpretation; and
2. Clock skew, clock drift and synchronization.

A consequence of these challenges is that a syntactic association need not necessarily lead to a semantic association between the artifacts. For instance, two files with creation timestamps 09:30:00 AM July 25th 2011 AEST and 09:30:00 July 25th 2011 GMT have a syntactic association but may not share a semantic relationship as these timestamps are 10 hours apart. It is necessary to take cognizance of these interpretation challenges when metadata associations are applied to timestamps. In this section, I examine such timestamp interpretation challenges.

4.10.1 Timestamps and Digital Events

We adopt the timestamp definition given by Dyreson and Snoddgrass [55]. Timestamps are an important part of metadata that are scrutinized during analysis. File systems typically record these three timestamps for each file that is stored within it. These timestamps indicate when a particular artifact was created, last accessed or last modified, as the case may be. Timestamps are also recorded on log files and network packet captures and these correspond to the events relevant to the respective logging context. In general, there are many types of events and in my research, I am concerned with five types of events:

1. File Create event: creation of a file in a file system
2. File Modify event: modification of a file in a file system
3. File Access event: accessing a file in a file system
4. Logged event: an event logged by some system or application (e.g., Web server, Internet browser)
5. Packet event: the arrival/receipt of a network packet on capture

The first three events are specific to files on file systems, the fourth event is specific to records contained in log files and the last event is specific to network packets in a network capture file.

4.10.2 Ambiguities in Timestamp Provenance

Different digital sources record events differently and therefore the representations and resolutions of the timestamps also differ. In fact, even if multiple sources were obtained from the same location, the values for their timestamps could differ greatly. For one thing, if the location information where an NTFS or an EXT file system image was found is not recorded, it may be lost forever, since these file systems only record time with respect to UTC. As a result, whether

the timestamp was recorded in Sydney (UTC +1000) at 3:30:00 PM July 1st or in New York (UTC -0400) at 10:30:00 PM the previous night, the timestamp would record a value corresponding to July 1st 5:30:00 AM UTC. Hence if the appropriate provenance of the timestamps is not recorded, despite the time-shift that is applied to the evidence on forensic tools, it can result in ambiguous timestamps which can lead to inconsistent timelines.

4.10.3 Interpreting Timestamps Using Forensic tools

On most forensic tools, the combination of source name and its type is sufficient to determine the timestamp representation format and its time reference. The event determines the specific event name or the file name as the case may be. These timestamps correspond to the MAC timestamps for documents on a file system. On Internet logs such as the history and cache, the semantics varies, but generally timestamps correspond to the last access of a URL (history) or the timestamp on the file system when a resource (represented by a URI) is saved (cache) on the file system. Typically, forensic analysis tools read the timestamps' values and while rendering, apply a *fixed* time zone shift to obtain the UTC (Universal Coordinated Time) value. In the case of timestamps from the NTFS or EXT file system, the timestamps are available in UTC and the shift is applied to obtain the local timestamp. The time shift corresponds to the time difference between UTC and the local time where the evidence was acquired. This time zone information is obtained out-of-band and all timestamps are adjusted with a uniform translation. The forensic tools process an entire forensic image at a time and hence do not maintain separate time zone information for each artifact within the image. That is to say, when a file system is analyzed, the same time zone offset is applied to the files in the file system as is to the Internet logs discovered within it. However, often file systems and logs from different homogeneous sources do not maintain the same time reference. I illustrate a generic model (in XML) for representing timestamps in Figure 4.16.

```
<timestamp>
  <source-name> name </source-name>
  <source-type> type </source-type>
  <event> event-name </event>
  <modified> value </modified>
  <created> value </created>
  <last-accessed> value </last-accessed>
</timestamp>
```

Figure 4.16 Generic timestamp structure

The timestamp model shown in Figure 4.16 is an abstraction of the representation for a timestamps as observed in most forensic tools. The timestamp values that are not recorded are

represented by null. The source name uniquely identifies the evidence source and the type identifies the type of source, such as a hard disk image or log or network packet capture. The event identifies the specific event that is represented and created, modified and last-accessed refer to the timestamps with usual meaning.

4.10.4 The Timestamp Interpretation Problem

The time-lining tools, currently in existence, do not carry forward the time reference information for analysis. While forensic toolkits such as Encase, FTK or Sleuthkit can take a time-reference as input, it is usually a fixed offset value common to all the contents on a forensic acquired medium. This is illustrated in Figure 4.17.

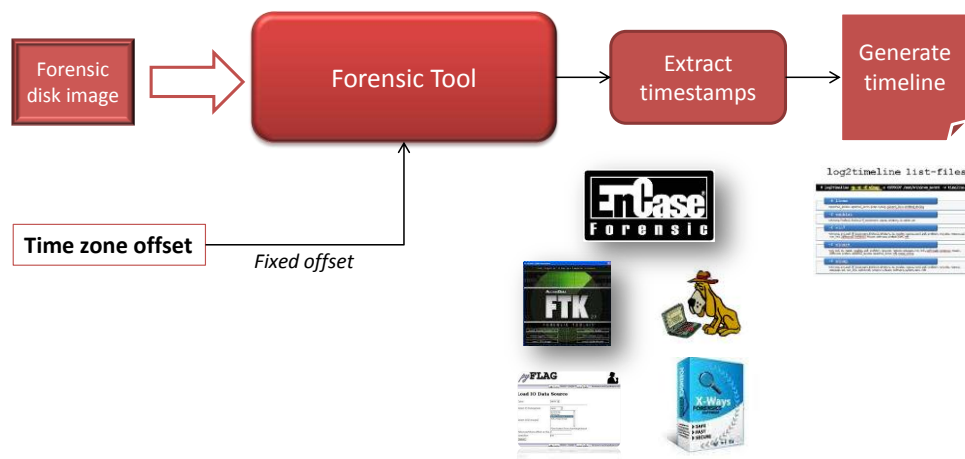


Figure 4.17 Timeline generation using traditional forensic tools

Consider how the conventional method works in the generation of a timeline. A forensic image is loaded into some forensic tool and the timestamps corresponding to the artifacts are extracted (using a fixed time zone offset) using the tools such as Encase, FTK, *Sleuthkit*. The timestamps are then sequenced using an analysis tool like *log2timeline*. Usually if the artifacts were obtained from the same source, all timestamps are treated as obtained in the same time zone. However, a hard disk as we all know is a mixture of all types of events, each having its own reference clock. Therefore applying a fixed time zone reference cannot always provide a homogeneous and consistent timeline. The time reference and the timestamp representation of these timestamps can significantly impact the timeline generated. For instance, if we consider a FAT file system with a Windows operating system, the files store timestamps as local system time while the Internet

Explorer application stores the browser log event timestamps in UTC, rendered in local time zone. I illustrate this challenge in Figure 4.18.

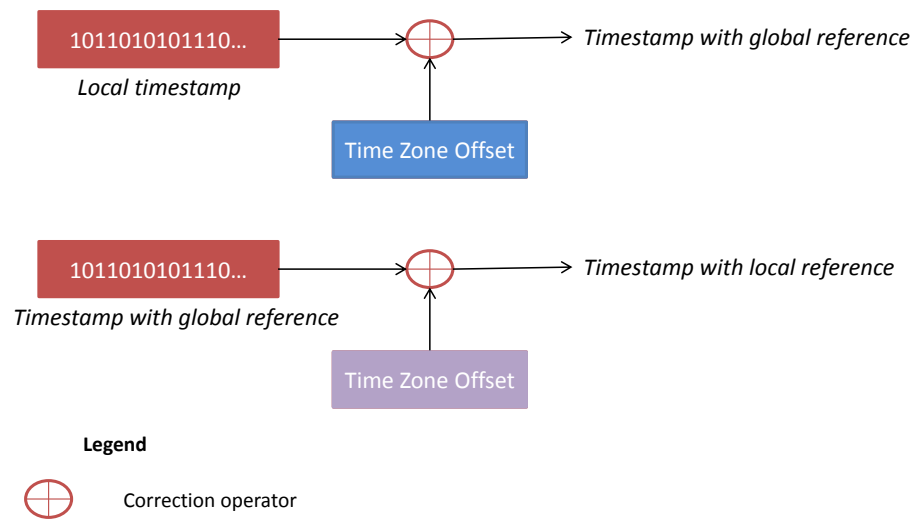


Figure 4.18 Differences in timestamp interpretation across timestamps with different timezone references

A timestamp can either have a local reference or some known global reference. For a local timestamp, one requires a time zone offset to determine its corresponding global value and vice versa for timestamp with global reference to determine its local time zone value. The two offsets are never the same. This results in two distinct problems with regard to timestamp interpretation. They are:

1. A timestamp in local time without zone information (in FAT file systems and ZIP file formats)
2. A timestamp in UTC time without zone information (in NTFS/EXTx file systems)

One important drawback with regard to present-day digital time-lining tools is that they do not interpret the value of timestamps obtained from the source during digital time-lining. The values are used as they are found on the source (in the appropriate representation format), except perhaps, when a fixed time zone shift is applied. I illustrate the ambiguities that result from challenges (1) and (2) in Figure 4.19 and Figure 4.20 respectively.

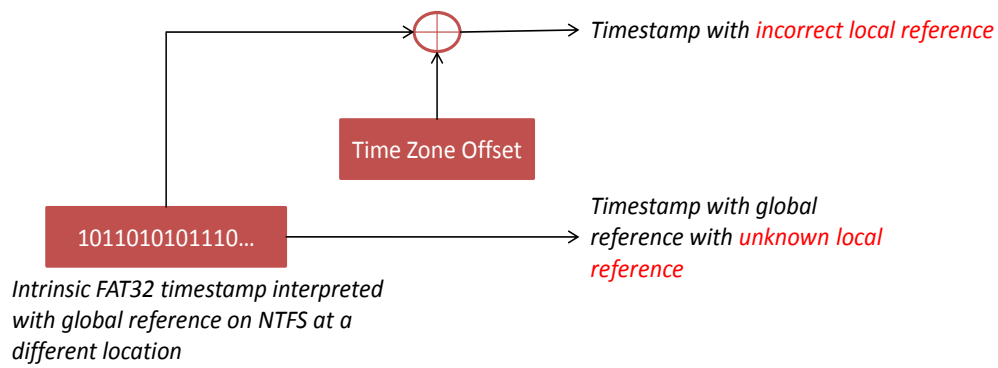


Figure 4.19 Intrinsic FAT32 timestamp interpreted with global timezone reference at a different location

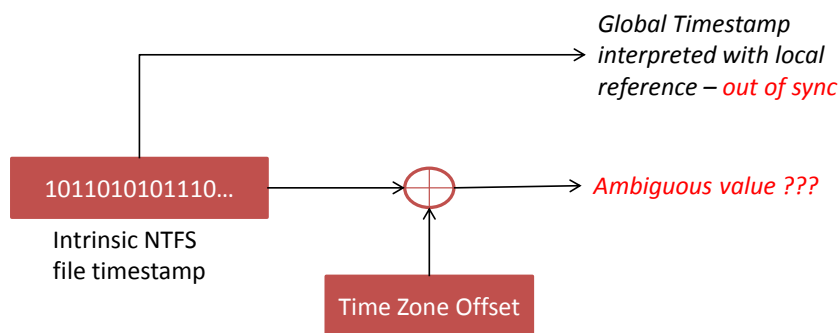


Figure 4.20 Intrinsic NTFS timestamp interpreted on a FAT32 file system

To address these challenges, it is necessary to distinguish the various homogeneous sources present even within a single forensic medium and develop a provenance model that is capable of recording sufficient provenance to facilitate accurate timestamp interpretation. The model should be capable of recording information relating to when and how a particular homogeneous source was acquired, its time zone shift with respect to UTC and ideally clock skew information. Such a provenance model will allow computing accurate timestamps to generate the unified timeline, is described in the following section.

4.11 Provenance Information Model to Normalize Timestamp Interpretation

The *Provenance Information Model* (or PIM) parallels the concept of Turner’s digital evidence bags [190], albeit with a practical outlook. While DEB records acquisition metadata such as date and time of acquisition, the size and contents of the source and so on, it fails to record time zone

information especially when dealing with FAT file systems on hard disks or ZIP archive files and so on.

4.11.1 Structure for Provenance Information Model

The PIM defines a structure for recording the time information associated with a homogeneous source for analysis that incorporates time zone shifts on individual timestamps to obtain values in reference to a single time zone. Each homogeneous source is associated with its own PIM to uniquely carry forward its time reference. The provenance information model for each homogeneous source records four important components that is carried forward along with the homogeneous source during analysis, viz.,

1. time zone information from where the homogeneous source was obtained,
2. any known clock skew for the homogeneous source when acquired,
3. summary of the acquisition process, and
4. assertions about events recorded in the homogeneous source.

The time zone information records the time shift of the event timestamps on a particular homogeneous source from UTC. Day light savings, if applicable, are also recorded alongside the time zone information. The PIM corresponding to the source provided in the DFRWS 2008 forensic challenge [51] contains UTC -0500 to denote the time zone of the location in the eastern coast of United States where the events were recorded. This information is recorded as a part of each homogeneous source identified in the evidence source. It is applied to the timestamps on files within ZIP archives, FAT user folder and the contents of browser cache to obtain global reference values (e.g. UTC) to generate a unified timeline. Known clock skew is also recorded and separately represented as a shift denoting number of seconds each timestamp is skewed off the reference clock. Unlike clock skew, clock drift presents a greater challenge as it is necessary to determine the exact rate at which the timestamps started to drift and the accumulated drift at the time of the acquisition (w.r.t. a reference clock).

4.11.2 Resilient Timestamps

Reference clock information for evidence is typically obtained out-of-band from the evidence location and transferred through manual documentation. This information, applied through forensic tools, incorporates a fixed offset to the evidence contents, without discrimination. It is

however, necessary to acknowledge that there can be multiple homogeneous sources within a single forensic medium and each source requires a separate storage mechanism to record the respective time zone shifts. *The PIM forms that medium*; the PIM is essential for FAT file systems where time zone information is not recorded. ZIP archives do not carry MAC information of their own, and only store the last modified timestamp of the files archived in them, that too in local time with reference to where the archive was created. Therefore, while examining ZIP archives, the PIM can be important to trace the provenance of the archived files. In essence, the PIM recorded for a particular homogeneous source is applied to each timestamp to derive a referenced local timestamp and corresponding global (UTC) timestamp for:

- i. a local timestamp with no time zone information; and
- ii. a global timestamp with no local time zone information.

Besides this, the reference clock information can also be used to reverse inadvertent time zone shifts caused by analysis tools while processing the homogeneous sources, rendering the timestamps resilient to time zone shifts which can produce a robust timeline.

By virtue of the resilience imparted to the timestamps, the PIM is not merely a place-holder for reference clock information; the PIM can also be used to validate and identify, if not correct, ambiguous or uncertain timestamps. When assertions are recorded in the PIM, those assertions can be validated during digital time-lining. A variety of assertions can be recorded in the PIM; for example, one may assert that all documents in a user folder have the same value for the metadata ‘Author’.

4.11.3 Identifying and Validating Inconsistent Timestamps

Maintaining the UTC and a local timestamp value for each timestamp serves two purposes; firstly, to digitally timeline the events with respect to global reference, the UTC is used to which all event timestamps, irrespective of the homogeneous source type are converted, and secondly, the local time zone can be used to allow for assertions and hypothesis within the PI of each homogeneous source that can be tested and reported back to the examiner on the outcome. For example, the examiner may posit that documents should have been used between working hours, i.e., the timestamps should have been recorded after 9 AM and before 5 PM on a weekday. Note that the examiner need not be certain that these values are necessarily correct. If this hypothesis was indeed true, it can allow one to omit files considered irrelevant and focus on a narrower group.

An examiner can make assertions such as, “*All timestamps found on a particular homogeneous source should have timestamps less than the date and time when that homogeneous source was acquired*”. When this assertion is satisfied, it guarantees that the homogeneous source has been processed according to proper procedures as a sanity check mechanism. On the other hand, if this assertion is not satisfied, one of two possibilities is likely, either the chain of custody is faulty, or the timestamps were intentionally tampered. While it is still possible for such timestamps to be found with no malicious intent, such decisions are left to the examiner.

The design and implementation of my provenance information model is discussed in Chapter 5 of this thesis. I conclude this chapter with a brief summary of the work reported.

4.12 Chapter Summary

In this chapter I conducted a review of contemporary forensic and analysis tools to abstract the different functionalities supported to analyze different sources of digital evidence. This review culminated in the design of the *functional Forensic Integration Architecture* which consolidated these functionalities and defined a new layer to group artifacts based on metadata associations. I conducted experiments to elicit the syntax and semantics associated with metadata associations which were determined through the identification of metadata matches. I generalized my findings which resulted in the *metadata association model*. These contributions directly address my research objectives stated in Chapter 1 towards developing a framework for identifying associations in digital evidence using metadata. The architecture has resulted in a design for an analysis engine to analyze metadata associations among sources of digital evidence. The design of this tool is discussed in the next chapter.

We identified and highlighted the interpretation challenges in processing timestamp-based associations across heterogeneous sources during analysis. To address this challenge, I developed the *provenance information model* which develops resilient timestamps for digital time-lining across multiple sources. I developed a prototype toolkit implementing this model and generate a unified timeline. The design of this tool is discussed in the next chapter.

“No amount of experimentation can ever prove me right; a single experiment can prove me wrong.”

- Albert Einstein

5. Prototype Implementation

In this chapter, I present a practical framework to validate the metadata associations model (MAM) designed in Chapter 4 to identify metadata associations across multiple sources and group them. This chapter describes my demonstrable implementations of the metadata association and provenance information models which were used to translate forensic questions into MAM based experiments. I have developed two prototype toolkits which I describe in this chapter. The first prototype, called the AssocGEN analysis engine, is used to identify generic metadata matches across digital artifacts and group the associated artifacts, and the second, called UniTIME, is used to incorporate the provenance information model for timestamp interpretation and generate a unified timeline across multiple sources.

5.1 Prototype Development One: The AssocGEN Analysis Engine

The AssocGEN Analysis Engine was my research prototype implementation of the MAM used to access heterogeneous sources of digital evidence and unify the analysis by identifying metadata matches between them. Its design adopts many of the principles proposed in *f*-FIA (Section 4.2) but exclusively focusses on the development of the Evidence Composition layer to combine digital artifacts using metadata associations. The AssocGEN architecture is shown in Figure 5.1. AssocGEN can extract metadata from digital artifacts belonging to forensic hard disk images, Internet browser logs (both history and cache logs) and network packet captures. AssocGEN was developed in Java and is cross-platform compliant.

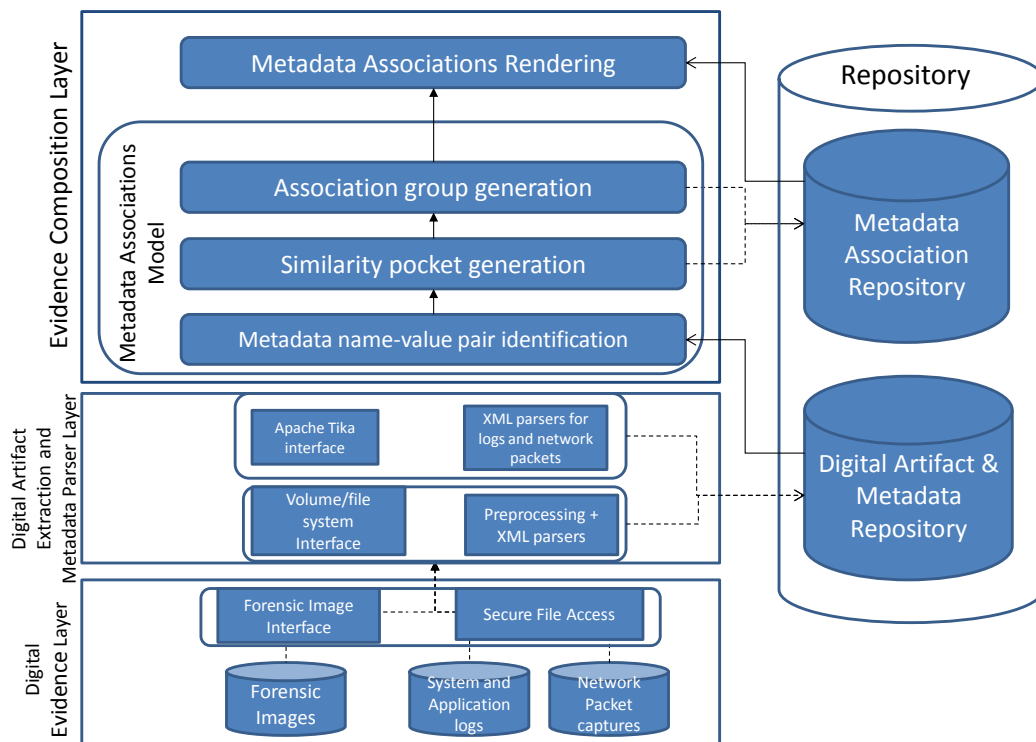


Figure 5.1 The AssocGEN architecture

5.1.1 Rationale for the Design

AssocGEN was primarily designed with the view of abstracting current technological support extended to heterogeneous sources of digital evidence. My review of contemporary forensic and analysis tools (refer to Chapter 4) informs this design. Since my primary focus in this analysis engine was to extract metadata associations across heterogeneous digital artifact, I adopted suitable software support to implement the lower layers that conform to forensic requirements [130]. Besides this, as I focused on using the metadata extracted/parsed from digital artifacts and not the entire evidence source, it was sufficient to implement these functionalities using existing software libraries. This rationale applies to the lower layer implementation of both prototypes, AssocGEN, in this section and UniTIME, described in the following section. The Digital Evidence layer provides binary-stream access to digital evidence. In current technology, the file system support provided in Sleuthkit, the evidence image libraries *ewflib* and *afflib* and the *Snorkel file system library* are potential candidates to provide this support.

Among these, *Sleuthkit*³¹ accesses a source of digital evidence as a monolithic bit stream and handling discrete objects such as digital artifacts and metadata can be an implementation

³¹ <http://www.sleuthkit.org/sleuthkit/>

challenge. The *ewflib*³² has similar concerns, best taken advantage of using commercial forensic toolkits like Encase which encapsulate using proprietary binary stream interfaces. The *afflib* accesses a source of digital evidence as inodes which store the attribute related to the contents. The abstractions modeled in the *afflib*³³ library are more conducive to raw binary data and stream processing rather than the discrete digital artifact abstraction which is the focus of my work. *Afflib* was therefore, restrictive in terms of being able to define a generic metadata structure to determining metadata matches. The *Snorkel*³⁴ library, on the other hand, provided the necessary abstractions to handle digital artifacts as read-only nodes with metadata. Besides, being developed in Java, it can readily integrate with other Java libraries for log parsing and network packet analysis which can be automated. Hence, I chose *Snorkel* to implement file system support and digital artifact traversal in forensic images.

With regard to the metadata parsers from file systems, there were three contenders, viz., the *libextractor*³⁵, *fiwalk*³⁶ and the *apache tika*³⁷ libraries. Of these, the *libextractor* was completely built in C and had (at that time) limited file metadata support to word processing documents. Besides this, the memory requirements to handle the metadata structures and determine associations during runtime were very demanding. *fiwalk* was also developed in C and more conducive to Linux environments where an 'inode' implementation was handy, but its metadata extractor was in its early stages of development. In comparison, the *apache tika* library was developed in Java which could be readily integrated with my digital evidence access layer implementation and provided the necessary abstractions to deal with metadata matches at the digital artifact level. The abstractions supported by *apache tika* were readily mapped to event semantics that allowed effective grouping of digital artifacts that were deemed related through metadata associations. Therefore, I chose *apache tika* to implement the metadata parsers in AssocGEN.

To process the log records and network packets individually, I processed logs and network traces and translated them into XML where each tag represented an attribute. The Internet browser logs and network packet captures, which were initially extracted as files from a file system, were converted into XML and then parsed into individual log records and network packets from their

³² <http://code.google.com/p/libewf/>

³³ <http://afflib.sourceforge.net/>

³⁴ <http://www.holmes.nl/NFIILabs/snorkel.html>

³⁵ <http://www.gnu.org/software/libextractor/>

³⁶ <https://github.com/yalemssa/fiwalk>

³⁷ <http://tika.apache.org/>

respective schema. The metadata obtained from each evidence source, viz., file system, or log or network packet capture, was represented as a list of hash tables indexed by the file path in the case of files and a numerical event ID in the case of Internet browser logs or network packet captures. The XML representation for the logs and the network packets contained tags which were extracted as metadata. I developed XML parsers to process Internet browser logs and network packet captures and extract the attributes of the records from the logs and the packets in the network packet captures in AssocGEN. The completed prototype toolkit spans over 20000 lines of Java code and consists of multiple modules. The modules are pluggable at runtime and can access and parse files and folders from most common file systems such as FAT32, NTFS, EXT2, EXT3 and HFS+, web page visitation and cache logs on browser applications and network packets contained within packet captures.

5.1.2 Digital Evidence Layer

The Digital Evidence layer was built using the Snorkel forensic library which is responsible for providing raw binary access through a forensic file system interface. The snorkel library mirrors the functionality of the *fiwalk* tool [69]. Internet browser logs and network packet captures were treated as record-based files and this layer provides preliminary secure access to such files. The digital evidence layer provides regulated bit-stream access to the various different digital evidence sources from the upper layers. The layer allows unidirectional data flow ensuring read-only access to forensic images, file systems, Internet browser logs and network packet captures implemented by the snorkel forensic image interface. The snorkel interface allows traversing multiple forensic images without compromising data integrity.

5.1.3 Digital Artifact Traversal & Metadata Parsing Layer

The digital artifact extraction and metadata parser layer is composed of third party applications that I designed to traverse the digital artifacts and parse the metadata. This layer is implemented using the Apache tika metadata extractor library to parse metadata from files and log analyzers to traverse log records and network packets and parse their attributes. I extract the metadata from files based on the file MIME type. The MIME type for a file is identified by determining its encoding type in conjunction with its magic numbers identifying the file beginnings and endings.

The browser logs are initially processed by a third party application (Nirsoft browser analyzer³⁸) into XML which is then read by my parsers to extract the attributes for individual browser events.

³⁸ http://www.nirsoft.net/web_browser_tools.html

The browser history is equivalent to a log that contains URI records; the specific web pages visited, its domain name, and the last visit timestamp were regarded as its metadata. Similarly, on a network packet, the packet timestamp, source and destination IP addresses, and protocol are regarded as associated metadata. The current implementation for parsing logs supports Internet browser (history and cache) logs for the Internet Explorer, the Mozilla Firefox and the Apple Safari, accessing all related log event attributes as the metadata corresponding to each browser log event. Besides this, the network module can scan captured network packets accessing all packet related attributes as the metadata corresponding to each packet.

5.1.4 Evidence Composition Layer

The Evidence Composition layer comprises of algorithms that seek metadata matches between the various digital artifacts and group them. These groupings are merged and can be presented to a forensics examiner for analysis. Although the MAM groups all digital artifacts which have associated metadata into an association group, AssocGEN is configured to prioritize based on metadata matches by determining the *source*, *ownership* and *timestamps* of digital artifacts, for instance, all digital images captured using a Canon Powershot A70 camera on September 11th, 2011.

Between two or more digital artifacts, a single metadata match can lead to a set of digital artifacts that have an identical value for that metadata tag name. Such a set was termed a *similarity pocket* as per Section 4.5. Each pocket is identified by the metadata tag name. A set of artifacts may have multiple metadata tag matches giving rise to multiple similarity pockets each including an identical subset of artifacts—multiply matched subset artifacts are a special case of a similarity pocket that I term a *multi pocket*. Similarity pockets may also overlap partially in regard to their elements, i.e., digital artifacts. If there are two overlapping similarity pockets within a single source of digital evidence, these are merged into a *similarity group* as per Section 4.5. When such similarity groups match across multiple sources, these are merged into an *association group* as per Section 4.5. Merging the overlapping similarity pockets is continued until all transitive overlaps are accounted for. When multiple similarity pockets are merged into a similarity group and multiple similarity groups into an association group, the individual similarity pockets and similarity groups in the repository are replaced with the resultant association group incorporating all the metadata matches.

Digital artifacts may belong to different types but have metadata tags with identical or similar semantics. Therefore, *metadata tag equivalence* was established for those metadata tags whose values tend to be of the same type, i.e., metadata tags that take the names of individuals, metadata tags that take the values of applications, metadata tags that take timestamps, and so on. Such equivalence relations are configured into AssocGEN ahead of execution depending on the diversity that the sources of evidence present. For instance, the author name on a document could match the username in a record from Internet browser logs or the attribute timestamp in browser history logs can match with the corresponding timestamp in network packet captures and so on. The algorithms terminate when all the digital artifacts in the digital artifact and metadata repository have been grouped or classified.

In each case, the set of N distinct homogeneous sources of digital evidence is provided as input. A source s_i contains N_i digital artifacts and the j^{th} digital artifact a_j^i is associated with a metadata vector $m_j^i = (m_{1,1}^{ij}, m_{2,1}^{ij}, \dots, m_{k,1}^{ij}, \dots, m_{M,1}^{ij})$. The naïve algorithm to determine metadata matches across the sources of digital evidence is described in Algorithm 5.1. The free-running variable t accounts for the different disjoint similarity pockets or groups generated within a source and association groups generated across all sources and t is a member of the set of natural numbers \mathbf{N} . The desired output is the set of all association groups represented by the set AG in the algorithm.

Association Grouping Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence

$A_i = \{a_1^i, a_2^i, a_3^i, \dots, a_{N_i}^i\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m_{1,1}^i, m_{2,1}^i, m_{3,1}^i, \dots, m_{N_i,1}^i\}$, the set of all metadata vectors corresponding to each $a_j^i \in A_i$,
 $j \in [1, N_i]$

Output: AG , the set of all association groups generated on S

begin algorithm

for each $s_i \in S$ **do**

for each $a_j^i \in A_i$ **do**

for each $m_{k,1}^{ij} \in m_j^i$ corresponding to the j^{th} artifact $a_j^i \in A_i$ **do**

$sp_{t,1}^{ik} \leftarrow \{a_j^i \mid j \in [1, N_i], (\exists v, m_{k,1}^{ij} = v)\}$

end for

end for

$SP^i \leftarrow \{sp_{t,1}^{ik} \mid k \in [1, M], t \in \mathbf{N}\}$

end for

for each $sp_{t,1}^{ik} \in SP^i$ **do**

$sg_t^i \leftarrow$ largest union (until transitive closure) over those similarity pockets that overlap on artifacts across all metadata $m_{k,1}^{ij} \in m_j^i$, for all $k \in [1, M]$

end for

```

SG ← {sgit | i ∈ [1, N], t ∈ N}
for each sgit ∈ SG do
    agt ← largest union (until transitive closure) over those similarity groups that
        contain at least one metadata match based on a metadata equivalence
        relationship between the sources
end for
AG ← {agt | t ∈ N}
Display AG as output
end algorithm

```

Algorithm 5.1 Association grouping algorithm

No processing occurs if all the similarity pockets are disjoint, viz., they contain no common artifacts. If metadata were unavailable in digital artifacts for any reason, those artifacts are removed to an *unclassified* list. This list can be separately presented to a forensics examiner who may manually examine the files for content using a different tool like Sleuthkit or FTK.

5.1.4.1 Metadata Equivalence in AssocGEN

AssocGEN allows the establishment of equivalence relationships between metadata tag names to support the identification of metadata matches across heterogeneous digital artifacts. In terms of the model, it allows the expansion of the similarity groups in each of the sources of digital evidence into association groups. I establish equivalence between the following sets of metadata:

1. Ownership, author(s) in files and usernames in system and application logs;
2. MAC timestamps, document metadata timestamps in files and log event timestamps in system and application logs and network packet timestamps in network packet captures;
3. IP addresses and domain names from DNS lookups in browser logs and network packet captures;
4. Filesizes from file system metadata with ‘Filesize’ and ‘Content size’ in document metadata;
5. ‘Subject’ and ‘Title’ metadata in Microsoft Office documents;
6. ‘Creator’ and ‘Publisher’ metadata in Microsoft Office documents; and
7. ‘Source’ in packet captures with ‘Domain’ in browser logs.

In each case where metadata equivalence is established, the metadata tag names are treated as identical and value matches are determined. Each value match gives rise to an association group if the digital artifacts corresponding to that association were not already a part of any other association group.

5.1.4.2 Configuring and Controlling Metadata Associations in AssocGEN

Typically, AssocGEN extracts all metadata from each digital artifact and groups artifacts according to the inherent metadata matches. As this can be an exhaustive approach with significant computational complexity, I have also developed an alternate implementation for AssocGEN that allows a user to specify a subset of metadata from the digital artifacts (either based on apriori knowledge or based on a cursory manual examination of the file metadata), often based on their application type, in order to contain the number of metadata matches found and hence constrain the size of the groups formed. This approach enables a user to focus on the relevant sets of associations and quickly identify the relevant artifacts for further analysis.

The AssocGEN user interface customized to analyze files from file systems is illustrated in Figure 5.2 and Figure 5.3. The user interface is customized to determine patterns that are specific to the type of files being analyzed and relevant during an investigation. Each association class indicated in the snapshot results in a classification that is used to prime the process of identifying associations between the files across these classes. For example, when a camera based classification is chosen, the digital image files are organized according to their EXIF metadata and the digital images that are associated across different cameras are identified using metadata associations. An instance of this can be digital images taken with different cameras but edited with the same photo-editing software. The resultant groupings contain those digital images that are associated by their subsequent manipulation rather than containing images captured with the same digital camera.

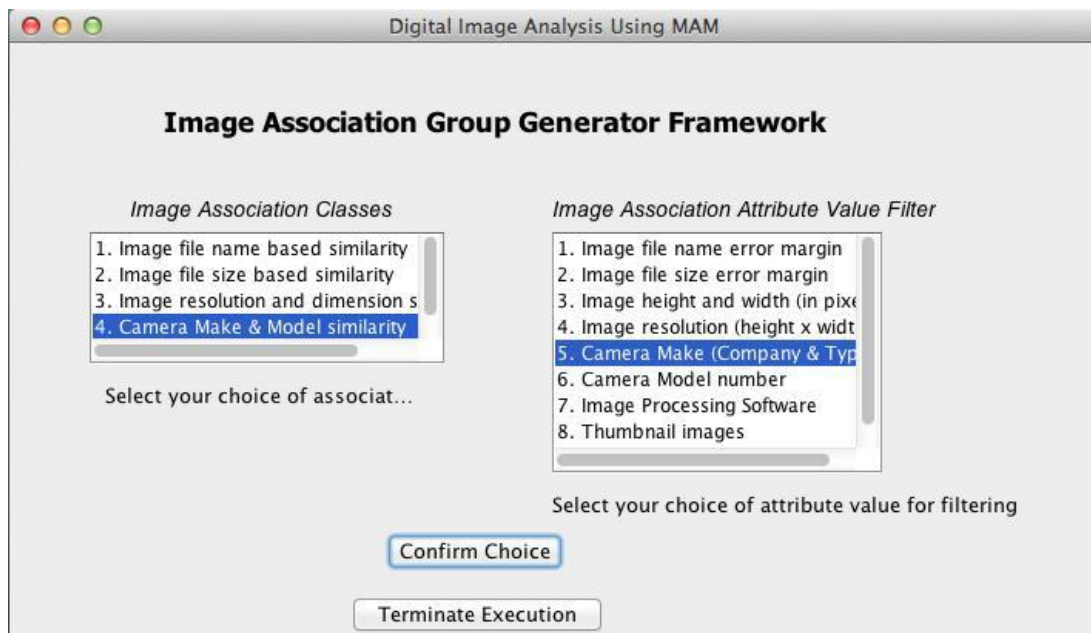


Figure 5.2 AssocGEN User Interface to analyze collections of digital image files from Digital Evidence

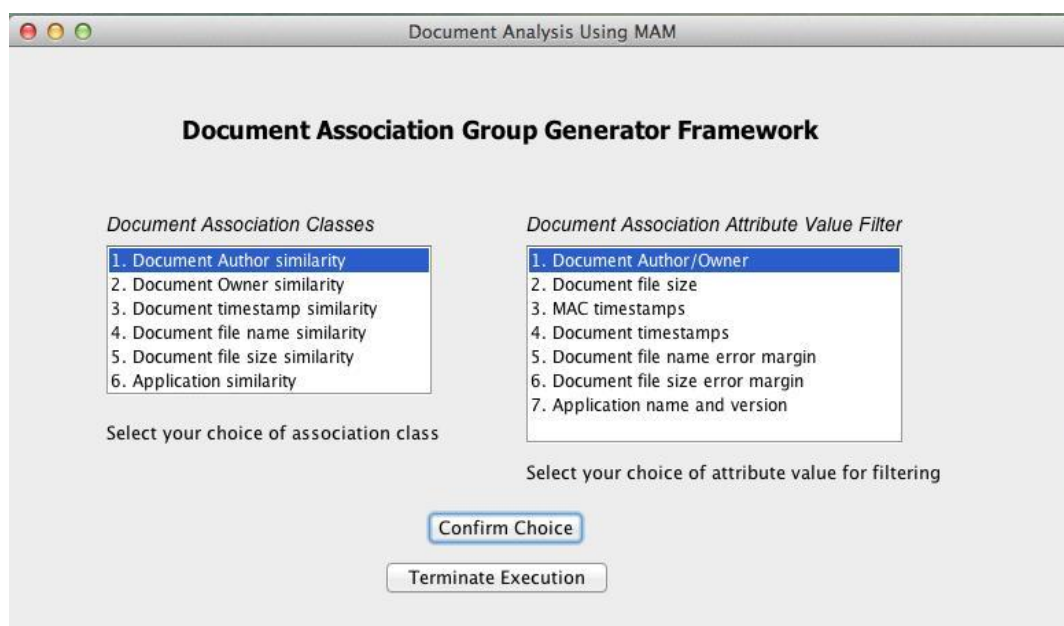


Figure 5.3 AssocGEN User Interface to analyze word processing documents from Digital Evidence

During analysis a forensics examiner may need to analyze all related artifacts on a source together and determining such related items can be a strenuous task. As observed earlier, this can involve multiple keyword searches and analysis to determine those sets of artifacts that are considered “relevant”. The AssocGEN engine allows a user to simply select an artifact at random and generates a list of all artifacts on the same source or across all sources (as configured) by determining all associated artifacts. It is also likely that an artifact that is found during a search is

related to other artifacts on a different metadata name. However, for the purpose of analysis, it is necessary to take such associations into account. Therefore, AssocGEN also determines all second level, third level and further levels of searches until all artifacts that can be potentially associated across all possible metadata combinations are determined. Such an incremental approach adopts an incremental search method as given in Algorithm 5.2. The desired output is the association group corresponding to the seed artifact a_j^i from the source s_i . This is represented by the set AG in the algorithm.

Incremental Association Builder Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence

$A_i = \{a_1^i, a_2^i, a_3^i, \dots, a_{N_i}^i\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m_1^i, m_2^i, m_3^i, \dots, m_{N_i}^i\}$, the set of all metadata vectors corresponding to each $a_j^i \in A_i$, $j \in [1, N_i]$

Seed: Vector m_j^i corresponding to the j^{th} artifact $a_j^i \in A_i$ for some source s_i , $i \in [1, N]$

Output: AG , the association group corresponding to seed artifact a_j^i for some $j \in [1, N_i]$ on source s_i

begin algorithm

$SG_i \leftarrow \emptyset$

procedure *similarity group* (**input:** *seed artifact* a_j^i):

for each $m_k^j \in M_j^i$ corresponding to the j^{th} artifact $a_j^i \in A_i$ **do**

$sp_k^i \leftarrow \{a_j^i | j \in [1, N_i], (\exists v, m_k^j = v)\}$

end for

$sg_i \leftarrow \bigcup_{k=1}^M sp_k^i$

$SG_i \leftarrow SG_i \cup sg_i$

end *similarity group*

Move all artifacts $a_j^i \in sg_i$ to list L ; remove seed artifact from L

while L not empty **do**

$temp \leftarrow$ first artifact in L

Perform *similarity group* (**input:** $temp$)

Remove $temp$ from L

end while

for each $s_i \in S$ **do**

for each $a_j^i \in SG_i$ **do**

Determine artifacts on all other sources where metadata equivalence relationship exists

Append artifacts to list L indexed as (artifact a_j^i , source s_i); omit repetitions

end for

end for

while L not empty **do**

$temp \leftarrow$ first artifact in L

Perform *similarity group* (**input:** $temp$)

Remove *temp* from *L*

end while

$$AG \leftarrow \bigcup_{i=1}^N SG_i$$

Display *AG* as output

end algorithm

Algorithm 5.2 Incremental association builder algorithm

An interesting feature to note about the incremental algorithm is that since the associations are symmetric by definition, any artifact from a particular association group provided as seed input will result in the same association group. This provides a method to verify the correctness of the generated group by randomly testing the algorithms for any two artifacts belonging to the group. If the results do not coincide, it indicates that the transitive closure is not yet met and the algorithm must proceed to completion.

In the sequel, I describe different metadata association algorithms that we've developed to answer the questions posed in Section 4.7 in the previous chapter.

5.1.4.3 Metadata Association Algorithms to Determine Artifact Relationships

The algorithms described in this section identify the relationships between artifacts as defined in Section 4.9. The relationships can exist based on an exact value match between two or more artifacts of the same type on the same homogeneous source or across heterogeneous artifacts based on a value match established through a metadata equivalence relationship on the corresponding metadata names across sources. In all the algorithms described below, the free-running variable *t* accounts for the different disjoint similarity pockets or groups generated within a source and *t* is a member of the set of natural numbers **N**.

In order to identify the digital artifacts belonging to the same source, I apply the *source relationship* (refer to Section 4.9.2) and identify all metadata associations that produce similarity pockets for each metadata in the source metadata family. Digital artifacts that produce multiple metadata matches are extracted from either similarity groups on a single source or from association group across multiple sources as per Algorithm 5.3.

Source Identification Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence
 $A_i = \{a^i_1, a^i_2, a^i_3, \dots, a^i_{N_i}\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m_{1,i}^i, m_{2,i}^i, m_{3,i}^i, \dots, m_{N_i,i}^i\}$, the set of all metadata vectors corresponding to each $a_j^i \in A_i, j \in [1, N_i]$

Set L of metadata corresponding to the *source metadata family* on each source s_i

Output: A list of sources stored in *orgn* and the corresponding sets of digital artifacts SP^i

begin algorithm

$orgn \leftarrow 0; SP^i \leftarrow \emptyset$

for each $s_i \in S$ **do**

repeat

for each $a_j^i \in A_i$ **do**

for each $m_{k,i}^i \in m_{j,i}^i$ and $m_{k,i}^i \in L$ **do**

$sp_{t,i}^{ik} \leftarrow \{a_j^i | j \in [1, N_i], (\exists v, m_{k,i}^i = v)\}$

end for

end for

$SP^i \leftarrow \{sp_{t,i}^{ik} | k \in [1, M], i \in [1, N], t \in \mathbf{N}\}$

$orgn \leftarrow$ list v of values corresponding to each similarity pocket in SP^i

until $|orgn| = |SP^i|$

end for

Generate a list *orgn* of individuals or devices from all $sp_{t,i}^{ik} \in SP^i$ where $j \in [1, N_i], k \in [1, M]$

Display *orgn*, SP^i as outputs

end algorithm

Algorithm 5.3 MAM based Source identification algorithm

For this algorithm, the list L maintains a list of those metadata that record values corresponding to source devices or software that were used to generate the artifact it was attributed to. The source device or software can be different from the source of digital evidence that contains the artifact. For instance, on a file, the list for possible sources can include the metadata ‘Author’, ‘Owner’, and ‘Computer-Name’. Where necessary, the metadata equivalence relationships are established across artifacts belonging to heterogeneous artifacts before executing the algorithm. The sets of pockets generated are arranged according to the source name which then characterizes the associations among the artifacts contained.

The verification condition ‘ $|orgn| = |SP^i|$ ’ in the algorithm is used as a measure to test the completeness of the set of similarity pockets generated. When the verification condition is met, it indicates that transitive closure is achieved and that the algorithm can successfully terminate. When this condition is not met, the transitive closure is not yet achieved and the algorithm must iterate until the condition is satisfied.

Having grouped artifacts that demonstrate the same source associations, it may be necessary to also determine some artifacts from that set which are modified. Typically, this can imply that artifacts belonging to some source were doctored using same software. However when two artifacts demonstrate the software edited relationship, it may need to be established, with the presence of a third artifact, that in conjunction with the first artifact exerts a majority relationship. This is because, with regard to digital image files where this relationship holds forensic value, sometimes the absence of metadata can imply software activity, as in the case of digitally generated image files and image files downloaded from the Internet (for a detailed discussion, refer to Chapter 6). In order to identify all digital artifacts that were edited with a particular piece of software, I apply the *unauthenticated modified relationship* (refer to Section 4.9.7) and for each pair, identify a third digital artifact, two of which can exert a *majority relationship* (refer to Section 4.9.8) over the third for the ‘Software’ in the source metadata family as per Algorithm 5.4. For this task, the digital artifacts are homogeneous in nature and naturally will be contained within similarity groups.

Edits Identification Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence

$A_i = \{a^i_1, a^i_2, a^i_3, \dots, a^i_{N_i}\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m^i_1, m^i_2, m^i_3, \dots, m^i_{N_i}\}$, the set of all metadata vectors corresponding to each $a^i_j \in A_i$, $j \in [1, N_i]$

Output: A list *sftw* of software and corresponding sets of digital artifacts SP^i

begin algorithm

for each $s_i \in S$ **do**

repeat

for each $a^i_j \in A_i$ **do**

for each $m^i_k \in M_i$ corresponding to the j^{th} artifact $a^i_j \in A_i$ **do**

$sp^{ik}_t \leftarrow \{a^i_j \mid j \in [1, N_i], (\exists v, m^i_k = v)\}$

end for

end for

$SP^i \leftarrow \{sp^{ik}_t \mid k \in [1, M], t \in \mathbf{N}\}$

Extract unauthenticated modification relationship $\{(a^i_j, a^i_k) \mid a^i_j R_{ua} a^i_k, j \in SP^i, k \in SP^i\}$ for each artifact a^i_j from similarity pockets in SP^i

for each (a^i_j, a^i_k) pair identified **do**

Identify a third artifact a^i_n such that $a^i_n R_m a^i_j$ for similarity pockets in SP^i

end for

```

         $stfw \leftarrow$  source metadata name corresponding to the software that established the
        modified relationship  $R_m$  on triad  $a_j^i, a_k^i, a_n^i$  from similarity pockets in  $SP^i$ 
    until  $|stfw| = |SP^i|$ 
    Display  $stfw, SP^i$  for  $s_i$  as outputs
end for
end algorithm

```

Algorithm 5.4 MAM based algorithm to identify software edits in digital artifacts

As in the case of Algorithm 5.3, the verification condition ' $|stfw| = |SP^i|$ ' in the algorithm is used as measure to test the completeness of the set of similarity pockets generated. When the verification condition is met, it indicates that transitive closure is achieved and that the algorithm can successfully terminate.

In order to determine the origin of downloaded files, I apply the *download relationship* (refer to Section 4.9.4) and determine log record y to establish that a file is indeed downloaded. For each log record y , I find a *happens before relationship* (refer to Section 4.9.3) with log record x such that $y \Rightarrow x$. The source metadata for log record x contains the origin of the download.

Resource Download Identification Algorithm

Given: S = the set of sources consisting of a user file system, temp files and browser logs $\{s_1, s_2, s_3\}$ respectively

$A_i = \{a_1^i, a_2^i, a_3^i, \dots, a_{N_i}^i\}$, the set of all digital artifacts belonging to source $s_i, i \in \{1, 2, 3\}$

$M_i = \{m_1^i, m_2^i, m_3^i, \dots, m_{N_i}^i\}$, the set of all metadata vectors corresponding to $a_j^i \in A_i$ for each A_i in source $s_i, i \in \{1, 2, 3\}$

Output: List *URL* of resource download sources corresponding to files in source s_1

begin algorithm

for each $s_i \in S$ **do**

for each $a_j^i \in A_i$ **do**

for each $m_k^i \in M_i$ corresponding to the j^{th} artifact $a_j^i \in A_i$ **do**

$sp_t^{ik} \leftarrow \{a_j^i | j \in [1, N_i], (\exists v, m_k^i = v)\}$

end for

end for

$SP^i \leftarrow \{sp_t^{ik} | k \in [1, M], t \in \mathbf{N}\}$

Identify all pairs of artifacts such that $a_j^2 R_s a_k^1$ where $a_k^1 \in SP^1$ and $a_j^2 \in SP^2$

for each temp file a_j^2 ,

Find all log records x in s_3 such that $a_x^3 R_d a_j^2$

end for

$URL \leftarrow$ values for source metadata from each log record a_x^3 identified

Display *URL* as output

end algorithm

Algorithm 5.5 MAM based algorithm to determine source of downloaded resources in user file system

In order to determine all digital artifacts that occurred simultaneously, I apply the *parallel occurrence relationship* (refer to Section 4.9.5) to extract those digital artifacts that are grouped based on associations from the timestamp metadata family. When all digital artifacts belong to a single homogeneous source, these artifacts are grouped into similarity groups and the digital artifacts span multiple sources, and they will form association groups. A new group is formed for each timestamp value that generates associations.

Simultaneous Access Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence

$A_i = \{a^i_1, a^i_2, a^i_3, \dots, a^i_{N_i}\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m^i_1, m^i_2, m^i_3, \dots, m^i_{N_i}\}$, the set of all metadata vectors corresponding to each $a^i_j \in A_i$,
 $j \in [1, N_i]$

Set L of metadata corresponding to *timestamp metadata family* on each source s_i

Output: List *concurrent* of sources and corresponding sets of digital artifacts SP^i

begin algorithm

for each $s_i \in S$ **do**

repeat

for each $a^i_j \in A_i$ **do**

for each $m^{ij}_k \in m^i_j$ and $m^{ij}_k \in L$ **do**

$sp^{ik}_t \leftarrow \{a^i_j | j \in [1, N_i], (\exists v, m^{ij}_k = v)\}$

end for

end for

$SP^i \leftarrow \{sp^{ik}_t | k \in [1, M], t \in \mathbf{N}\}$

concurrent \leftarrow Number of timestamp values giving rise to similarity pockets in SP^i

until $|concurrent| = |SP^i|$

end for

Display *concurrent*, SP^i as outputs

end algorithm

Algorithm 5.6 MAM based algorithm to determine all artifacts affected by parallel events

As in the case of Algorithms 5.3 and 5.4, the verification condition ' $|concurrent| = |SP^i|$ ' in the algorithm is used as measure to test the completeness of the set of similarity pockets generated.

In order to determine all digital artifacts that are similar in structure, I apply the *structural similarity relationship* (refer to Section 4.9.6) using the application metadata family. Single metadata value matches will result in similarity pockets and multiple metadata matches are grouped into similarity groups. By virtue of structural similarity, this property applies to digital artifacts on a single type of homogeneous source, i.e., file systems, browser log files, network packets, etc.

Similar Structure Identification Algorithm

Given: $S = \{s_1, s_2, s_3, \dots, s_N\}$, the set of all discrete homogeneous sources of digital evidence

$A_i = \{a^i_1, a^i_2, a^i_3, \dots, a^i_{N_i}\}$, the set of all digital artifacts belonging to source s_i , $i \in [1, N]$

$M_i = \{m^i_1, m^i_2, m^i_3, \dots, m^i_{N_i}\}$, the set of all metadata vectors corresponding to each $a^i_j \in A_i$, $j \in [1, N_i]$

Output: List of lists L containing artifacts that are structure similar (in dimensions and formatting)

begin algorithm

$L \leftarrow \text{empty}$

for each $s_i \in S$ **do**

for each $a^i_j \in A_i$ **do**

for each $m^i_k \in m^i_j$ corresponding to the j^{th} artifact $a^i_j \in A_i$ **do**

$sp^{ik}_t \leftarrow \{a^i_j \mid (\exists v, m^i_k = v), j \in [1, N_i], t \in \mathbf{N}\}$

end for

end for

$SP^i \leftarrow \{sp^{ik}_t \mid k \in [1, M], t \in \mathbf{N}\}$

end for

for each $sp^{ik}_t \in SP^i$ **do**

$sg^i_t \leftarrow$ largest union (until transitive closure) over those similarity pockets that overlap on artifacts across all metadata $m^i_k \in m^i_j$, for all $k \in [1, M]$

end for

$SG \leftarrow \{sg^i_t \mid i \in [1, N], t \in \mathbf{N}\}$

for each $sg^i_t \in SG$ **do**

$ag_t \leftarrow$ largest union (until transitive closure) over those similarity groups that contain at least one metadata match based on a metadata equivalence relationship between the sources

end for

$AG \leftarrow \{ag_t \mid t \in \mathbf{N}\}$

for each $ag_t \in AG$ **do**

 Extract a similar structure relationship for each artifact $a^i_j \in ag_t$ such that for some artifact $a^n_k \in ag_t$ there exists a relation R_{ss} such that $a^i_j R_{ss} a^n_k$ where $i, n \in [1, N]$ and $j \neq k$

 For each R_{ss} append to L the set of all artifacts from ag_t that are linked by this relation

end for

Display L as output

end algorithm

Algorithm 5.7 MAM based algorithm to identify all similarly structured artifacts

5.1.4.4 Mapping Forensic Discoveries to Digital Artifact Relationships

Until now, I have described the algorithms that are implemented as part of the Evidence Composition layer in my AssocGEN analysis engine to determine artifact relationships based on metadata associations. The artifact relationships are indicative of one or more events of interest that may have transpired during the creation and/or modification of the artifacts concerned. During forensic analysis, it becomes necessary to discover such events in the context of an investigation. Often, this can involve pivoting on events specific to one metadata family. For instance, in a collection of digital images, a user may be interested in finding the makes and the models of all the digital cameras used to take digital photographs and corroborate file system timestamps extracted from these digital photographs against their EXIF timestamps. If the timestamp differences between the EXIF and the file system timestamps are large then, this can inform a user if a photograph may have resided on other sources before being created on this file system. On the other hand, if an investigation involved IP theft, the user may prioritize matches based on author and/or owner metadata tags to determine the names of individuals other than the owner and their photographs from the image collection. Similarly, in a collection of word processing documents, supposing the user is interested in identifying all the authors and their organizational affiliations, one can then select the ‘Author’ and the ‘Company’ metadata from word processing documents and analyze the association group generated. The multi pockets, thus generated, represent the sets of documents where the author name and the company name have identical values. The examiner may also refine an initial listing of association groups based on investigation requirements by modifying the set of matches sought by AssocGEN. Such refinements may be required where an initial grouping (using standard classification methods) does not reveal any interesting or anomalous activities. Therefore, AssocGEN allows an examiner to filter the set of metadata associations thereby controlling in the number of metadata matches discovered during analysis.

While AssocGEN focuses on metadata value matches to determine associations between digital artifacts, I discussed the interpretation challenges relating to implementing such matches for the timestamps in metadata in Chapter 2. To address the challenge, I proposed the Provenance Information Model in Chapter 4 which incorporates the timestamp semantics relating to each homogeneous source and allow comparisons for the purpose of developing a unified timeline. In

the sequel, I present my design of the prototype toolkit called UniTIME implementing my Provenance Information Model.

5.2 Prototype Development Two: UniTIME unified time-lining tool

The UniTIME tool was designed to accept the sources of digital evidence as input and convert them into one or more homogeneous sources with corresponding Provenance Information Model information (refer to Section 4.10). The timestamps within and across multiple homogeneous sources are adjusted using the respective PIMs to generate a unified timeline. The contribution of UniTIME is three-fold:

1. Computing unambiguous UTC time values for timestamps by overlaying a PIM to a corresponding homogeneous source
2. Computing location or local time zone information based on a PIM; and
3. Validating timestamp-based assertions that are recorded in a PIM for each homogeneous source.

5.2.1 Design Overview

UniTIME was developed in Java to traverse sources of digital evidence, such as forensic hard disk images, Internet browser logs and network packet captures and harmonize them using provenance information to generate a unified timeline. UniTIME can parse timestamps from file system and document metadata on files, the Internet Explorer and Mozilla Firefox browser history and cache logs, and PCAP packet captures. To parse timestamps from browser logs and network packet captures, I integrated third party applications to export log records and network packet trace as events in XML. The timestamps are then converted to UTC, and validated against *related* timestamps for consistency and sorted to generate the timeline. The relationships are determined based on grouping the events determined through metadata associations. The interpretation logic for acquiring the true timestamp from different homogeneous sources using PIM, implemented in UniTIME is shown in Figure 5.4 which illustrates the time reference embedded in the PIM for each homogeneous source and their respective resolution. Two values, one the UTC timestamp and the other, the local timestamp, are computed. Additionally, provenance metadata of the source, like the tag information [190], are included in the Provenance Information to identify inconsistencies. Tag information included with homogeneous sources in UniTIME includes:

1. date and time of the homogeneous source acquired;
2. size and content list of folders; and
3. total size of each homogeneous source.

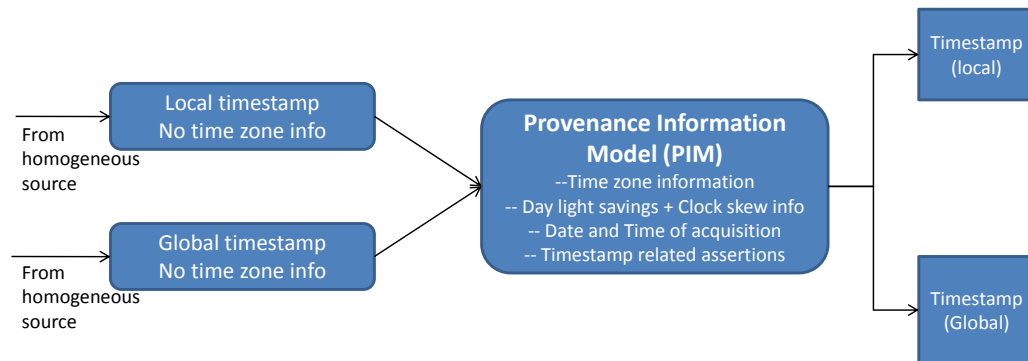


Figure 5.4 Timestamp interpretation logic for the digital time-lining tool

5.2.2 UniTIME tool architecture

The design of the UniTIME tool was based on the *f*-FIA and the functionality of timestamp analysis belongs to the Evidence Composition layer, as a part of the Knowledge Representation and Reasoning sub-layer. The tool traverses the sources and identifies the homogeneous sources from which the digital artifacts and their timestamps are accessed. These artifacts, if extracted, are stored in the *Homogeneous source and Digital Artifact* repository. Where it is necessary to only generate a timeline from the sources, it was sufficient to traverse the artifacts and parse the timestamps from metadata for run-time computation. On the other hand, if it is expected that the digital artifacts would be re-used (or possibly combined) with other information during analysis, then the extracted artifacts are stored into the repository. Separate file metadata parsers, Internet browser history and cache log parsers and network packet parsers were implemented to parse the metadata from the artifacts. If the metadata are expected to be re-used, they are extracted and stored into the *Metadata & Timestamps* repository. For each homogeneous source traversed, a reference PIM is created which stores the relevant information for timestamp interpretation. The PIM is populated from out-of-band information. The UniTIME tool architecture is shown in Figure 5.5.

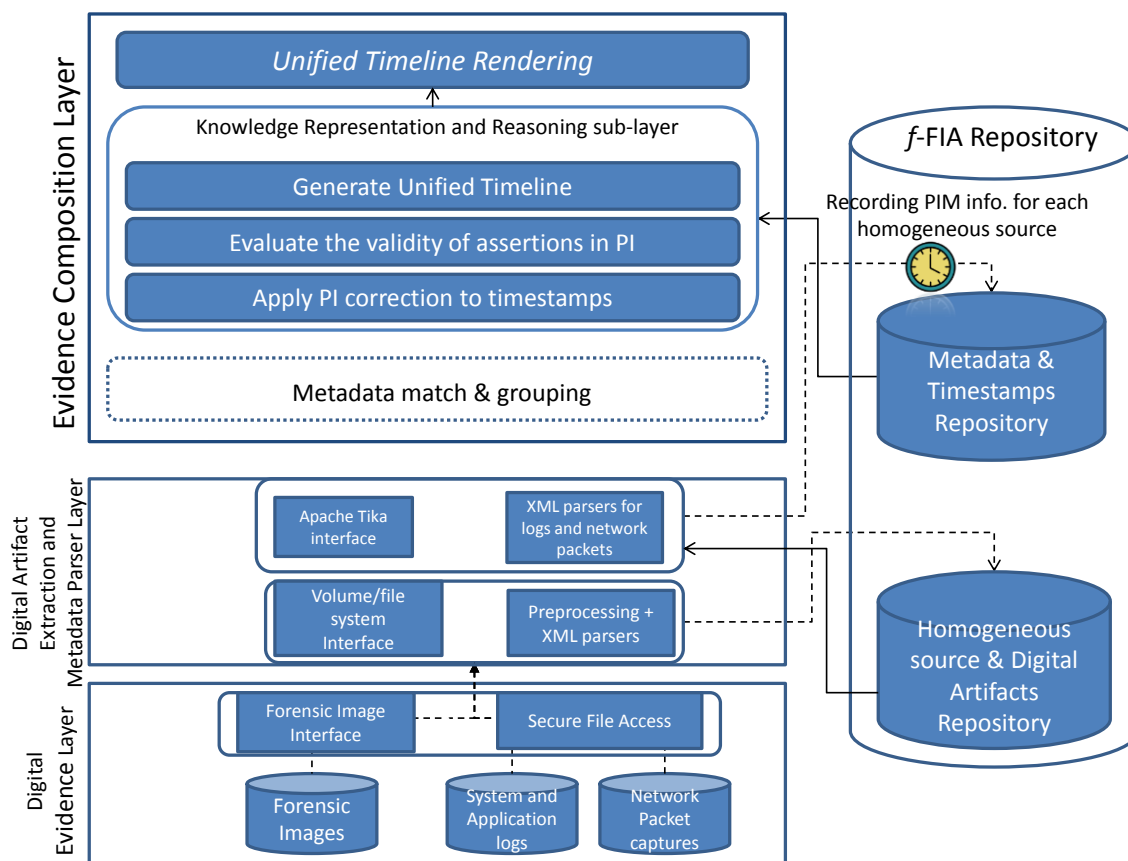


Figure 5.5 UniTIME Architecture based on f-FIA

5.2.3 Dataflow in UniTIME

The timestamp corrections using the PIM were applied as follows: If the homogeneous source was a FAT file system, then the document metadata and the MAC file system metadata are time shifted to denote the time in the local time zone where the homogeneous source was acquired and in UTC. If the homogeneous source was a homogeneous NTFS file system or an Internet browser log, then the UTC timestamp is duly recorded and the local timestamp is computed using its PIM information and validated against the assertions. Files stored in an NTFS file system, which could have originated from a FAT file system or ZIP file archives, are identified³⁹ prior to the timestamp corrections and treated as such. Figure 5.6 depicts the data flow corresponding to the timestamps corrections and validations conducted using the PIM.

³⁹ I applied the hypothesis that timestamps within NTFS/EXT file systems which had 2-second resolution and represented timestamps in even-second intervals are likely to have originated from a FAT file system or a ZIP file. All such files are isolated and a correspondence is established to determine their PI.

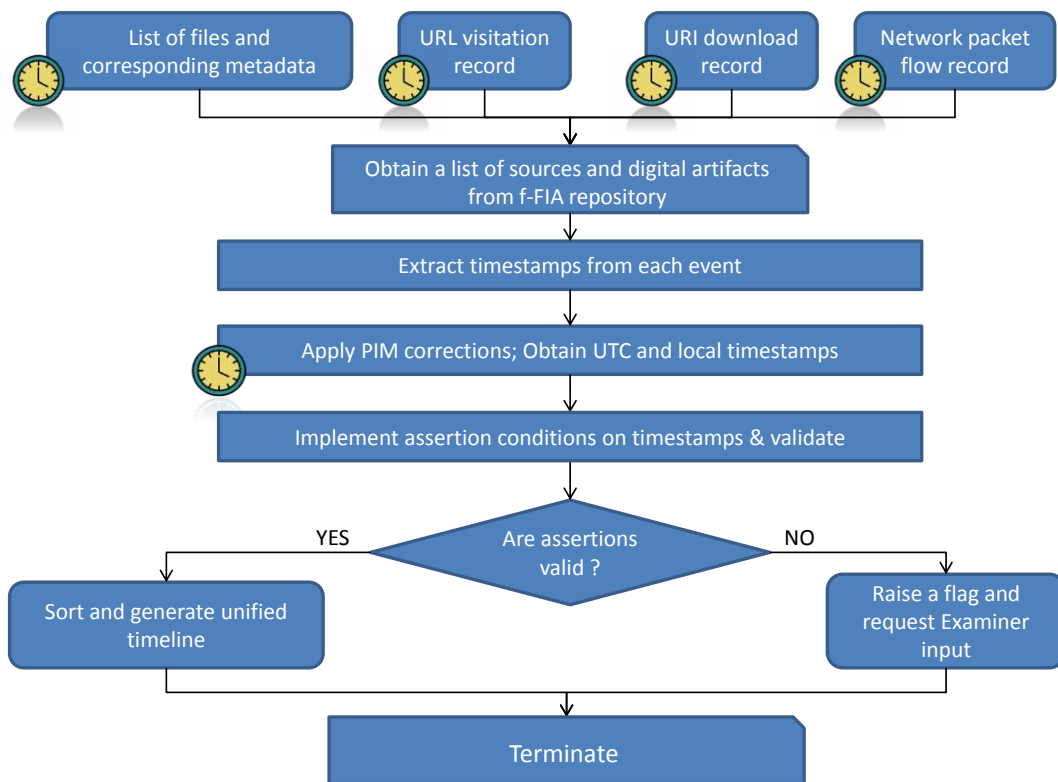


Figure 5.6 UniTIME Dataflow during Timestamp analysis

5.2.4 Maintaining Resilient Timestamps

When the timestamps across all homogeneous sources are corrected, the events are digitally time-lined. The tool provides the examiner the option to view these timestamps in UTC or in a selected local time zone according to its source. Additionally, the tool also provides the examiner the option of choosing to assert statements in the PIM after applying corrections to the timestamps. If the examiner chooses to assert, then the timestamps are validated against the assertions, otherwise the tool proceeds to the sorting of each list followed by the generation of the unified timeline. This ability enables an examiner to initially analyze the timestamps in an unbiased manner and assert afterward, to determine the differences, if any exist. To illustrate this feature, consider a scenario where an examiner is examining a set of emails and some documents from a file system. Let the assertion state, *“the document metadata in documents found as attachments in emails should occur before the corresponding email server timestamps”*. Once the appropriate PIM corrections are applied, the examiner can choose not to assert and generate a timeline of all activities, both the file timestamps and email timestamps from mail servers. While the activities may all *appear* consistent, if the examiner had asserted the statement, the examiner could have discovered that the documents were created after the email was received according to the timestamps in the document

metadata. Such anomalies are flagged by the tool. In the following section, I describe my experimental methodology.

5.3 Mapping Forensic Context into MAM experiments

The MAM discovers associations between digital artifacts that are inherent in the metadata. While traditional approaches for conducting forensic analysis rely on searching and classifying the digital artifacts in the sources of digital evidence, these techniques presume the existence of prior knowledge about the digital artifacts or the nature of an investigation. While that strategy may work within the scope of a focused investigation, during general forensic analysis, it is necessary to discover and report all relationships and higher-level associations that may exist between the digital artifacts. Unless some specifics with regard to the values being searched for or attributes that are likely to demonstrate any unique patterns are known a priori, keyword searches and classification offer limited help. However, even if an examiner does not have all the information needed to conduct the analysis, the metadata about the digital artifacts store this information, if only partially, that can be used to guide the process and extract the relationships.

Naturally, the application of the MAM is more suited to analyzing collections of digital artifacts where only the end goal is known, which is the extraction of all relationships and higher level associations between the digital artifacts, but the means to achieve this is, at best, vague. The absence of any prior information to conduct an analysis makes the MAM readily suited to the analysis of collections of digital image files and word processing documents. Often, such helpful information is only available after the analysis has begun. When a single file or a few sets of files are not available to seed the analysis process, we may apply the MAM to that collection of digital artifacts and group them based on the metadata associations.

5.3.1 Hypotheses for Experimentation

In my work, I evaluated the metadata association model (MAM) by applying the model to specific collections of files and online application logs to identify the origins of files, files that were doctored (in content) and posed as originals and to determine the user activity sequence on an online download session. I conducted these experiments using the hypothesis based testing method [61, 149] as stated in Chapter 3. For my purposes, I developed the following hypotheses to evaluate the utility of MAM using the AssocGEN analysis engine.

1. If we apply the MAM to a collection of files and determine R_s on sets of artifacts, it will give rise to discrete sets of files created by the same source/owner/device/software.
2. If we apply the MAM to a collection of files and determine R_{ss} on sets of artifacts, it will give rise to sets of artifacts structurally identical with regard to file size/content type/encoding structure/file formatting information.
3. If we apply the MAM to user files and application browser logs to determine R_d and R_h on the sets of artifacts, it will give rise to sets of artifacts (in an association group) which when sequenced will identify a particular user's activities tracing the user's browsing sessions and file downloads.

5.3.1.1 Controllability of MAM experiments

In the context of setting up a controlled environment, my experiments were conducted on isolated logical images of a user file system while the network packet captures and web browser logs were exported into XML. In my experiments designed to validate the model, inputs were drawn from digital artifacts across the sources. There were a finite number of metadata in each dataset⁴⁰ and the values taken by each metadata were discrete and finite. Using AssocGEN, I parsed the metadata without altering the integrity of the files. The engine searched the metadata pool for exact and partial matches and grouped the files conforming to each match into similarity pockets. I merged overlapping groups into similarity groups across homogeneous sources on a single physical source and into association groups across multiple physical sources. Redundancy was eliminated by merging the associations that contained overlapping artifacts. The groups were then organized into appropriate relationships using the semantics of the associations. The resultant groupings (in textual form) were presented to us for further input. The inputs can range from specifying a subset of metadata and re-compute the metadata groupings or the identification of one or more association groups that are listed to analyze the relationships embedded in the member artifacts.

5.3.1.2 Metrics and Measurements

From my definition of a similarity group in Chapter 4, I know that it is the set of all those artifacts which are related either directly based on a metadata association or indirectly through one or more artifacts such that transitive closure is satisfied. Besides, I also established that the similarity

⁴⁰ The datasets are described in Chapters 6 and 7 of this thesis

groups within a single source are mutually exclusive. Based on these two properties, to analyze the functional completeness of the metadata association model, I introduced a parameter called the *association index* (ai). The association index ai for a source is defined as the average of the ratio of the size of the similarity group that an artifact belongs to the total number of artifacts on that source. By definition, the range of values taken by ai is $[0, 1.0]$, where $ai = 0$ indicates that the artifact in question is isolated while $ai = 1.0$ indicates that the artifact is highly connected and all artifacts are related to the said artifact. The following relationships hold with regard to the association index ai :

$$0 \leq ai \leq 1.0 \quad \dots (1)$$

$$ai = \frac{1}{\text{number of association groups}} \left(\sum_i \frac{|ag_i|}{N} \right) \quad \dots (2)$$

where $\sum_i |ag_i|$ is the number of digital artifacts in the association groups as determined using digital artifact i as the seed and N is the total number of digital artifacts being considered. In my experiments, on a given source, I computed the association indices for all the artifacts on the source and determine the mean ai value that is assigned to the source.

To study the effectiveness of the metadata associations generated on files and their relative advantage when compared with the traditional techniques for individual file analysis, I define two parameters, the *effort margin* \mathbf{r} and its complement, the *grouping efficiency* $\mathbf{\eta}$ as metrics.

The effort margin is a measure of the fraction of effort as against individual file analyses when conducting a forensic analysis. The effort margin is computed as the ratio of the sum of the number of association groups to the number of groups to be analyzed in the worst case⁴¹. The value ranges from 0 to 1, where 0 represents zero effort for the examiner and 1 represents effort identical to that which is necessary to carry out the task of individual file analyses using traditional forensic tools. The effort margin can take a value 1 if and only if all the digital artifacts remain unassociated after applying the model, leading to a separate group for each digital artifact. The effort margin can take a value 0 only theoretically since the least value for the numerator in the ratio is 1 which results when all the digital artifacts get grouped into one association group.

⁴¹ In the worst case, the number of association groups equals the number of digital artifacts in the source.

The grouping efficiency is a measure of the degree of closeness between the digital artifacts in digital evidence, across all sources. It is computed as $1 - \text{effort margin}$. The value for grouping efficiency ranges from 0 to 1, where 0 represents that no association groups were generated, implying that all the artifacts remained unassociated, while a value of 1 represents that all the digital artifacts were grouped together. The grouping efficiency can take a value 1 only theoretically since the effort margin can only take non-zero values in practical scenarios. In short,

$$\text{Effort margin } \mathbf{r} = \frac{\text{number of association groups}}{\text{number of association groups in the worst case}} \quad \dots (3)$$

$$\text{Grouping efficiency } \boldsymbol{\eta} = 1 - \mathbf{r} \quad \dots (4)$$

The effort margin should be interpreted as the fraction of full effort necessary to analyze the digital evidence after applying the MAM. The full effort is deemed to be applied when the digital artifacts are analyzed individually. The grouping efficiency values should be interpreted as the percentage reduction in volume of digital evidence resulting from the application of the MAM to determine metadata based associations. The digital evidence is deemed to be at full volume when all the digital artifacts are unassociated.

Although these metrics are defined using association groups which apply to the groups of digital artifacts across sources, without loss of generality, the same definitions can be interpreted even within a single source by replacing association groups in the ratio with similarity groups related within the source of digital evidence concerned. When this modified definition is applied to each individual source of digital evidence, it can provide a forensics examiner an assessment of the total effort involved in analyzing all the digital artifacts contained in that source, based on the degree of closeness exhibited by the artifacts.

5.3.1.3 Reproducibility of MAM experiments

For my experiments, each dataset (described in Section 6.4 and Section 7.3) was imaged (logical image) using the FTK imager tool and stored on the computer executing the AssocGEN analysis engine program. The digital evidence layer that can access evidence in raw format provided a handle to access this source. The digital artifact traversal and metadata parser layer traversed the digital artifacts (files) and parsed the metadata. The evidence composition layer provided us with the option to select the metadata tags thereby allowed us to main control over the number of associations sought and determined. After determining the metadata matches, the similarity

pockets were aggregated into similarity groups and associations groups. When I sought a particular file, the tool automatically listed the *associated* digital artifacts on a console and highlighted the nature of the association with the selected file. When I selected a file from the image, all the associated files were listed on a textual console. Figure 5.7 illustrates such a grouping on digital image files achieved using AssocGEN. The digital images shown in the figure were grouped based on the structural similarity relationship R_{SS} identified using the application family metadata pertaining to image dimensions.

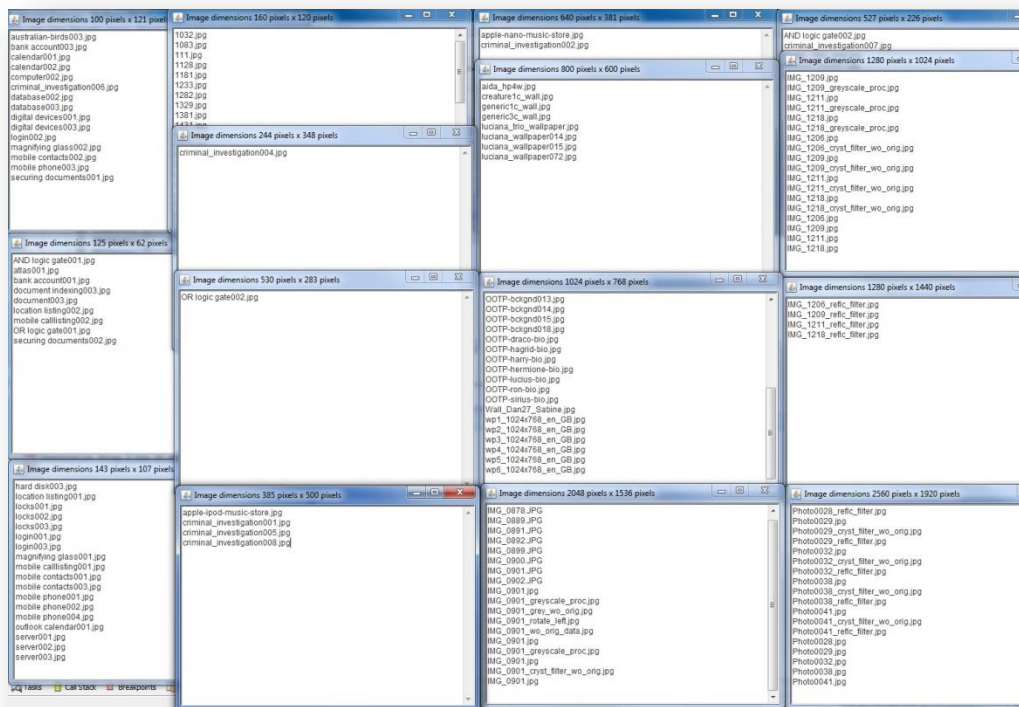


Figure 5.7 Example of file grouping on digital image files

After each experiment, I loaded the source on FTK 3.2 and examined the metadata in the user view mode. For each file suggested by the tool based on the metadata groupings, I compared the respective file metadata to verify the groupings. When file system metadata did not produce matches, the files were exported to the local file system and examined using FTK 3.2.

In the sequel, I motivate the need for analyzing digital image files and word processing documents using the AssocGEN analysis engine to elicit answers relevant to the six forensic questions, viz., *who, what, when, where, how and why.*

5.4 Forensic Analysis of Digital Images & Word Processing Documents

Digital image files are popular and it comes as no surprise that they form the subject matter of many digital investigations. When analyzing a collection of digital image files, one may begin with an image chosen at random and methodically analyze all related image files. Typically, the images in the collection can be classified in different ways to determine “related” image files. Such classification techniques can include, source, scene, time-instants, image dimensions, and so on. Unless the images in the collection were systematically captioned, keyword searches may be of little use.

Forensic analysis of documents is a critical component to the process of forensic reconstruction of activities, especially on Microsoft Windows based computers. More than 80% of the world’s computers run the Windows operating system [198] and a considerable number of these use the Microsoft Office document suite. Therefore, it comes as no surprise that Windows based documents and in particular, Microsoft Office documents are commonly encountered during investigations. When analyzing a collection of documents, one may initially identify a subset of documents based on some keywords. Using the outcome from file analysis, one can identify further keywords or contents based on which other related documents are traversed.

Traditional methods presume the existence of some prior knowledge about the files and hence cater to that presumption. Hence, one may not be able to determine the scope for all types of patterns that can be determined on a given collection using these traditional techniques. In situations where the presumption may not be relevant, this presumption has a tendency to mislead the analysis. It is therefore necessary for a grounded approach which evaluates the scope for analysis and establishes a framework to determine all patterns thereof. I demonstrate the application of MAM-based analysis to digital image files and word processing documents in Chapters 6 and 7. I frame one or more of the six forensic questions into MAM-based experiments and generate association groups based on metadata value matches using my AssocGEN analysis engine.

5.5 Chapter Summary

In this chapter, I presented my approach to designing practical experiments to evaluate the models proposed in my research. I developed three hypotheses to verify my proposed models, viz., the metadata association model implemented in the AssocGEN analysis engine and the provenance

information model implemented in the UniTIME timeline analysis tool and discussed the verification of my prototype implementations. I presented the design of two research prototypes, the AssocGEN analysis engine and the UniTIME unified time-lining tool and discussed how they implemented the MAM and PIM respectively. In the next two chapters, I demonstrate the use of the models through experiments using my prototypes. I discuss the characteristics of my datasets and rationalize their utility in regard to my experiments and present the insights gained by applying my approach.

“There is no such thing as a failed experiment, only experiments with unexpected outcomes.”

- Richard Buckminster Fuller

6. MAM Based Analysis of Digital Images

In this chapter I focus on the application of my Metadata Association Model to collections of digital image files to elicit metadata based associations for analysis. I assume no prior knowledge of the digital image collections in my experiments in the application of the MAM. The experiment demonstrates the functional completeness of the model in two modes of operation: determining *need-based* and *exhaustive* image file associations.

6.1 Classification vs. Association

When analyzing a collection of digital image files, a typical analysis can involve classification. Image classification is of many types, image source-based, image dimension-based, digital camera-based, image timestamp-based, and so on [17, 18]. Source based classification, for instance, will decompose the collection into sets of digital photographs, edited photographs and digitally generated images. This process enables a forensics examiner to group similar or homogeneous image files so that they may be analyzed together. Traditional classification uses single or multiple parameters based on which digital image files are grouped for analysis [16-18]. However, such parametric classification is mostly syntactic and often the burden of determining related digital images falls on the individual. This task requires comparison of different classes which can involve the images being re-classified several times, using different sets of parameters,

to determine in how many distinct classes an image file may be classified. Often, the knowledge and experience relating to different types of classification which are likely to reveal such insights is not readily available [67].

Association based analysis focusses on identifying those digital images which are likely to occur across such groups and is not merely bound by the rules defined in traditional classification. This is illustrated in Figure 6.1. On the same collection of digital image files, while a classifier may take a classification parameter as the seed input around which to build a cluster of files, the MAM approach does not need any such input. Besides this, whereas a classifier may give rise to image classes containing similar image files that are homogeneous with regard to that classification parameter, the metadata association instead generates groups⁴² that contain image files “related” based on their metadata.

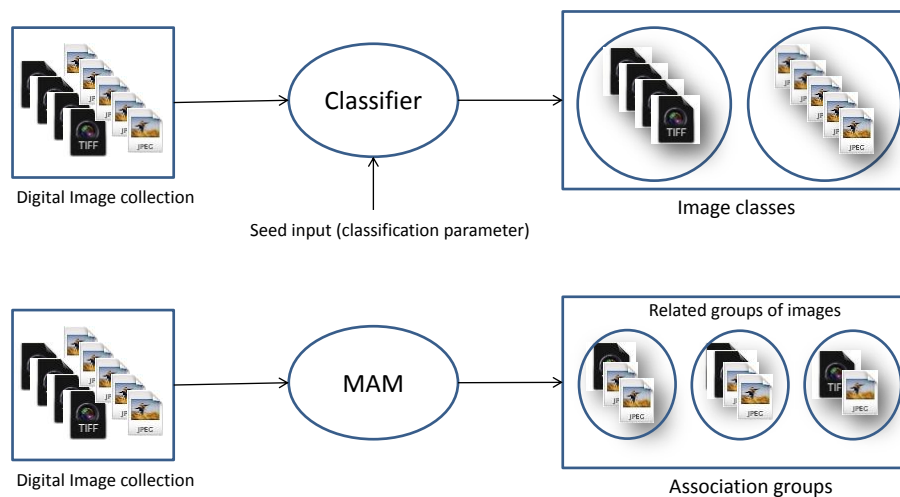


Figure 6.1 Illustrating the differences in Image classification vs. association

By identifying images on a given grouping that are “associated”, the MAM allows one to explore the scope for related or similar images when analyzing within a context. In my evaluation, I develop the context by classifying the digital images in each collection according to its source. I classify digital images in a given collection as:

1. *Digital photographs;*
2. *Software processed or edited photographs;*

⁴² While one may perceive digital artifacts being grouped based on metadata values as a classification process, the ability to group three or more digital artifacts together such that each pair demonstrate a different metadata match between them can be achieved using the MAM but not using any single classification process.

3. *Digitally generated images*; and
4. Images with *incomplete image metadata*.

Once these classes of digital images are determined, I identify metadata associations using value matches between the digital images across all the classes to form association groups. The association groups are then analyzed with regard to the six questions identified by Casey [32] that are relevant during forensic analysis.

6.2 Conducting Forensic Analyses on Collections of Digital Images

When analyzing a collection of digital image files as part of an investigation, many forensic questions can be raised during the analysis, of which some are listed below:

1. How many digital cameras can be identified from the digital image metadata? How many image files belong to each of these cameras?
2. How many digital photographs are doctored? How many Internet downloaded images are doctored? What photo-editing software was used?
3. Are there other “similar” digital image files without source metadata? How can such related digital image files be identified?
4. Which of the digital image files were downloaded from the Internet? If so, can the source of these image files be determined?

While some of these questions can be answered in part or whole using traditional classification, often it is up to an examiner to analyze the individual classes to identify inter-image relationships. I believe that identifying such relationships can be useful to a forensic examiner during a digital forensic investigation. To determine answers to such questions, it is necessary to recognize that no single classification method can provide all the answers and it is necessary to determine relationships between the images to extract all higher-order associations that exist both within a particular source class and across such classes. Such a task requires exhaustive classification using all individual parameters (from metadata) as well as all combinations of multiple parameters to determine where the digital image files overlap and group them. The association groups generated from the MAM, on the other hand, achieve this task readily and simplify the task of identifying related images to a search task within an association group. Through its automation, the MAM integrates this task and eliminates the need to manually identify such related images.

6.3 MAM Based Analysis of Digital Image Collections

The Metadata Association Model is intended to identify image files related through metadata and group them together. In the absence of a specific context, the MAM can be evaluated based on identifying quantifiable metrics identified in Section 5.3. When a particular context is provided, the MAM can be configured to seek specific patterns that extract the specified relationships inherent to the collection of digital image files. I demonstrate the two modes of evaluation in my experiments discussed in Section 6.5. In general, when applying the MAM to a digital image collection for analysis, there are two modes of operation, viz., need-based and exhaustive.

1. *Need-based.* Given a single digital image file, it is always possible to exhaustively list all digital image files in a collection that are associated on metadata.
2. *Exhaustive.* Using the MAM, it is possible to determine all metadata associated digital image files in the collection.

We demonstrate both modes of operation using the MAM. In the need-based analysis mode, a forensics examiner may identify a small set of digital images and each image is then used to identify a chain of related digital images based on the metadata associations identified in the image collection. This would be suited to tracing the origin of digital image files if there is suspected online activity. The exhaustive analysis mode is suitable to group digital images in an arbitrary collection where a specific starting point for the analysis is unavailable. In this case, the digital image files are grouped first, based on their metadata, and the groupings are then used to guide the analysis. To achieve this, I have identified multiple collections of digital images obtained from different digital still cameras, images edited using photo-editing software and those downloaded from the Internet. Digital image files belonging to these classes vary in the number of metadata that affect the number of the associations that can be determined between them.

6.3.1 Criteria for Selecting Digital Image Collections

There are many criteria that govern the identification of associations in digital image files. In my work, I am concerned with the associations that exist in metadata. As discussed in Section 4.7, metadata can be classified into four families that are relevant to forensic analysis, source, ownership, timestamps and application; each type can contain one or more individual metadata that can produce matches leading to metadata associations between the digital image files.

The association index ai (defined in Section 5.3) is a measure for determining the quality of metadata associations that can be derived out of a given dataset. The index provides an estimate of how “connected” a dataset is and is given by the mean of the association index computed for all the image files in that dataset. An image file containing a large number of metadata is likely to generate a large number of metadata associations and is likely to be highly connected. On the other hand, an image file which contains few metadata may give rise to only a few metadata associations and thereby be less connected. An image file which does not generate any metadata associations is therefore “unconnected”. It is noteworthy that the association index for a given collection is correlated with the grouping efficiency η . The higher the value of association index ai , the higher the value of grouping efficiency η .

To apply the MAM and analyze collections of digital image files for the case studies described in this chapter, it was necessary to identify digital image collections (experimental datasets) which span the spectrum of highly connected to less connected. Typically, digital image files that were captured using one or more digital still cameras may generate many metadata matches including those that pertain to *source*, *ownership*, *timestamps* and *application*. Consequently, collections containing such digital images are likely to be highly connected. On the other hand, collections where the digital image files were downloaded from different sources, e.g., downloaded from different websites while browsing, can produce very narrow groups of image files and hence are likely to result in less connected digital image files. Besides this, if such image files do not contain application metadata, the scope for finding such associations is further reduced.

In order to determine answers to the questions we’ve posed in Section 6.2, the digital image files in a dataset are required to have certain properties in regard to their metadata. These properties are as follows:

1. At least one metadata from each metadata family should be available:
 - a. Metadata identifying one or more digital still cameras and/or computer software;
 - b. Metadata pertaining to the format and structure of the digital image; or
 - c. Metadata pertaining to time instants when specific events affected a digital image file.
2. Digital artifacts referring to the same instance of a digital image must demonstrate *existence* and *source* relationships.

3. Digital image files with identical or similar filenames being stored in different file formats must demonstrate *existence* and *source* relationships.
4. References to digital image files across log files must demonstrate a *happens before* relationship.
5. All available file system and application metadata must be authentic.

Corroboration of related digital image files has been an integral part of forensic analysis [26, 28, 50]. In order to identify image files and corroborate, the dataset must contain digital image files with identical source metadata information (one or more metadata values corresponding to the source metadata can be equal, threshold association may also hold). When two or more digital image files with identical source metadata values are stored in different formats (as regular or backup or temporary files), the files must demonstrate the existence R_e and source R_s relationships (metadata such as filename, file location, computer software). During analysis, it is customary to determine the sequence (timelines) of events involving the digital image files. To achieve this, a digital image file must support at least one timestamp metadata.

6.3.2 Metadata & Metadata Families in Digital Image files

We identify the digital image metadata at their respective metadata families relevant during forensic analysis in Figure 6.2. A collection of digital image files can be organized according to the image file names and their respective locations on a particular source of digital evidence. The metadata that allow one to do that belong to the *source* metadata family. Another metadata pertaining to this family, viz., ‘software’ metadata is usually found in digital images if the images were edited. When this metadata value is present and there are no discernible EXIF markers, it could indicate a digitally generated image file.

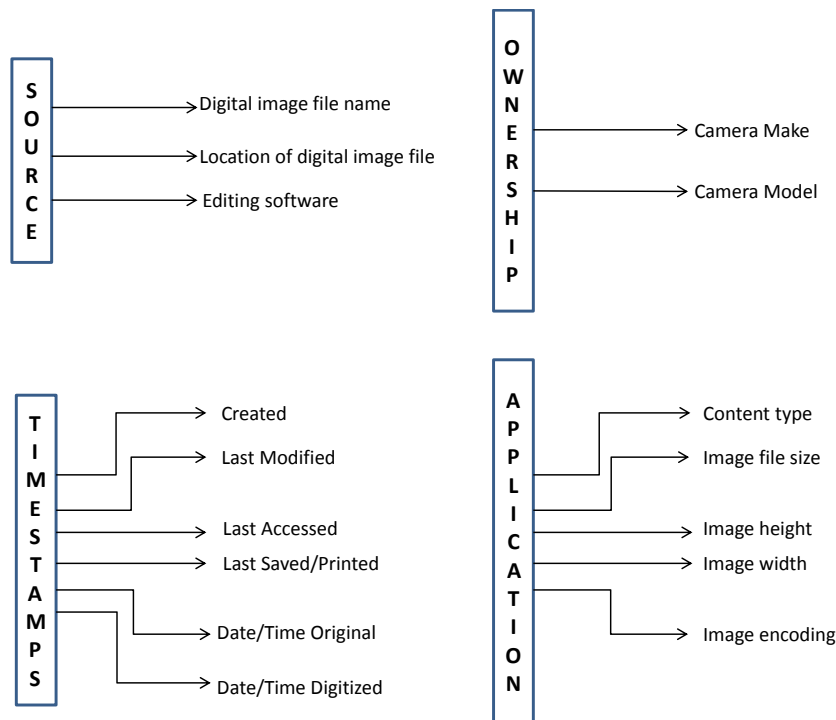


Figure 6.2 Digital image metadata tags of interest in Digital Investigations

Digital image files also must be identified based on the device used to record or capture the digital image files and the metadata that allow us to do that are the EXIF metadata Camera make and model metadata tags. The EXIF metadata in the digital image files store information about the digital still camera and technical details about how a digital photograph was captured. Such groupings not only identify all the cameras used in generating the collection, but they can be used to identify the number of digital images generated by a camera of a particular make and model. These metadata belong to the *ownership* metadata family.

The MAC timestamps and the EXIF timestamps, where available, belong to the *timestamp* metadata family and identify events corresponding to creation, modification and access of the image files.

Image dimensions can help one gauge the granularity of digital image files and is a useful pre-analysis metric; the higher the image dimensions, the better the level of detail in the image file. Such metadata and those such as image file size and image content type that provide information regarding the features of digital image files belong to the *application* metadata family.

Digital image files do not store author information; rather they record the details pertaining to devices such as digital still cameras, computers and computer-based software used in creating or

editing these images. As a result, the software and camera devices are identified as source and ownership information pertaining to namesake metadata families in my experiments.

6.4 Datasets

While examining a source of digital evidence for digital images, an examiner is likely to discover images from different sources, viz., images recovered from carved data [50], images that are digital photographs, images edited or digitally generated using software and images downloaded from the Internet. These different types of digital images are shown in Figure 6.3.

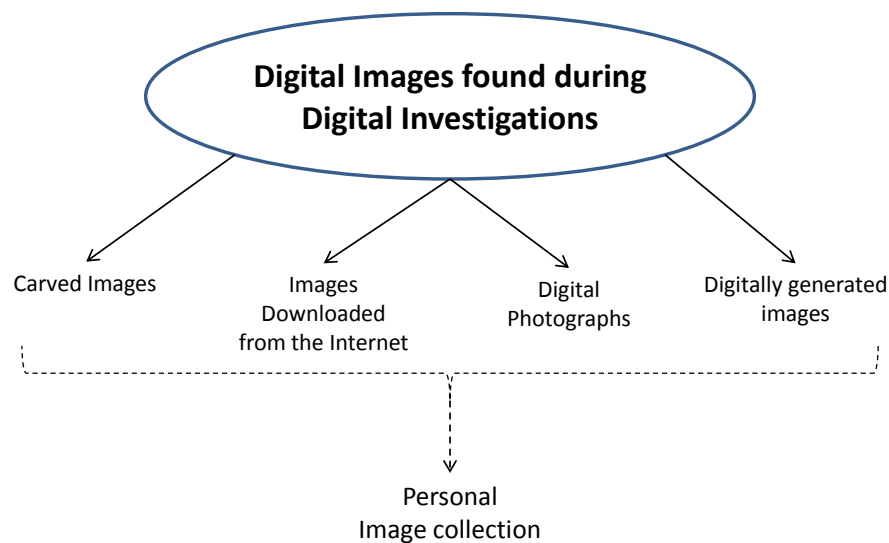


Figure 6.3 Different probable sources for digital images discovered in digital evidence

Each collection of images has a different level of metadata associated with it that can either enhance or impede the grouping. Usually, images from carved data have incomplete or no metadata and hence a grouping based on metadata is likely to result in a high effort margin and low grouping efficiency. Images from the Internet can be downloaded in several ways and popular methods include downloading images from Google image search results and downloading compressed archives from where the images are then extracted. While the Google database may not include image metadata unless it is voluntarily provided during uploading, archives usually omit image metadata during compression. As a result, the chances that metadata is present in such images is likely to be low, which could also lead to a high effort margin r and low grouping efficiency η . Images that are digital photographs store a variety of metadata provided by digital technology for better management. As these images are rich in metadata, they are likely to result in low r and high η . Digitally generated images and those edited by software are increasingly

storing valuable information in the image metadata and hence fall in the same category for \mathbf{r} and η . In any personal collection, the images found are usually a mixture of digital images across such sources, and hence the grouping efficiency is determined by the majority fraction of image sources.

6.4.1 Digital Image Datasets

We describe 5 datasets of digital image collections acquired from various sources to conduct my experiments and generate association groups.

6.4.1.1 Drew Noakes Digital Photograph Collection

The Noakes digital photograph collection [142] can be characterized as a *personal collection* that contains 126 digital images of which 124 are digital photographs taken with over 20 different digital cameras. Of these 124, 7 digital photographs were processed using Adobe Photoshop 6.0. Noakes had downloaded the remaining two image files from the Internet and did these files did not contain camera related metadata.

6.4.1.2 NPS-Canon-Images: Digital Corpora Collection

The NPS-Canon-Images can be characterized as a *carved image collection* that was obtained from Digital Corpora [68]. It contains a set of 6 digital forensic images⁴³ containing 52 JPEG images⁴⁴ created as an exercise for image carving and analysis. Of these, 34 images can be fully recovered with valid metadata to perform association grouping. The remaining 18 images do not have sufficient metadata to lend themselves suitable for association grouping and were hence discarded. Each valid image is a still shot containing a screenshot of text either on a Mac computer or writing on a piece of paper. The image dimensions vary from 640×480 to 3072×2304 and the image file sizes range from 103 KB to 2.70 MB. All the images in this dataset have been captured from a single camera, viz., a Canon Powershot SD800 IS. All images were recorded on the aforementioned camera from the afternoon of 23rd Dec 2008 to late on 24th Dec 2008. The image resolution along the vertical as well as the horizontal is 180 dots per inch. None of the images have been modified by image processing software⁴⁵.

⁴³ This was a forensic digital image of the 6 different sources obtained from Digital corpora. Each forensic image was a raw image of the file system from a single digital camera in which all digital photographs were taken.

⁴⁴ Downloaded from URL <http://digitalcorpora.org/corp/images/nps/nps-2009-canon2/>

⁴⁵ After corresponding with the author of the Digital Corpora collection, I concluded that these relatively high-resolution digital photographs were not part of any particular digital investigation and were created as basic image forensic exercises.

6.4.1.3 Assorted Image Collection

The assorted image collection can be characterized as a *personal image collection* that contained 491 images from a volunteer's laptop. There are 62 original camera images taken on two different cameras, viz., 34 from a Canon Powershot A400 and the 28 from a Samsung SGH F480 mobile phone camera. All images taken with the Canon camera are set to image dimensions 1280×960 (1.2 MP) while those taken with the Samsung camera are set to 2560×1920 (5 MP). The image file sizes on the Canon camera range from 300 KB to 1.62 MB while those taken with the Samsung camera range from 1.26 MB to 1.42 MB. These digital images were genuine digital camera images and did not contain the metadata tag 'software'. I intended to study which operation and what software introduces the "software" tag in a camera image. I identified Adobe Photoshop, GIMP, IrfanView, Paint.NET, Photoscape and Photostudio in my experiments and selected a set of 5 images at random from each camera and subjected them to a set of image transformations. Each image was subjected to lateral rotations, grayscale representations and a Gaussian filter by the different photo editing tools. This exercise generated a total of 250 images which were also part of this dataset.

The remaining 179 images were downloaded image files. Of these, 55 images are screensaver images downloaded from the WarnerBros Harry Potter website [84] and among these 12 are computer generated images. Their dimensions range from 800×600 to 1050×800 and the image sizes range from 55 KB to 600 KB, and one computer generated image is 2.25 MB. There were another 124 images which were downloaded from the Internet in response to Google Image search queries concerning mobile phones, digital cameras, flash drives, computers, laptops, rack storage and Australian birds. These images ranged in file size from 2 KB to about 160 KB.

6.4.1.4 Govdocs1: Digital Corpora Collection

The Govdocs1 collection can be characterized as a *downloaded image collection* that contains 2157 digital images. It was obtained from the Digital Corpora [70, 71] repository and had 1000 folders each containing 1000 commonly found files such as documents, image files, text files, and HTML pages, from which I filtered the digital image files for my experiments. Of these 1000 folders, I selected the first 10 folders (from 000 to 009) and 6 subset folders (from subset0 to subset5) and filtered only the digital image files⁴⁶. The digital images in this collection were all

⁴⁶ The statistical characteristics of the collection are given at <http://digitalcorpora.org/corpora/files/govdocs1-simple-statistical-report>. Each folder in the repository is a random collection of files that are statistically congruent with the file distribution in a "regular" user workstation. Hence, the set of images in this dataset were representative of the entire collection and sufficient for my purposes to demonstrate the metadata association model.

downloaded by Garfinkel [70] from different sources from the Internet in response to Google Image search queries covering several topics and range from a 1 KB to a few hundred kilobytes in size. Of these 2157 digital images, 1891 have been edited on different versions of the Adobe Photoshop software, 124 are thumbnail images and none are computer generated images. Of the 1891 digitally edited images, 207 were digital camera photographs taken using 7 different camera makes and 20 different camera models. All the image files in the collection had file system metadata as provided by the host workstation.

Since all the images from this collection were downloaded from the Internet and did not generate sufficient number of metadata matches using EXIF metadata, I focused on highlighting the association groups generated based on image file sizes and the JPEG image dimensions, where available. Thumbnail images can be classified according to the NLA guidelines [135] to identify and separate the thumbnail image files. In my evaluation of the datasets, I have imposed an additional criterion that such an image file shall be less than 10 kilobytes in size.

6.4.1.5 Dresden Image Database

The Dresden image database [79] can be characterized as a *digital photograph collection* that contained 8896 digital photographs at the time of the download⁴⁷. This database was created for the purposes of forensic investigation of digital still camera based photographs. These digital images have been taken with over 20 different camera models. Multiple cameras of the same make and model are also used to account for device variations and 36 devices have been used in all. Complete characterization of the digital images in this image collection is given by Gloe and Bohme [78].

6.4.2 Dataset Characteristics

A summary of all the digital images datasets is presented in Table 6.1. It lists the metadata and the number of digital images in each dataset which contained these metadata. Additionally, it lists the images that contained image dimensions and MAC timestamp information. This characterization is used as reference in the following chapter to evaluate the accuracy how the association groups adhere to these categories.

⁴⁷ At the time of download, the web repository was being updated. Since then, the repository has grown to 16,384 digital photographs across 72 camera models. The latest statistics from the analysis are available on request.

			EXIF Metadata		JPEG Metadata		File System Metadata		
Serial No.	Dataset Volume	Number of images in the dataset	Digital Camera make and model	Date/Time Original	Image dimensions	Software tag	ONLY file system meta-data	MAC time-stamps	File size
1	374 MB	126	124	123	126	7	2	126	126
2	126 MB	52	34	34	34	none	none	34	34
3	1.6 GB	491	312	153	491	53	179	491	491
4	6.8 GB	2157	207	205	2157	1891	1891	2157	2157
5	24.2 GB	8896	8896	8896	8896	none	none	8896	8896

Table 6.1 Image characteristics of the five datasets

6.5 Conducting the Experiments

In this section, I describe two experiments used to analyze a collection of digital images. The first experiment applies the need-based analysis method and the second experiment applies the exhaustive analysis method.

6.5.1 Determining the Provenance of Downloaded files

In this experiment, I developed a systematic method to identify the provenance of digital images downloaded from the Internet. Given a user's file system, browser history and cache logs and emails, determining the origin of the files discovered from the sources of digital evidence is a non-trivial task. Using my Metadata Association Model, however, I can group heterogeneous digital artifacts belonging to different sources of digital evidence together. Unlike classification, metadata associations, derived though metadata matches on the digital artifacts, can reveal certain higher-order relationships which can be used to determine the origin of a particular file. Using this method, the file in question is tracked from the user file system under examination to the different logs generated during online user activity to its point of origin in the Web. Since traditional forensic tools can find pieces of evidence for extraction, this methodology proposes a significant improvement in conducting and automating forensic analyses which have thus far been in the realms of human investigation and analysis. This experiment was demonstrated using Dataset 5 summarized in Table 6.1. The image files in this dataset alone contain existence, source, and download metadata and the happens before relationship that was necessary to establish the image

files as downloaded resources and then trace their origins. If suitable sources were available for the other datasets, this can be repeated although similar results are likely to be observed.

Given a snapshot of a user's file system, it is necessary to determine the origin of the files discovered. Figure 6.4 displays the *Downloads* folder on the user file system where I am interested in the origin of the digital image highlighted.

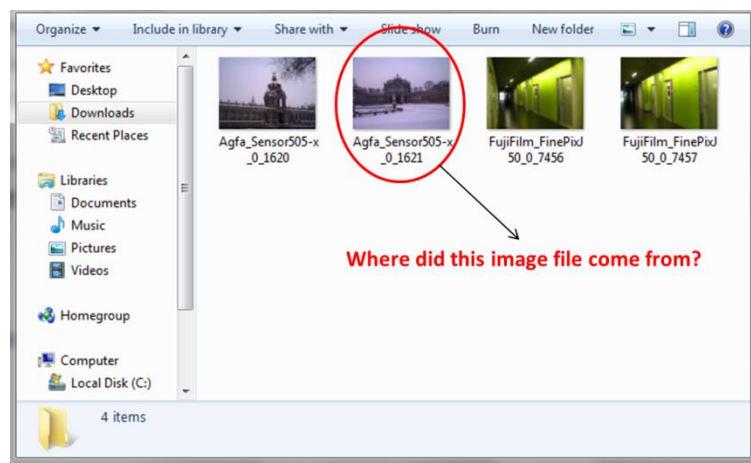


Figure 6.4 Snapshot of the user's file system containing some digital image files

Since it was likely that the image files in question were downloaded during some form of Internet activity, the remaining file system was searched for files whose filenames resemble them. In this case, I discovered the presence of a copy in the temporary files folder corresponding to the user's Internet Explorer browser activity. Figure 6.5 illustrates the discovery of an identical copy of the image file in the temporary files folder.

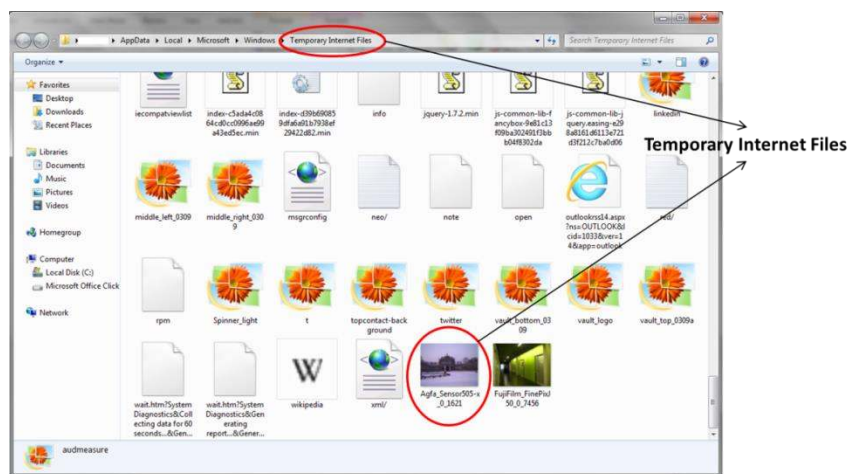


Figure 6.5 Snapshot of the user's temporary internet files

Having determined the existence of at least one file in the temporary Internet files folder, I extracted the browser cache, sought file matches and determined their respective attributes, as metadata. Figure 6.5 establishes the relationship between the resource discovered in the browser cache and the file discovered in the temporary Internet files folder.

6.5.1.1 Method

The method followed was as follows. Set up a virtual machine with Windows 7 operating system and create a user account. Generate the following constructed scenario.

1. Login to the user account and set up a user email account.
2. Capture steps 3-5 concerning user's browsing activity using Wireshark network packet capture.
3. Using the Internet browser, browse the Internet, arbitrarily choose a website and view the images on that website.
4. Download images to the user's computer.
5. Access user's email and view the messages. View the attached images using the browser and download attachment files to the user's computer.
6. Isolate the computer and create a virtual machine snapshot of the user file system. Isolate the Internet browser history and cache logs for analysis.
7. Examine the sources using traditional forensic tools. Use FTK to examine the file system forensic image. Use web analysis tools to examine the Internet browser logs.
8. Determine the origin of the images discovered on the user's computer. Corroborate the results of the web analysis tools against the packet capture.
9. Independently, use AssocGEN tool and load the different sources.
10. Traverse the user's computer using AssocGEN and determine the files containing Existence relationships.
11. Determine Download relationships on these files and establish Happens before relationships from the log source.
12. Using metadata associations, identify all relationships to determine the source of the files. Generate final groupings for analysis.

Repeat the steps using a different Internet browser.

6.5.1.2 Basis for the experiment

During a forensic examination, when a computer is identified, the traditional approach maintains a monolithic forensic image of the computer [171]. This forensic image is examined using a forensic toolkit like FTK to ascertain the file system's contents. Once the contents are ascertained, each file is individually analyzed and its metadata is examined. By virtue of the monolithic nature, the files are examined in isolation and unless the origin of the file is stored in the file metadata, it is likely to be missed. Besides this, the user's Internet activities can only be deciphered when the browser history file is examined. Since history files only record web access records, unless the forensic examiner simultaneously searches the browser cache records and compares it against the files in the user's computer, the origin of the file cannot be determined.

On the other hand, the AssocGEN tool, by design, segregates user documents, temporary Internet files, system and application logs including browser history and cache logs, and network traces as distinct sources. Since AssocGEN is developed based on the MAM, multiple digital artifacts can be accessed to determine associations. The associated artifacts are then grouped together, irrespective of the source they originated from. The dataset used in this experiment is summarized in Table 6.2.

Characteristics	
User files	47, 699 (30 GB)
Temporary Internet files	8916
Browser history	115832
Browser cache	128624
Network packets in trace	35035

Table 6.2 Summary of the evidence analyzed and their characteristics

While FTK treats browser logs as mere files as identified in my review of forensic tools in Section 4.1, AssocGEN treats them as independent user activity logs and enables the identification of events that occurred on the file system affecting one or more files. Using the MAM, the log records are grouped with the related files tracing the event sequences to help an examiner.

6.5.1.3 Observations

We use AssocGEN to first process the user file system and load the files and their metadata (after parsing) into the *f*-FIA repository. Figure 6.6 is a snapshot of AssocGEN loading the user files created using the procedure listed in Section 6.5.1.1 into the repository.

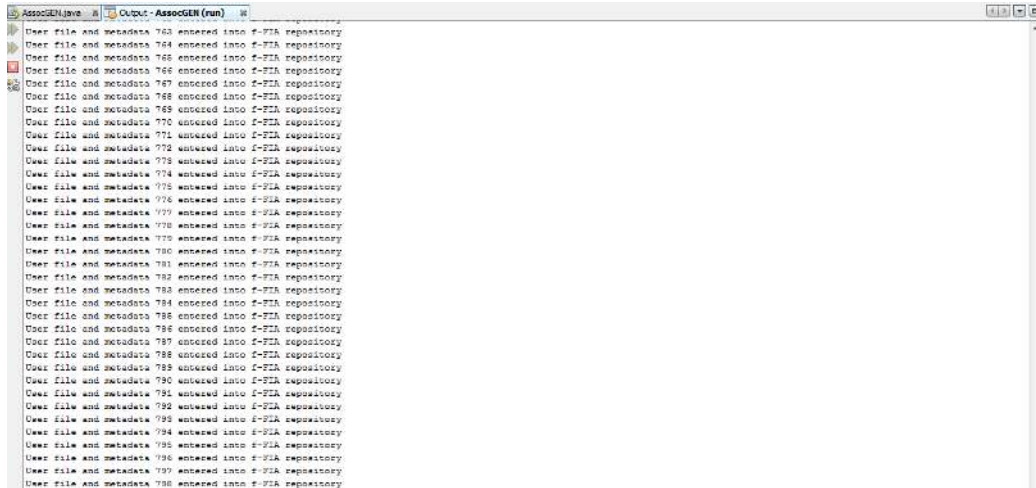


Figure 6.6 AssocGEN analysis engine processing a user file

Once the user files are completed, AssocGEN tracks all Internet based activity which includes traversing the temporary Internet files folder used by the web browser to temporarily store downloaded web resources. Figure 6.7 is a snapshot of AssocGEN traversing the temporary Internet files before parsing the metadata and loading them into the *f*-FIA repository.



Figure 6.7 AssocGEN processing temporary Internet files

After the files are processed, AssocGEN extracts the browser history and cache events which are, likewise, loaded into the repository with their respective attributes. After this, the analysis engine generates all metadata associations. Once the associations are generated and stored into the repository, it then discerns the relationships that exist among the associations which can provide the origins of the image files in question. Figure 6.8 is part of the Java source code of the

execution logic in AssocGEN used to determine metadata associations in evidence followed by the extraction of relevant relationships leading to the determination of the origin of the image files.

```

46
47 public static void getRelationshipsInEvidence () {
48     // determine existence relationships on files
49     Relationship.determineExistenceRelationshipsonFiles(fileEvent.getTempInetFileList(), fileEvent.getFileSystemEventList());
50     Relationship.displayRelationshipsonConsole ();
51
52     // determine happens relationship on logs
53     Relationship.determineHappensRelationshipsonLogs(ievt.getIECacheEventList(), ievt.getIEHistEventList());
54     Relationship.displayRelationshipsonConsole ();
55
56     // determine download relationship on cache logs and temp inet files
57     Relationship.determineDownloadRelationship(ievt.getIECacheEventList(), fileEvent.getTempInetFileList());
58     Relationship.displayRelationshipsonConsole ();
59 }
60
61 public static void main(String[] args) {
62     // TODO code application logic here
63
64     fileEvent = new FileSystemFileEvent ();
65     ievt = new InternetEvent ();
66
67     getMetadataAssociations ();
68     getRelationshipsInEvidence ();
69 }
70
71

```

Figure 6.8 AssocGEN code logic

The *existence relationships* R_e are determined to exist between the user files and their copies in the temporary Internet files folder, the *happens before* relationships R_h are determined between the browser logs obtained from the browser history and cache and the *download* relationships R_d are determined between the browser cache and the temporary files. The relationships determined from the metadata associations for the Internet Explorer browser are shown in Table 6.3. The results were found to be identical when I repeated this experiment with the Mozilla Firefox browser.

Number of distinct relationships	Internet Explorer
Existence relationships R_e	142
Happens relationships R_h	424
Source relationships R_s	3
Download relationships R_d	424

Table 6.3 The discovered metadata based relationships in the evidence

Since I was primarily interested in establishing the origin of the digital image files discovered on the user's file system, I only focused on that set of image files, 142 in number. These were the digital image files that were discovered in the temporary files folder of the user's computer. When I compared the browser logs (history and cache), I derived 424 relationships which aided in identifying 424 unique resources that were visited and downloaded. Each resource identified contained a *happens before* relationship R_h with a corresponding record in the browser history log. A similar relationship was also determined between the browser cache and the temporary files folder giving rise to 424 unique files being discovered in the temporary files folder. These included the 142 digital image files and other web resources such as validation scripts (.js) and bitmap images (.bmp). For the sake of this exercise, I only focused on identifying those *download* R_d and *happens before* R_h relationships identified between the user's computer and the web domain ascertained as the origin. Other activities including normal web browsing activities of the user were omitted.

The relationships also identified 3939 digital image files on the user's file system which were captured using three distinct digital still cameras, namely, an AgfaSensor 505, a FujiFilm_FinePixJ50, and a Pracktika_DCZ5.9 as determined from their EXIF metadata. These digital image files indicated 3 respective source relationships with the digital image files whose origin is my subject of discussion.

6.5.1.4 Analysis

Once the relationships are determined in evidence, the files' origins are determined by mapping the web page linked to the download of the resource leading to the identification of the files stored in the temporary Internet files folder and their presence in the user file system. Figure 6.9 shows the pairings of the image files discovered on the user's computer and their respective web page origins.

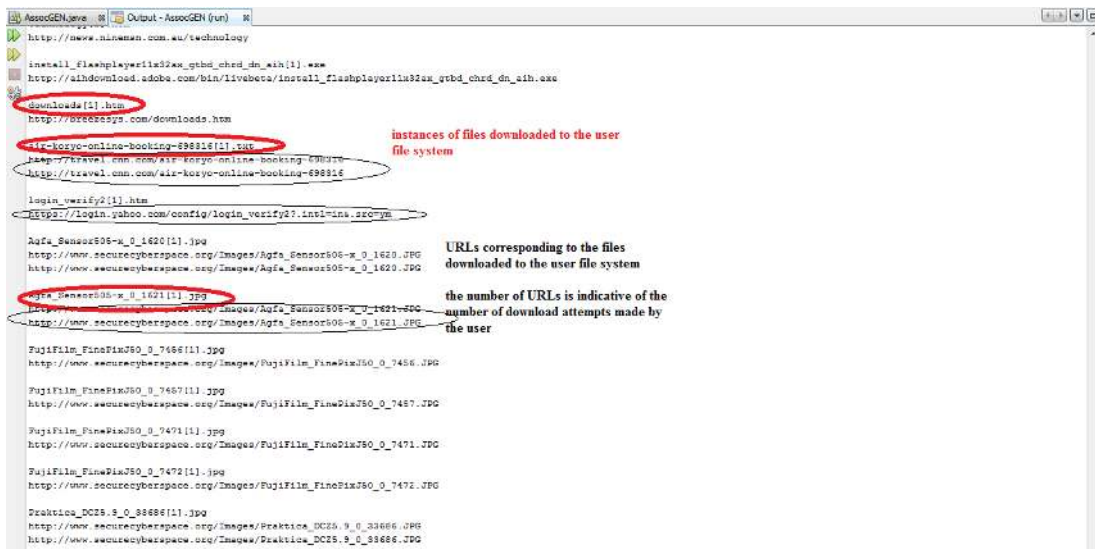


Figure 6.9 AssocGEN pairing of the image files with their respective web page origins

In each grouping that is shown in Figure 6.9, the image file name is printed first followed by the URL corresponding to the web page visited in the browser log. The groupings where multiple URLs are listed with a digital image, indicate multiple visit counts that represent the number of additional copies that were downloaded to the user file system. In all, there were 142 digital images that were downloaded from the specified web domain and also 282 other web resources such as validation scripts (.js) and bitmap files (.bmp) which were discovered on the user's temporary Internet files folder. Besides this, the metadata associations determined that the user file system also contained 3939 digital image files which were taken with 3 different digital still cameras (an AgfaSensor 505, a FujiFilm_FinePixJ50, and a Pracktika_DCZ5.9) and exhibited structural similarity relationships with the digital image files downloaded from the specified web domain. These findings suggest that these digital images were also likely to have been downloaded from the same web domain, although there is no current trace of this in the evidence other than the image relationships determined.

To corroborate the findings, I analyze the browser history logs (Figure 6.10) and determined the origin by tracking the URL in the attribute corresponding to the resource in question. In Figure 6.10, the presence of the image file on a website is identified which also provides us with a URL.

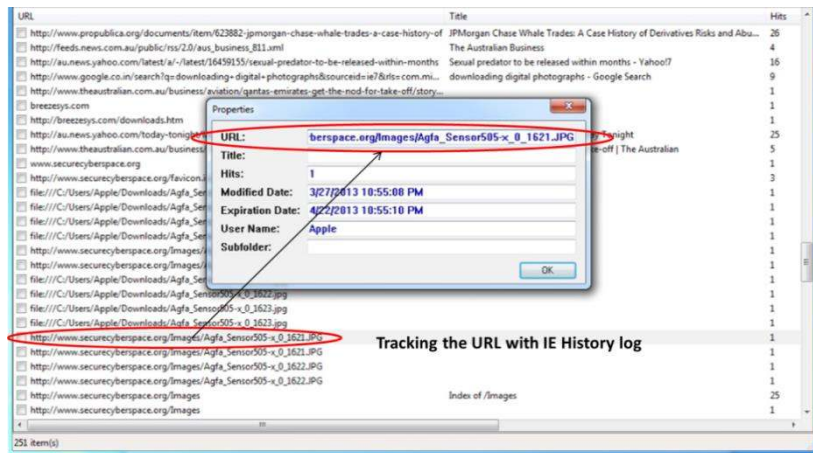


Figure 6.10 Analysis of Internet Explorer History - identifying the origin of download

To corroborate this finding, I visited the website (webpage snapshot illustrated in Figure 6.11) and determined that the image file is indeed listed. In addition, I also noted the presence of other files which are likely to be present on the user file system.

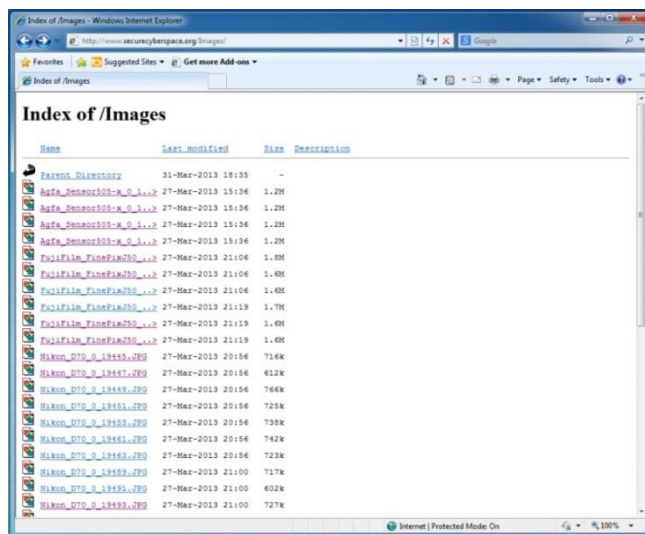


Figure 6.11 Snapshot of the specified webpage corroborating the listed files in the user's computer

When the findings were corroborated against the network packet trace, I obtained a similar assessment as illustrated in Figure 6.12. However, if I were to incorporate the network trace as another source of evidence into AssocGEN, then the analysis engine will simply group the respective TCP sessions between the domain of origin and the user's computer and incorporate it into the association groups corresponding to the appropriate relationships.

This suggests that these image files were also likely to have been downloaded from the website, although a recent search reports that these image files have now been taken down from the

website. To corroborate that the user had indeed downloaded these image files from the identified origin, I analyzed the network trace using Wireshark which is illustrated in the Figure 6.12.

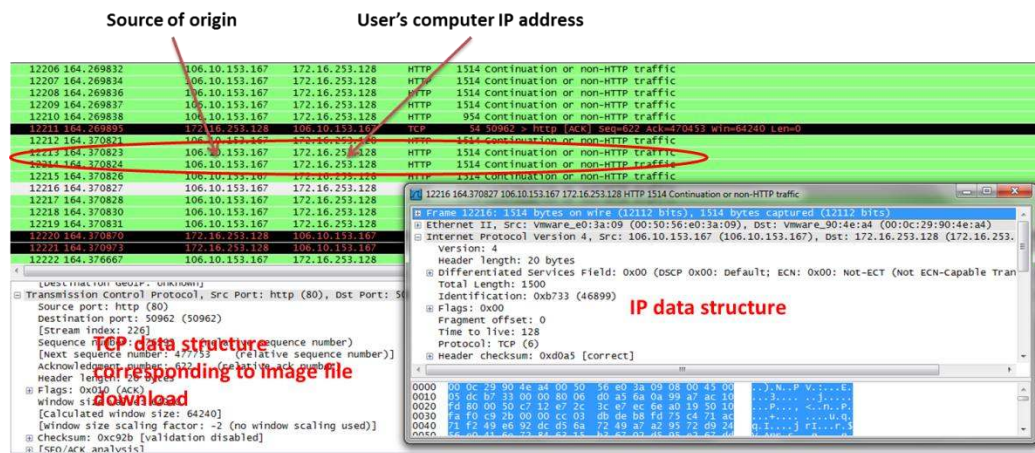


Figure 6.12 Corroborating the findings with network trace analysis

6.5.1.5 Conclusions

Via this case study, I have thus demonstrated the use of the Metadata Association Model to determine relationships between different sources of digital evidence, viz., a user's file system, browser logs and temporary Internet files to discover the origin of digital image files downloaded from the Internet.

6.5.2 Image Analysis

In this experiment, I develop a systematic method to grouping digital image files in a given collection for analysis. When we analyze collections of digital images, it is important to gain an understanding of how many digital photographs, digital generated composites, edited photographs and downloaded images exist in a collection. This is typically provided by standard classification techniques which identify the source based on a number of different known parameters. However, it is also important to determine those images that relate to certain images of interest which belong to a particular class. For instance, when we classify digital images according to their source, how do we determine the set of related images to a given digital photograph?

- Which are photographs taken with the same camera model and that were edited?
- Which are edited photographs from different camera models using the editing software discovered in step (a)?
- Which are photographs that were digitally generated using the editing software discovered in step (a)?

- (d) Which are photographs that are potentially downloaded images having identical image settings to the image of interest?

By their very nature, these and other such questions necessitate one to study the relationships that exist in the documents, a task that requires content analysis, usually by an individual. I demonstrate the application of the Metadata Association Model to determine such information from diverse collections of digital image files.

In regards to the nature of the analysis, it concerns the grouping of those metadata which belong to the families that elicit answers to the six questions identified by Casey [32] pertaining to forensic analysis. In this experiment, I studied the nature of associations that result in specific relationships as identified in Chapter 5 for varying values of association index ai across the datasets. The purpose of this experiment was to conduct a triage on collections of digital image files using the MAM eliciting common context across image files based on metadata based relationships. The effectiveness of the number of relationships discovered is measured using the grouping efficiency η .

6.5.2.1 Method

When analyzing a collection of digital images, classifying the source of the digital images is a common starting point. I classified each digital image collection based on source into four classes by grouping them using metadata which are only present under certain conditions. The source classification method is summarized in Table 6.4. The symbol ‘✓’ denotes the presence of the metadata and ‘×’ denotes the absence of metadata.

Classification Category	File System Metadata	EXIF metadata	JPEG metadata	Editing Software metadata
<i>Digital Photographs</i>	✓	✓	✓	×
<i>Software Processed</i>	✓	✓	✓	✓
<i>Computer Generated</i>	✓	×	✓	✓

<i>Incomplete Image Metadata</i>	✓	×	×	×
<i>Unclassified</i>	✓	×	✓	×

Table 6.4 Image Classification based on source

6.5.2.2 Expected Behavior

Digitally processed photographs for instance, are likely to generate many association groups since the images in this class will be found both under the category for camera make and model and the photo-editing software. The file system metadata which is present in each of the 4 categories are also likely to generate several metadata matches. Besides this, the MAC timestamps and EXIF timestamps can be used to generate a unified timeline of the digital images. The JPEG metadata that describe the image dimensions can be used to classify images based on the image resolution. Oftentimes, however, images which are rather small in size and dimensions could just be thumbnails and may be ignored for the purposes of analysis. The complete implications of all the associations generated between the different categories are discussed in the following subsection. The set of possible associations that can be identified among the various lists is illustrated in Figure 6.13.

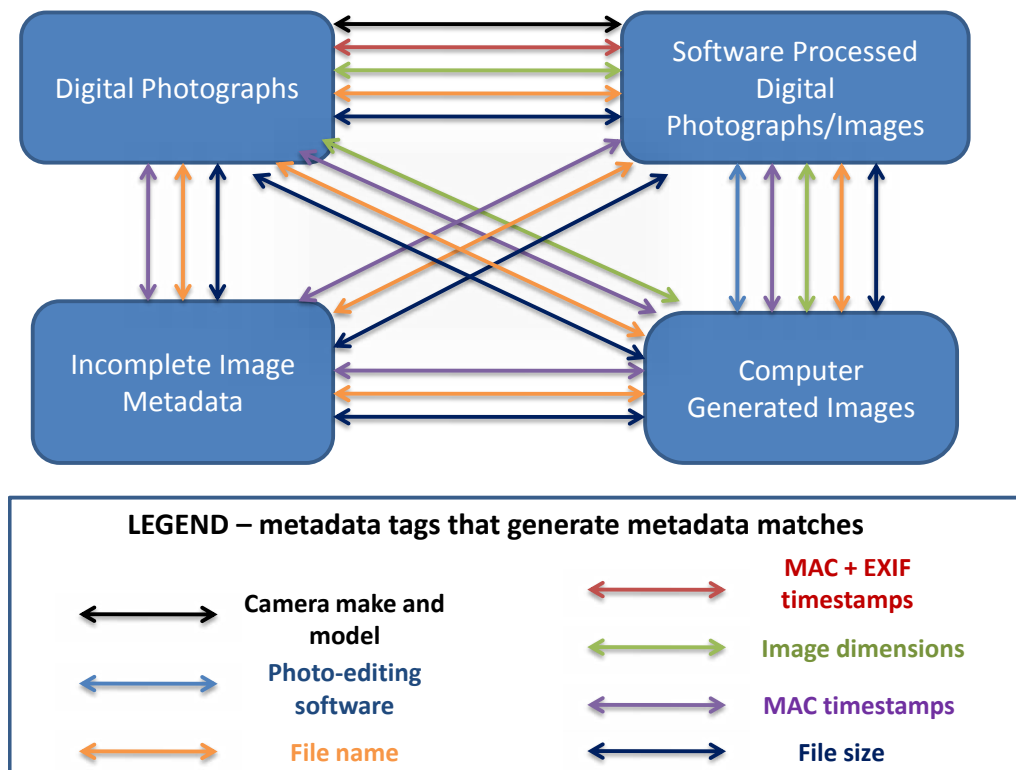


Figure 6.13 Possible metadata associations between the different lists

As Figure 6.13 shows, digital images from all collections can potentially generate metadata associations based on all file system metadata, i.e., Filename, File size and MAC timestamps. Where exact value matches were rare, a threshold margin of 20% was allowed. This was practiced on both numeric and string metadata values. On numeric metadata, the margin indicated a value $\pm 20\%$ of the reference value while on string metadata, the margin referred to difference in characters of up to 20% the size of the larger string.

Since digital photographs taken with the same camera use identical software and hence tend to store digital images with similar file names, the metadata associations and the groups subsequently generated can be used to determine if such is the case. Where this hypothesis is falsified, an examiner would be able to conduct analysis on that specific association group to determine the cause.

When digital images are associated based on image dimensions, it allows one to prune the set of thumbnail images which would have, under usual conditions, been downloaded by a browser when one visits a web page represented by a URI. In a collection such as the one identified in Dataset 4 which contains a significant number of images downloaded from the Internet, I believe that these associations and the groupings can aid a forensics examiner to focus on a smaller set of digital images, while excluding the thumbnail images from further analysis. However, if such a thumbnail was generated while processing an image using some photo-editing software, this will be determined by the source classification method as a computer generated image and the examiner can analyze that image as deemed necessary.

6.5.2.3 Observations

All the files that contained the EXIF metadata generated the *source* R_s relationship between them and between any two such image files f_1 and f_2 , the relation $f_1 R_s f_2$ held, which was applied associatively. Similar associations led to the identification of the other classes. Digital photographs that were edited and stored in the same collection gave rise to the *unauthenticated* R_{ua} relationship which was later confirmed after establishing the *existence* R_e and *majority* R_m relationships on image files from temporary files. The existence relationship was established between the digital photograph and the temporary file while the photograph and the temporary file exerted the majority relationship over the edited image from the collection.

We also identified the high-resolution and thumbnail images in the digital image datasets by overlaying the image dimension information from the *application* metadata family. The set of digital images labeled “unclassified” were then presented to the user for separate analysis. The results of applying the common source identification method to my digital image datasets are shown in Table 6.5.

Data-set No.	Dataset Volume	Number of images in the dataset	Digital Photographs	Images Edited with Software	Computer Generated Images	High-res images (> 1 MB)	Images with Incomplete Metadata	Thumbnail Images (< 10 KB)
1	374 MB	126	124	7	0	23	2	0
2	126 MB	52	34	0	0	3	0	0
3	1.6 GB	491	312	53	12	48	179	82
4	6.8 GB	2157	207	1891	0	0	1891 ⁴⁸	501
5	24.2 GB	8896	8896	0	0	7919	0	0

Table 6.5 Results of Common Source Identification for Image Datasets

The set of discovered metadata associations between the different lists among the digital images from the different datasets is shown in Figure 6.14. Figures 6.14 (a) through (e) illustrate the number of metadata matches that were determined across the different source categories.

We regard datasets that contained few metadata associations with the adjacent image classes as basic datasets (refer to Figure 6.14 (a), (b) and (e)). In such datasets, the image associations were predominantly within the images in the same source class. Dataset 1 contained 124 digital photographs of which 7 were processed with Adobe Photoshop and 2 images belonged to the category Incomplete Image metadata. Since the 7 Software Processed images was essentially a subset of the Digital Photographs, I was able to determine all possible metadata matches as shown in Figure 6.13 between these categories. Datasets 2 and 5 only contained digital photographs and hence there were no other categories to determine metadata matches with. However, I determined metadata name-value matches on all the metadata listed in the legend below with the exception of photo-editing software metadata tag amongst the images in the respective datasets.

⁴⁸ These images were determined to be downloaded from the Internet based on the information provided in the Digital Corpora repository regarding the source of these digital images.

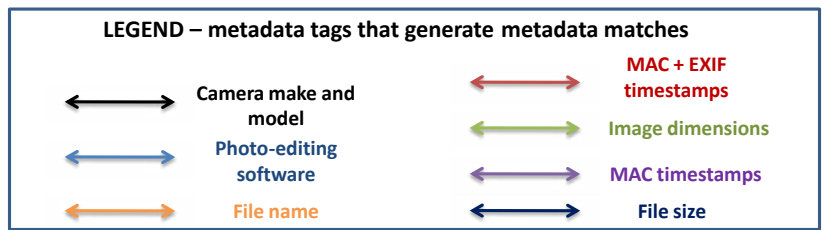
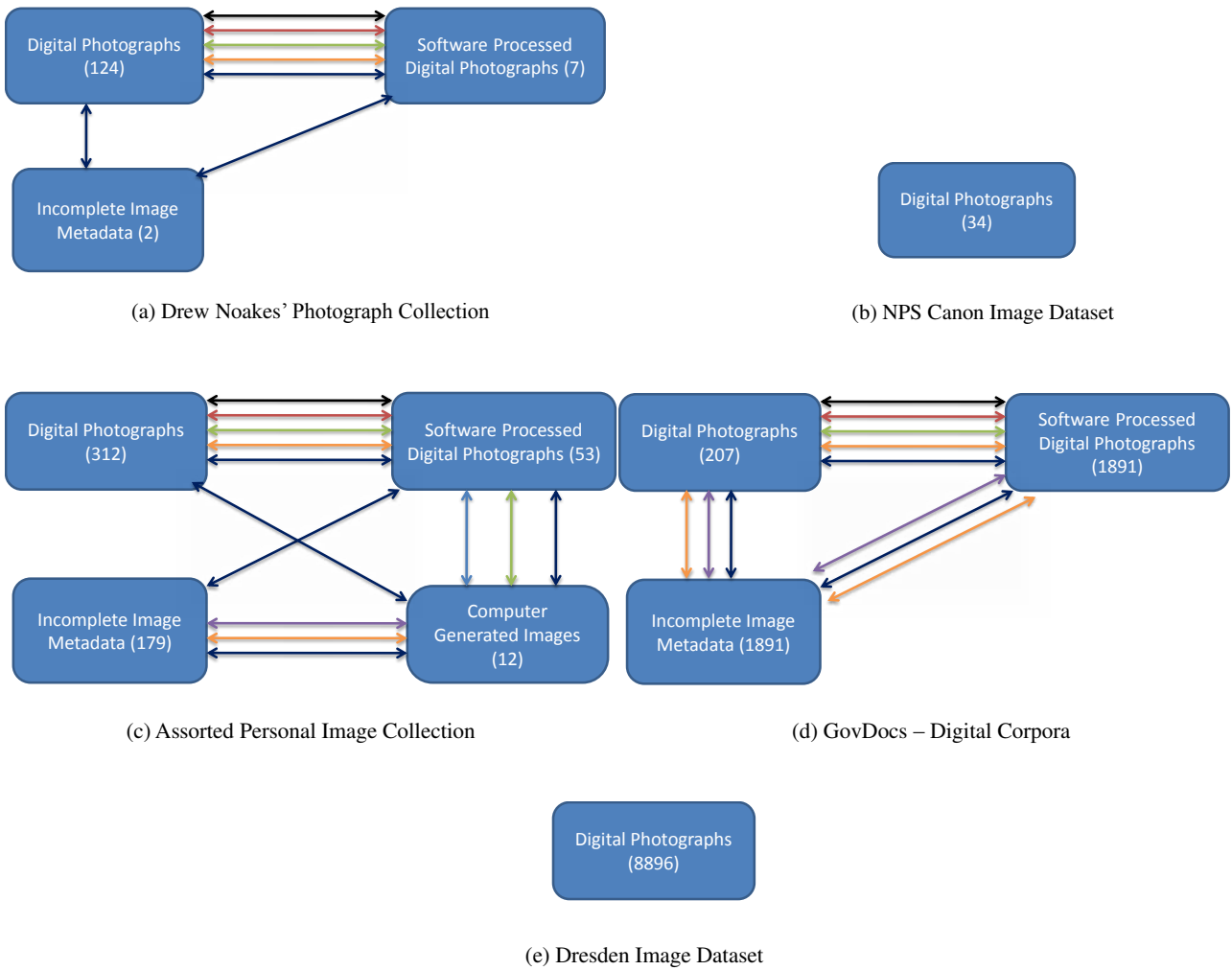


Figure 6.14 Metadata associations discovered among the digital images from across all the datasets

We regard datasets that contained significantly large number of metadata associations across the different image classes as assorted datasets (refer to Figure 6.14 (c) and (d)). In Dataset 3, I discovered overlapping sets with regard to the Digital Photographs and the Software Processed Images. The set of Software Processed Images also overlapped with the set of images that contained Incomplete Image Metadata, primarily on the ‘Software’ metadata tag. There were 179 images identified under the category of Incomplete Image Metadata, however, 53 of those contained the ‘Software’ metadata tag. Hence, between the two sets of categories Digital Photographs and Software Processed, and Incomplete Image Metadata and Computer Generated, I

discovered all possible metadata associations as defined by Figure 6.13. Based on the overlaps observed between the categories Software Processed and Computer Generated, the only photo-editing software found was Adobe Photoshop, albeit different versions. Some images in this collection were intended to be used as desktop background images and, therefore, were primarily found in standard image dimensions of 800×600 , 1080×800 and 1200×1080 . I observed that digital images that were similar in image dimensions were also similar in their file sizes.

Dataset 4 had three distinct categories, Digital Photographs, Software Processed Images and Images with Incomplete Image Metadata, and the missing category was Computer Generated Images. Although the application of the classification method discovered 1891 images as belonging to Incomplete Image Metadata, the Digital Corpora repository identified all these image files as being downloaded from the Internet. All digital photographs in this dataset were processed using Adobe Photoshop and the Digital Photograph category is hence a subset of the set of Software Processed Images. Moreover, I observed that merely using the digital camera make and model alone failed to classify the images correctly since many of the images in this dataset contained EXIF metadata with the exception of the camera make and model. I believe that since all these Digital Photographs were processed using Adobe Photoshop, the camera details were stripped during one of the many editing operations that may have taken place. Based on my correspondence Simson Garfinkel, this may have occurred before it was copied to the repository. Between these different categories that were detected in this image dataset, I discovered all possible metadata associations as suggested in Figure 6.13.

In datasets which contained both Digital Photographs as well as Software Edited Images, the number of associations discovered was the greatest since these digital images contained the most metadata that could generate interest during a forensic investigation. In contrast, the list of Images containing Incomplete Image Metadata were the most isolated group with the exception of Dataset 3 where the images downloaded from the Harry Potter website and those downloaded in response to Google search queries generated metadata associations among each other on 'File name', 'File size' and the 'last modified timestamp'⁴⁹. This is due to the fact that these images were indeed Computer Generated Image files that were downloaded from the Internet and consequently belonged to both categories.

⁴⁹ The other two MAC timestamps, namely, the creation timestamp and the last access timestamp, mimicked the value of the last modification timestamp, since this was accessed via the local file system into which the image collections were downloaded.

There were many associations discovered using the image dimensions but this can be attributed to the fact that images of a certain size tend to have a specific resolution which is relative common across many files. For instance, the image dimensions 800×600 and 1080×800 were commonly found in computer generated images which are also the standard desktop resolution ratios on common computer monitors. All Digital Photographs that were processed using software, for instance the images under both categories in Datasets 3 and 4, were found to have similar image dimensions. Many of these image files also had the same or similar file names and were hence pocketed together, based on the Filename metadata tag. A snapshot of the results displayed on Dataset 3 using AssocGEN is shown in Figure 6.15.

Where the digital image files contained insufficient metadata, I correlated the available metadata with the temporary files and Internet browser logs⁵⁰. In cases where I observed the existence R_e and the download R_d relationships and a source match using file name metadata, I labeled those image files as *potentially downloaded files*. This was later established as a fact when I determined the happened before R_h relationship between the browser logs. The potentially downloaded files were present in the temporary files and the filename metadata matched against the resource name in the browser cache logs.

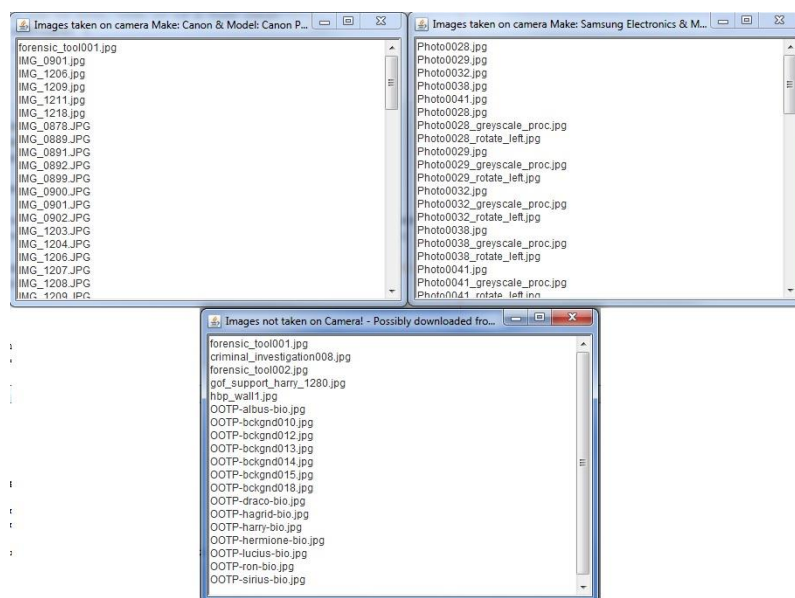


Figure 6.15 Snapshot of AssocGEN displaying the results of classifying digital image files based on source

By tracking the domain name identified in the browser log, I identified the URLs on the browser history log that corroborated with the timestamps against the cache log. The results of association grouping applied to all the digital image datasets is shown in Table 6.6. My findings are discussed

⁵⁰ It is presumed that the activities were tracked and logged to generate the necessary sources for analysis.

below. In the effort margin reported in column 7, I have listed each unassociated digital image as a single association group for computing the r and η values. My observations are analyzed in the sequel.

Data-set No.	Dataset Volume	Number of images in the dataset	Association index (ai)	Association groups, Unassociated images	Effort margin (w/o unassociated images)	Effort Margin (inclusive of unassociated images)	Grouping efficiency η
1	374 MB	126	0.42	4, 2	0.032	0.048	0.952
2	126 MB	52	0.21	1, 18	0.029	0.366	0.634
3	1.6 GB	491	0.06	7, 179	0.022	0.379	0.621
4	6.8 GB	2157	0.001	10, 1891	0.037	0.888	0.112
5	24.2 GB	8896	0.86	36, 0	0.004	0.004	0.996

Table 6.6 Results of association grouping to Image Datasets

6.5.2.4 Analysis

In Dataset 1, I discovered 4 different camera makes and models that were used to take the Digital Photographs, of which 7 were edited using Adobe Photoshop 6. Two images with Incomplete Image Metadata remained unassociated. With regard to the grouping efficiency η , it is very close to 1 given the number of images that were grouped based on metadata associations, $\eta = 1 - \frac{(4+2)}{126} = 0.952$. All valid digital images in Dataset 2 belonged to Digital Photograph and captured with a single camera. With regard to the efficiency η , it is low owing to the large number of unassociated files. Since 18 of the 52 images did not have sufficient metadata, those files remained unassociated bringing down the overall value for grouping efficiency. However, all the 34 digital photographs were grouped together and thus $\eta = 1 - \frac{(1+18)}{52} = 0.634$. In Dataset 3, the original set of 62 photographs, in addition to the 250 digital images generated using various photo-editors, resulted in 312 digital photographs. Of these, only 35 photographs were discovered with the ‘Software’ metadata tag from Adobe Photoshop. Among the images in this collection with Incomplete Image Metadata, 12 were identified as Computer Generated Images as they contained

the ‘Software’ metadata tag in addition to file system metadata. With regard to η , there were 6 association groups resulting from the digital photographs and one separate association group for all the Computer Generated Images. Additionally, there were 179 images without sufficient metadata and this impacted the overall value for grouping efficiency. Therefore, $\eta = 1 - \frac{(7+179)}{491} = 0.621$. In Dataset 4, I observed the presence of metadata tags for camera make and model only in 207 digital images among the entire collection. The ‘Software’ tag on the other hand was detected in 1891 digital images. Other types of EXIF metadata however were present in all these 1891 digital images. Hence, these digital images are categorized as Software Processed Images and not Computer Generated⁵¹ Images. With regard to η , there were 10 association groups generated from the set of 207 Digital Photographs and 1891 unassociated images. Naturally, this Dataset had the lowest value for grouping efficiency. Therefore, $\eta = 1 - \frac{(10+1891)}{2157} = 0.112$. All the digital images in Dataset 5 were Digital Photograph images and none of them contained the ‘Software’ metadata tag. Since there were no unassociated files, all the digital photographs were grouped into 36 association groups. Each association group obtained in this case corresponded to the distinct cameras used in generating this collection. Therefore, this set has the largest efficiency with $\eta = 1 - \frac{(36+0)}{8896} = 0.996$.

In order to compare the values for the effort margin r against the reduction factor proposed in theory, it must be noted that reduction factor only applies to the groups sans the unassociated artifacts. Therefore, I compute the effort margin values discounting the unassociated digital images from the Datasets and have listed the values in Table 6.4 for column labeled *effort margin (w/o unassociated images)*. I observe that these effort margins are a fraction of the effort required to analyze the individual image files for Dataset 5; this is due to the fact that there were no unassociated digital images and all the image files formed distinct groups of related digital photographs. In the presence of metadata associations leading to grouping of image files, the effort involved in analyzing the images reduces to a fraction of the total effort needed to analyze them individually.

The digital image collections that contained digital photographs typically contained multiple photographs from the same digital camera. All digital photographs from the same camera generate

⁵¹ The description of these images on the Digital Corpora website shows that none of the images were specifically generated using a computer, which was taken into account in this classification.

source relationships between each other and consequently are grouped together in the same association group. Naturally, each digital image in that group finds all other digital images from the same group. As a result, if one of the digital images in an association group had an *ai* value of 0.3, all the other digital images in that group also had the same value. In general, we may state that each digital image had an *ai* value which is the fraction of the total number of digital images in the Dataset that were associated with that image. Therefore, Datasets that contained digital photographs (both normal and edited) produced higher values for *ai* as against datasets that contained fewer digital photographs. In Datasets 1 and 5, I found high averages for the *ai* values since there were little or no unassociated digital images. In Dataset 2, while most digital images were digital photographs and associated based on source and ownership relationships, a third of the Dataset was unassociated which resulted in a lower average value for *ai*. In Datasets 3 and 4, I observed very low values for the average *ai* since a significant number of digital images in these Datasets were unassociated.

6.5.2.5 Ground Truth

The macro-level classification was determined to be accurate on Datasets 2 and 5 where the digital image files were primarily digital photographs and were not edited. However, Datasets 3 and 4 involved a significant number of digital image files downloaded from the Internet, and in the absence of sufficient metadata or alternate sources to corroborate the evidence like temporary Internet files or browser logs associated with the image file downloads, the Existence R_e , the Download R_d and the Happened Before R_h relationships cannot be established. As a result, the precise nature of the different operations performed on these digital images could not be established. The unassociated digital images were removed to an unclassified list. Such files were individually analyzed by examining the corresponding forensic images under FTK.

6.5.2.6 Conclusions

Through the case study in this section I have thus demonstrated the use of the Metadata Association Model to determine which files are related to a particular matter of interest by using standard image classification groups and identifying cross linkages using metadata based value matches.

6.6 Discussion

During forensic investigations, investigations often require information on the circumstances and conditions prevalent during periods of interest. The semantics associated with metadata usually relate to events (e.g., timestamps) and consequently, determining matching metadata values correspond to identifying identical or related events. Metadata underlines the context to describe the *situational similarity* during the life cycle of the digital images stored in digital evidence. Using metadata associations, we can *automatically identify and group*:

1. a digital photograph and any altered version of itself together;
2. an edited image with digital generated images using a particular software;
3. digital image files with log records that identify the event sequence tracing the file download from the Internet;
4. a digital photograph or a digital generated image with image files that are related or similar containing partial metadata; and
5. all thumbnail image files.

The ability to automatically identify and group such related sets of digital image files based on metadata associations simplifies the process of analysis for an examiner. Metadata associations can be used to validate hypotheses by comparing different metadata values across the digital images from a known source and establish consistency among them.

We now present a brief discussion on the parameters used in my study and the use of digital image relationships based on metadata associations to conduct analysis.

6.6.1 Association Index ai vs. Grouping Efficiency η

The association index ai assesses a collection of digital artifacts holistically and is an average measure of the number of associated digital images that can be discovered by pivoting on a single digital image in a collection. The grouping efficiency η quantifies the number of association groups and it quantifies the benefit perceived in the analysis after applying the MAM in comparison with traditional file-based methods. Although efficiency η is influenced by the association index ai , its values are generally larger than ai values in my datasets. This is due to the relatively large number of unassociated digital artifacts that effectively result in single member

association groups (treated so, for the purpose of analysis). Each member association group has a small association index and when averaged over a large collection, it can bring down the overall value of ai for that collection. On the other hand, η is a ratio of the number of association groups to its worst case scenario. In cases where there are single member association groups, its contribution is moderated over the collection.

6.6.2 Digital Image relationships and analysis

When it is suspected that one or more digital image files were downloaded, this can be established by identifying the download R_d and the happens before R_h relationships between the image files and the respective browser log files. Digital image files that demonstrate an existence R_e relationship indicate the presence of another copy of that image and this can be used to determine duplicate image files in a collection. Besides this, when such pairs of image files also exhibit a source R_s relationship along with unmodified authentication R_{ua} relationship, an edited image file is likely to be present whose original image is identified using the existence R_e relationship. Naturally, during image analysis, these image files can be starting points when no other information is available regarding the image collection. Each camera make and model identified through a source R_s relationship is a potential source of digital evidence discovered. Digital image files that demonstrate a parallel occurrence R_{po} relationship are likely to have been operated on using some software if there is an exact metadata match and further analysis of the content may be warranted in cases where a R_{ua} relationship is not observed. Digital image files which exhibit the structural similarity R_{ss} relationship are likely to possess identical image resolution capability and encoding indicating that their content can be analyzed using the same tool. This can be useful if an unknown application format is detected during the examination of the image collection. Images with incomplete image metadata, unless they contained illicit content, can be spared from unnecessary analysis. However, that may be ascertained only through content processing using an alternate tool.

6.7 Chapter Summary

In this chapter, I studied the use of the Metadata Association Model to analyze collections of digital image files. Depending on the nature of the forensic analysis, I demonstrated two methods, viz., *need based* and *exhaustive* to determine metadata associations and group the related digital images. I discussed the formation of association groups across multiple source classes by determining metadata matches between them. I illustrated the use of digital image file

relationships to determine instances of image downloads and identify the origin of these downloads.

While image processing attempts to capture the content-related information about a digital image, the metadata, on the other hand, records and transports the situational information of the digital image. Using metadata belonging to the four metadata families, I have shown that it is possible to determine digital image relationships through metadata associations to find answers to questions pertaining to the analysis of digital image collections.

In the following chapter, I demonstrate the use of the Metadata Association Model to analyze word processing documents.

*“I read, I forget;
I see, I remember;
I do, I understand.”*
- Confucius

7. MAM Based Analysis of Word Processing Documents

In this chapter, I focus on the application of my Metadata Association Model to collections of word processing documents to elicit metadata based associations and scope a forensic analysis. I assume no prior knowledge of the word processing document collections in my experiments. This chapter demonstrates the functional completeness of the model in two modes of operation: determining *need-based* and *exhaustive* document associations. When analyzing word processing documents, it becomes necessary to determine some important parameters:

1. The authors, and authors' affiliations;
2. File names and patterns;
3. The range of file sizes;
4. The applications used and their frequency of use;
5. The authors who created the most number of documents; and so on.

In my research, I apply my method to group the metadata matches into groups and determine these characteristics.

7.1 Conducting Forensic Analysis on Collections of Word Processing Documents

When collections of digital image files are analyzed as part of an investigation, many forensic questions can be raised during the analysis, some of which are as follows.

1. How many document authors can be identified from the word processing document metadata? How many documents belong to each of these authors?
2. How many document files were downloaded? If any, can the source of these files be determined?
3. How many downloaded documents are edited? What was the editing software used?
4. Are there other “similar” document files without ownership/authorship metadata?
5. Which of the document files were downloaded via Emails? If any, can the parties of the email be identified? What were the mail carriers? Is there a mail client with a copy of the relevant emails?

To determine answers to such questions, it is necessary to recognize that no single classification method can provide all the answers and it is necessary to determine relationships between the word processing documents to extract all higher-order associations that exist both within a particular source class and across such classes. Such a task requires exhaustive classification using all individual parameters (from metadata) as well as all combinations of multiple parameters to determine where the document files overlap and group them.

Word processing documents store metadata relating to the author and owner of a document which I use to determine the names of individuals and the names of computer software that were used to create/modify the document. Documents also have metadata which record characteristics pertaining to the structure of a document, word count, page count and so on. These metadata are useful in determining how documents were created, and stored on a file system. When this metadata is used in conjunction with the author/owner metadata, it can determine all users who created similar documents.

7.2 MAM Evaluation Using Word Processing Document Collections

While traditional approaches in analysis have presumed the existence of knowledge regarding the digital artifacts in the sources of digital evidence, the Metadata Association Model does not assume the existence of prior knowledge. I demonstrate the two modes of evaluation in my experiments discussed in Section 7.4. In general, when applying the MAM to a collection of word processing documents for analysis, there are two modes of operation, viz., need-based and exhaustive, as described in the previous chapter.

7.2.1 Criteria for Selecting Word Processing Document Collections

The association index ai (defined in Section 5.3) is a measure for determining the quality of metadata associations that can be derived out of a given dataset. The index provides an estimate of how “connected” a dataset is and is given by the mean of the association index computed for all the documents in that dataset. A document that is likely to generate a large number of metadata associations is likely to be highly connected while a document that contains few metadata may give rise to only a few metadata associations and thereby be less connected. A document which does not generate any metadata associations is therefore “unconnected”. As previously mentioned in Chapter 5, the higher the value of association index ai for a source, higher is the value of the grouping efficiency η .

In order to determine answers to the questions we’ve posed in Section 7.1, the word processing documents in a dataset are required to have certain properties in regard to their metadata. These properties are as follows:

1. At least one metadata from each metadata family should be available:
 - a. metadata identifying one or more software applications;
 - b. metadata pertaining to the format and structure of the word processing document; or
 - c. metadata pertaining to time instants when specific events affecting the word processing documents occurred.
2. Digital artifacts referring to the same instance of a word processing document must demonstrate *existence* and *source* relationships.
3. Word processing documents with identical or similar filenames being stored in different file formats must demonstrate *existence* and *source* relationships.

4. References to word processing documents across log files must demonstrate a *happens before* relationship.
5. All available file system and application metadata must be authentic.

Corroboration of related word processing documents has been an integral part of forensic analysis [26, 28, 50]. In order to identify word processing documents and corroborate them, the dataset must contain word processing documents with identical source metadata information (threshold association may also hold). When identical copies of word processing documents are stored in different formats (as regular or backup or temporary files) are discovered, they must demonstrate the existence R_e and source R_s relationships (metadata such as filename, file location, computer software). During analysis, it is customary to determine the sequence (timelines) of events involving the word processing documents. To achieve this, a word processing document must support at least one timestamp metadata.

7.2.2 Metadata & Metadata Families in Word Processing Documents

We identify the digital image metadata at their respective metadata families relevant during forensic analysis in Figure 7.1. A collection of word processing documents can be organized according to the image file names and their respective locations on a particular source of digital evidence. As discussed earlier, title or subject metadata can often throw light on understanding whether or not the document has been used as a template in creating the material while leaving the metadata untouched. ‘Creator’ and ‘Publisher’ metadata help identify some of the additional software used in generating the content. Such metadata belong to the *source* metadata family.

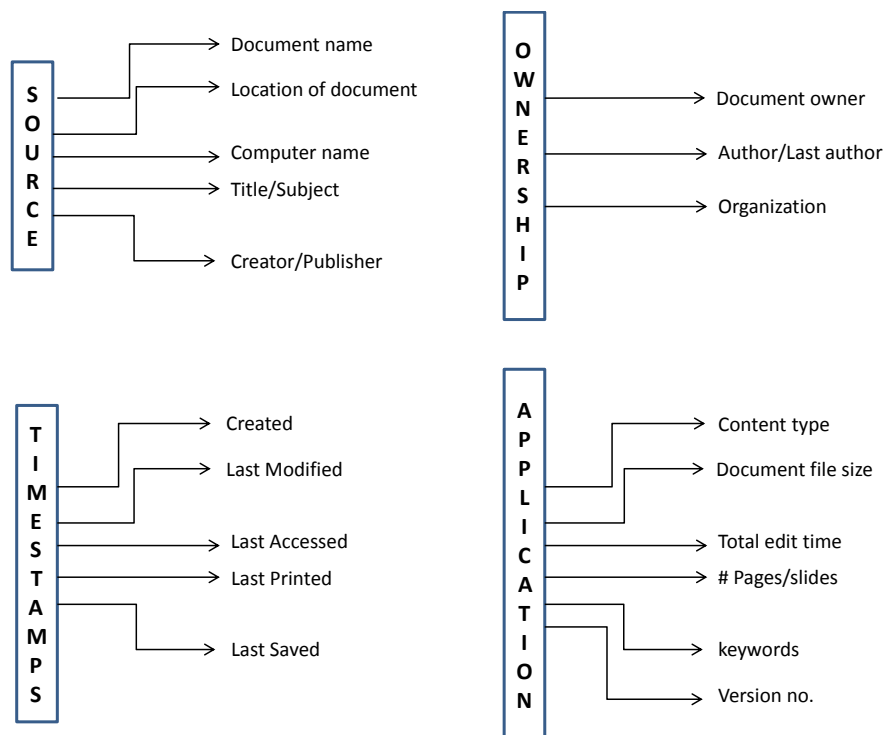


Figure 7.1 Word processing document metadata tags of interest in Digital Investigations

When dealing with documents, it may be necessary to identify the author(s), their affiliations with an organization or company, when and who last modified the document and so on. The metadata that allow one to do that belong to the *ownership* metadata family.

The MAC timestamps and the document timestamps, where available, belong to the *timestamp* metadata family and identify events corresponding to creation, modification and access of the word processing documents.

Metadata such as the number of pages, slides, etc., retain some content context. ‘Keywords’ is another metadata which, if available, could provide alternate keywords to examiners while exploring related documents or other digital artifacts from one or more sources of digital evidence. Such metadata that provide information regarding the features of word processing documents belong to the *application* metadata family.

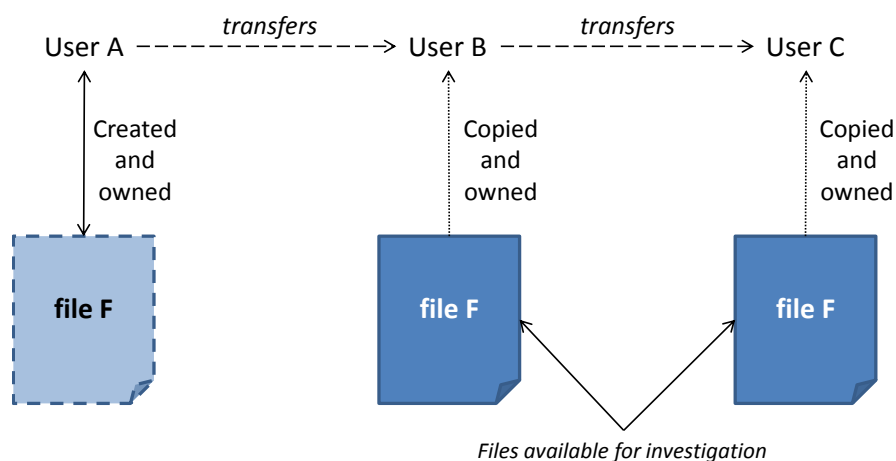
In the sequel, I describe the use of metadata associations to automatically corroborate a fact, in this case, to solve the classical file ownership problem introduced by Buchholz and Spafford [22].

7.3 Ascribing File Ownership Using Association Groups

Buchholz and Spafford noted that when sufficient metadata is recorded, it can aid in determining answers to the six questions listed by Casey [32] concerning forensic analysis. The metadata associations model takes this approach one step further by determining metadata based associations across digital artifacts to discover relationships and provide answers to investigation related questions.

7.3.1 File Ownership Problem

Consider the problem described by Buchholz and Spafford [22] where the owner of a document and the involvement of individuals is under investigation. I have adapted this problem to illustrate the benefits of using my framework to examine multiple sources of digital evidence and the model to identify metadata associations for solving this problem. There are three users, User A, User B and User C. User A creates a file F. User A then communicates with User B and transfers a copy of file F and User B in turn transfers a copy to User C. The ownership of the document is transferred to User B and then User C once each of them received a copy of the file F. The question for an examiner posed by Buchholz and Spafford [22] is who is responsible for file F? I interpret this question as who is the author of the contents found in file F? I illustrate this file ownership problem in Figure 7.2. From the description of the problem, I inferred that initially an examiner has access to some form of digital evidence, one source each from User B and User C, without loss of generality.



Who is responsible for file F?

Figure 7.2 The file ownership problem

7.3.2 Discovering User A

Since file system metadata only records the last known owner, the involvement of User A in the generation of the document cannot be traced from a simple examination using any forensic tool. Buchholz and Spafford [22] have observed that in the presence of *only* file system metadata, and particularly, the owner of the document, *this problem cannot be solved*. Moreover, they note that in the presence of a finite sized field to hold user information, the question cannot be answered, in general. In my work, I relax the assumptions slightly to include the use of document metadata in addition to file system metadata for identifying the original owner. Document metadata which are inherent to the document they are associated with are persistent across file copies over networks and this can be utilized in determining the provenance of the file.

From a digital forensics standpoint, Buchholz and Spafford advocate that if a forensics examiner is uncertain as to who (or which process) is responsible for an artifact, especially when multiple candidates exist (Users A, B and C are listed as the owners in their own copy of file F), one has to simply assume that all of them are responsible and retain information supporting that hypothesis. If the above scenario was investigated using conventional forensic toolkits, then some form of a digital evidence source would have been seized from User B and User C. Despite the potential for forensic tools to examine the two sources together, merely using file system metadata will identify both User B and User C simultaneously as the owners of file F, which is a fallacy. User A is never identified during the examination. Therefore, it is likely that the original author cannot be traced or could be wrongly identified.

However, using my approach, the sources acquired from User B and User C can be examined together using AssocGEN thereby allowing the examiner to corroborate the sources of digital evidence, in a tool-supported manner. This would then allow the parsing of not only file system metadata, but also the document metadata from the two copies of file F. The document metadata from the two copies of file F, from User B and User C, generates metadata matches and forms association groups that include the metadata 'Author' and 'Filesize'; the metadata value based on which the group was generated identifies *User A*, who remained undiscovered previously. Then, if a copy of the file F is also acquired from User A for analysis using AssocGEN, the metadata matches generated between this file and the two copies from Users B and C in addition to a timeline of the event timestamps from the three copies of the file F shows that User A is the actual owner.

7.3.3 Automatic Corroboration of Evidence Using Association Groups

With AssocGEN, a user can configure the tool to determine all metadata associations. The tool traverses the two sources and identifies the different digital artifacts and parses the respective metadata. After parsing the metadata, it identifies the metadata matches and groups them into similarity pockets (refer to Section 4.5). The similarity pockets containing overlapping digital artifacts are grouped and presented to the user. In short, these are the set of steps that take place:

1. Mount the two sources of digital evidence.
2. Traverse the sources and parse metadata from all digital artifacts.
3. Identify all metadata name-value pair matches and combine the respective files to form similarity pockets.
4. Merge overlapping similarity pockets into association groups.
5. Present the groupings to the forensic examiner.

Once the groupings are presented, the examiner can skim the groupings and find metadata matches that relate to the provenance of file F. In this context, the metadata tags ‘Author’, the MAC and document timestamps and the ‘Filesize’ are relevant. When the examiner studies the groupings, it will be found that the two copies of file F, one each from Users B and C are grouped together based on the metadata tag ‘Author’. Interestingly, the value contained is neither User B nor User C, but ‘User A’ although the owners of these files are listed respectively as User B and User C. Since internal metadata persist when documents are copied over networks, the ‘Author’ metadata generates a match between the two files identifying User A.

7.3.4 Conclusions

Application metadata in tandem with file system metadata will also generate multiple matches that correspond to information that relate to who, when, where and how, leading to association groups that characterize the similarity of the two copies of file F. Using AssocGEN, I was able to determine inherent Source and Application metadata relationships between these files that shows that neither User B nor User C was the original owner of the file. For the purpose of completeness, the provenance of the file can be established if a copy of the file is obtained from User A for comparison against the two copies of the file from Users B and C. Besides, the

timestamps recorded on the application metadata, which will predate the MAC timestamps⁵² discovered on copies of file F with B and C, will establish User A as the original author of file F. Hence, the use of metadata associations can aid in attributing the ownership of the file to the correct individual.

7.4 Datasets

To evaluate the MAM, it was necessary to identify datasets which span the spectrum of highly connected to weakly connected artifacts. Typically, word processing documents that were created by users from the same organization may generate multiple metadata matches including those that pertain to ownership, timestamps and document structure and template. Consequently, such collections are likely to result in highly connected documents. On the other hand, collections where the word processing documents were downloaded from different sources, e.g., downloaded from different websites while browsing can produce very narrow groups of image files and hence are likely to result in weakly connected or unconnected documents.

7.4.1 Word Processing Document Datasets

We describe two document datasets acquired from two different sources to conduct my experiments and generate association groups. In my datasets, the documents are largely created using the Microsoft Office application and therefore always generate metadata matches with regard to the application metadata. However, for the purposes of my research, I have limited my definition of a homogeneous source with regard to the MIME type associated with the document files and omitted the metadata match arising from the application metadata since this can render the association group formed trivial.

7.4.1.1 Desktop Dataset

The desktop Dataset was a collection of Microsoft Office documents from a personal computer containing 976 Microsoft documents. The computer was provided by a student volunteer that contained 49703 files in total. The period of data accumulation was between 2008 and 2011. The files contained in the collection were related to the student's research work over this period. This collection has 752 Word documents, 154 PowerPoint files and 70 Excel spreadsheets. More than half of the remaining files were system and application files, 4,533 were Adobe PDF files, 10715

⁵² The document timestamps were created according to User A's copy which will persist despite the copy. Since the MAC timestamps on the file for Users B and C are created after the file was copied from User A, the document timestamps will predate the MAC timestamps.

were digital image files, a few HTML files, logs files and several programming script files. The documents do not belong to any particular investigation. Primarily, most of the documents in this collection belonged to the volunteer and created by the volunteer, albeit with different user names and with multiple organization affiliations and while a few PowerPoint files were downloaded from the Internet. The collection contains multiple author names and 7 distinct organizations overall. The minimum file size is 10 KB and the maximum file size is 11.62 MB in this collection. Both Word documents and the PowerPoint files in this collection have template (.POT) files which were downloaded from the Internet. Word documents which are modified and updated as different versions share the same title in the metadata field. This dataset is summarized in the first row of Table 7.1.

Amongst the application metadata, the document's author was the most commonly occurring metadata tag in this dataset, in 822 out of the 976 documents. The documents recorded this field differently and I obtained the count from both 'Author' and the 'Last-Author' metadata⁵³ tags. When a document is created, the title of the document is recorded by Microsoft Office (Windows version) in the metadata tag 'Title' or 'Subject' which was the next most commonly occurring metadata tag. During document analysis, comparing the filename and this particular metadata tag could inform one if the document was created on its own or was derived from another document. In the latter case, the title could differ from the filename which would indicate that it is derived from another document. The organization the author is affiliated with is denoted by metadata tag 'Company' which was found in a little fewer than half of the documents in this dataset. 'Application-Name' which records the specific version of the application that created that file and the 'Last-Modified-Date' or 'Last-Printed-Date' was found in a little fewer than 600 documents⁵⁴. I was always able to extract the MAC timestamps and other file system metadata (such as file name, file path, and file size) for all the documents in the dataset.

7.4.1.2 Govdocs1: Digital Corpora Collection

The second Dataset was obtained from the Digital Corpora [68, 70, 71] repository containing several Microsoft Word documents, PowerPoint files and Excel spreadsheets. The document collection was downloaded from the corpora which contained documents and other files

⁵³ Although the values for metadata tag 'Author' need not coincide those of 'Last-Author', in my dataset, these values coincided where both were available. Hence, I deemed the tags to be equivalent in my experiments. Among the set of documents where both tags were present, I arbitrarily chose to count from metadata tag 'Author' and omitted 'Last-Author'.

⁵⁴ There were several documents in which the metadata tag was present but the corresponding value was NULL. Such documents were not included in the counting.

downloaded from the Internet by Simson Garfinkel for digital forensic research purposes. I downloaded folders 000 to 010 and subset0 to subset5 totaling 17399 files. The documents from this source contained a large number of metadata and adequate metadata per file to be included in my experiments. Within the documents in my collection, there were 2970 documents of which there were 1282 Word documents, 1044 PowerPoint files and 644 Excel spreadsheets. The remaining files were Adobe PDF files, HTML files, text and digital image files. The documents do not belong to any particular investigation and have been acquired from several different authors. These files were downloaded by Garfinkel and loaded in the Digital Corpora repository directly without any additional modifications to files, excepting the change in filenames [71]. These documents were downloaded from this repository and used in as-is condition in my experiments. The collection contains 103 different authors and over 80 distinct organizations overall. The minimum file size is 6 KB and the maximum file size is 58.6 MB in this collection. All files in this repository are numbered sequentially from 000 followed by the subfolder ID further followed by a 3-digit file ID between 000 and 999. The filenames themselves have been assigned arbitrarily. The filenames hence do not have any connection with the content of the file. Since all the files in this dataset were downloaded from the Internet and renamed, the file system metadata for the documents in this collection were taken from my local file system into which these files were downloaded. The file system metadata values are accurate from the point of the download, which are used in subsequent metadata based associations. The application metadata that were present at the time of download are used to report the values. This dataset is summarized in the second row of Table 7.1.

Amongst the application metadata, the document's author, taken both from 'Author' and 'Last-Author' metadata tags, was the most commonly occurring metadata tag in this dataset, appearing in 2921 out of the 2970 documents. The metadata tag 'Company' was found in 2702 documents in this dataset. The metadata tag 'Title' was discovered in 2406 documents. 'Application-Name' was found in 2760 documents and the 'Last-Modified' or 'Last-Printed-Date' was found in 1723 documents. I was always able to extract the MAC timestamps and other file system metadata (such as file name, file path, and file size) for all the documents in the dataset.

			Microsoft Document Metadata						File System Metadata	
Serial No.	Dataset	Dataset Volume	Author /Last author	Company /Organi- zation	Appli- cation name	Key- wo- rds	Last modi- fied/ printed date	Title /sub- ject	MAC Time- stamps	File size
1	Desktop (956)	1.6 GB	822	440	577	334	583	660	976	976
2	Digital corpora (2970)	6.8 GB	2921	2702	2760	1362	1723	2406	2970	2970

Table 7.1 Summarizing the Microsoft document metadata from the different datasets

7.4.2 Metadata Availability in Document Datasets

To gain a comprehensive understanding of the metadata distribution on documents within a standard file system, I calculated the frequency of occurrence of metadata from multiple collections of Microsoft Office documents. Based on my study of 10 workstations, a standard workstation was found to contain documents of different file formats (e.g., .DOC, .PPT, .XLS, .DOCX, .PPTX, .PPS, PDF, .TXT, .LOG, etc.) and from across different versions recording metadata to varying degrees of detail. Such a preliminary study was required in my research to understand the metadata distribution so that we may utilize the most frequently occurring metadata to determine associations and derive inferences during a forensic investigation.

7.4.2.1 Documents & Metadata Distribution

In the desktop machine, all the files across the entire file system were counted and reported. In the Digital Corpora dataset, the number of files downloaded from the main repository were counted and reported. Each metadata tag name was counted exactly once. The documents from each of the three formats and the respective cumulative number of metadata discovered on each type are shown in Table 7.2.

Source Machine	Name of files on Source	MS-WORD, No. Of distinct metadata	MS-PPT, No. Of distinct metadata	MS-XLS, No. Of distinct metadata
Desktop machine	49703	752 Word docs, 38 metadata	154 presentation docs, 38 metadata	70 spreadsheet docs, 17 metadata
Digital Corpora	17399	1282 Word docs, 59 metadata	1044 presentation docs, 36 metadata	644 spreadsheet docs, 17 metadata

Table 7.2 Preliminary statistics of relative metadata richness of different Microsoft Office document types

In Table 7.2 there were over 30 metadata tags in MS-WORD and MS-PPT files in comparison to fewer than 20 for MS-XLS files. Among these MS-WORD files (DOC extension), Microsoft Office 2007 Word files (DOCX extension) are the most metadata rich image files and similarly for MS-PPT, Microsoft Office 2007 PowerPoint (PPTX extension) documents. An interesting finding from my analysis of metadata in Microsoft Office document collections was that the same metadata could be referred to by different tags depending on the application version. For instance, while Word documents belonging to Word 2003 or earlier used the metadata tag ‘Last-Save-Date’, Word 2007 documents used the metadata tag ‘Last-Modified’ to refer to the timestamp when the document was last modified. While ‘Application-Name’ metadata tag was used on Word 2003 and the document version referred to the application name, the metadata tag ‘Creator’ was used on Word 2007 to refer to the same value. These results identified MS-WORD and MS-PPT as the document types containing the maximum number of metadata describing a document among the various word processing files analyzed in my Datasets.

Largely, Microsoft Word files were found to contain the most number of metadata closely followed by the PowerPoint files. Both sets of documents contain metadata that describe the user context on who created and/or modified the document. This includes metadata such as ‘Author’, ‘Last Author/Group’, ‘Creator’ and ‘Organization’. The metadata also describe the application context that was used in creating/modifying the document. The metadata that come under this category include ‘Application-name’, ‘Application-ver’ and ‘Publisher’ etc.

Notably, not all metadata tags were discovered in all Microsoft Office documents, even within the same type. Often, this can be attributed to the fact that Microsoft Office applications have evolved

over the years and some of the earlier versions did not record as much metadata as compared with recent versions. For instance, Microsoft Office 97 only recorded the ‘Author’ name and did not record application metadata which were only introduced since Microsoft Office 2003.

7.4.2.2 Metadata Availability & Frequency

In this section, I compute the frequency of metadata tags found in the documents from the two datasets. The purpose of the metadata frequency study is to estimate the most commonly occurring metadata in Microsoft Office documents which can be relied upon to determine metadata matches. I have tabulated the relative percentage occurrence of the most relevant metadata in Microsoft Office documents files with regard to forensic investigations. The value was computed as the ratio of the number times that particular metadata was found in that collection to the total number of documents in the collection expressed in percentage. This is shown in Table 7.3. The application metadata tags listed in this table were discovered in all the three different Microsoft Office document types, although the percentage of occurrence in MS-XLS documents alone was marginally lower (between 5–10% lower) compared to MS-WORD and MS-PPT. The highest percentage of occurrence was discovered for ‘Author’ metadata in MS-WORD documents which was as high as 99% and the lowest was for metadata tags ‘Manager’ and ‘Security’ in MS-PPT, which was as low as 3–4%.

In Table 7.3, ‘Author’ represents the metadata naming the author of the document. In some cases, when the author is logged in as part of an organization as listed by metadata ‘Company’, the author is listed by their username within that organization. I believe that identifying these aliases can be very useful during forensic investigations. Often, the aliases could provide reference to the user’s email account within that organization which can subsequently be searched for evidence if required. I was always able to extract the MAC timestamps and other file system metadata (such as filename, file path and file size) for all documents in each of the collections; hence the 100% availability.

		Word Processing Document Metadata						File system Metadata	
Machine source	Total No. of all documents in the collection	Author	Company	Title/Subject	Last-Printed Date	Last-Save - Date	Total/Edit Time	MAC timestamps	File size
Desktop	976	86 %	46 %	69 %	57 %	61 %	79 %	100 %	100 %

Digital Corpora	2970	95 %	91 %	81 %	58 %	96 %	66 %	100 %	100 %
-----------------	------	------	------	------	------	------	------	-------	-------

Table 7.3 Percentage occurrence of metadata tags from across all word processing documents

The ‘Last-Printed-Date’ and ‘Last-save-Date’ metadata recorded by the corresponding applications can often be used to validate if a document has been tampered with. These timestamps could be corroborated with the MAC timestamps to establish timestamp consistency. Frequently, documents are iteratively modified and each version of the modification is stored with similar or different filenames. Interestingly, the original document records the ‘Title’ when it is created, which is often duplicated on all subsequent iterations. In my experiments to determine metadata associations on the Desktop dataset, I discovered several documents that were multiple iterations of a single document and all these documents shared the same value for metadata tag ‘Title’, although the filename was differently recorded in these documents.

The aforementioned situation is also often observed when a particular document is modified based on revisions and new content is added into iterative versions of the same document. In this scenario, the documents may contain the same subject or title in metadata but a different filename for each file. I believe that such discoveries can help forensic examiners rule out documents which need not be included for subsequent analysis. Metadata recording the last printed date and total edit time can often inform examiners on any recent activity on these document(s) or if an unusually large or small amount of time has been spent in editing.

7.4.3 Dataset Characteristics

Metadata pertaining to the ownership metadata family can provide answers to Question 1 listed in Section 7.1 and identify the different individuals connected with a single document or a set of documents. This information was, by and large, available on both the document datasets. The source metadata family identified that all documents were created on a Microsoft Windows XP SP2 machine using the Microsoft Office 1997–2003 software products. In some cases, I also determined the specific names of the computer systems on which the documents were created. This allowed us to group the documents that were created on the same machine, and hence possibly by the same individual. Where I noticed deviations from this expected behavior, it was found to be due to the machine being a shared computer within an organization as identified by the ownership metadata. The timestamp metadata family indicated that the files were last operated on at least six months prior to creation of the dataset. Documents that were created on the same

date and time as those modified or accessed were identified as “derived” documents whose content was obtained from the modified or accessed documents. A one-to-one relationship was established using the source relationship and the structural similarity relationship.

In the following section, I discuss the experiments conducted using AssocGEN by applying my metadata association model to collections of documents for analysis.

7.5 Conducting Experiments

During forensic investigations when examiners are faced with the challenge of analyzing several documents, they may begin the process by conducting keyword and/or string searches on the collection and lists the set of documents that match the criteria [50]. However, since this is purely based on string matches, some context may be missed. One of the advantages of my Metadata Association Model is that the similarity pockets and association groups can aid in discovering missed context, stored in document metadata. My first experiment demonstrates a method to expand the scope of basic string search techniques using my method.

Given an arbitrary document collection, it may be necessary to determine document characteristics such as those listed in Section 7.1. These characteristics will provide a better understanding of the document collection (for triage purposes) for an examiner to take informed decisions during analysis. In my second experiment, I illustrate how to determine document characteristics by identifying metadata matches and generating association groups. I group the similarity pockets from individual metadata matches into association groups to identify documents of importance during document analysis in two document datasets. I generate the association groups in two different ways:

1. obtain metadata from the set of files identified by keyword search and determine metadata associations to generate association groups against the rest of the document collection; and
2. determine metadata associations automatically based on primary metadata to characterize the document collection in terms of number of authors, file name and size similarity, number of organization affiliated with authors, the most active author and so on.

Approach 1 is suitable when there is a subset of documents that are of interest to a forensics examiner and hence, one is interested in identifying those documents that are related the set of

interesting documents. Approach 2 may be suited for situations when an examiner is examining a collection to identify what's contained in them. It can automatically identify and group the related documents and it is then sufficient for an examiner to study the groups generated rather than examine the individual documents. Each group, by virtue of the metadata associations embedded within, are likely to contain related information.

7.5.1 Identifying relevant documents with limited context

When I use keyword searches during analysis, the goal is to obtain all search hits for the values that pertain to the keywords. However, it may also be important to determine what other documents are related to the set of files that were identified from the search hits and thereby expand the scope of the search. Typically, this would involve examining each file from the search hit list and determining additional keywords and conducting further searches. As we've discussed in Chapter 2, this approach is not scalable both in volume and diversity. For instance, if a Microsoft Word document was identified using a search keyword, how would I identify the image files that were created/downloaded along with such a file? Metadata based associations provide a scalable model to determine such related files and group them, and the approach is amenable to automation.

We used two datasets, the Desktop dataset and another containing a user's Internet activity. This dataset was created by combining sets of Web documents and Microsoft Office documents that were downloaded from the Internet in response to Google search queries "*bomb*" and "*explosion*". This produced 154 files consisting of 3 Word documents, 4 PowerPoint presentations, 2 Excel spreadsheets, 20 HTML files, 56 GIF image files, 18 JPEG image files, 38 JavaScript files and 13 Cascading Style Sheets.

7.5.1.1 Method

We identified 3 keywords for each case study and discovered 18 document matches in the first and 16 in the second. I identified the keywords "*architecture*", "*evidence*" and "*research*" for the Desktop dataset. The keywords for the second case study were obtained from filenames in the responses to search queries. I identified the keywords "*explosion*", "*bomb*" and "*c4*" for this dataset. In this case study, the matches were determined from the Microsoft Office documents in addition to the HTML files in the collection. Consequently the metadata associations determined were primarily based on matches in filenames and download timestamp matching with the file creation timestamp from the file system.

We used the AssocGEN analysis engine to extract metadata from the files identified in the keyword search to determine metadata name-value matches with the remaining documents in the collection. The results of the metadata associations discovered are summarized in Table 7.4.

Datasets	Association index (<i>ai</i>)	Grouping efficiency η	No. of Documents discovered from keyword search	Total No. of metadata associations discovered	No. of Documents discovered from metadata associations	Avg. No. of associations per file (Col #3 / Col #2)
Desktop Dataset (976 files)	0.21	0.43	18	108	68	6
User's Internet activity files (154 files)	0.62	0.87	16	132	82	8.25

Table 7.4 Outcomes from determining metadata associations on keyword matches

7.5.1.2 Observations

In the Desktop dataset, I discovered 18 files that matched from a total of 976 files for the 3 keywords. Since all the files in this collection were Microsoft Office documents, I used the metadata name-values from these 18 documents to list the set of all other documents in the collection which generated similarity pockets. When I discovered multiple sets of documents matching on more than one metadata tag, I combined them into association groups and listed the set of all metadata matches discovered among them. For this dataset, the metadata associations were determined based on primary metadata since secondary metadata was not sufficiently populated to generate many associations. The metadata that did generate associations were document Author, Organization, file size and file name similarity. This resulted in the discovery of 68 other documents in the collection that were associated with the documents identified from the keyword search.

In the second dataset, I discovered 16 files that matched from a total of 154 files for the 3 keywords. The files determined by the keyword matches contained 2 Microsoft Word documents and 3 PowerPoint documents in addition to HTML files. The metadata associations for this case study were primarily determined from file system metadata, viz., filename, filesize and file creation timestamp. As discussed earlier in this section, I corroborated the download timestamp with the file creation timestamps and additionally discovered associations based on metadata

filename and filesize. On the Microsoft Office documents, I discovered that the 'Author' was identical in one Word, one PowerPoint and two Excel spreadsheet documents. Moreover, file name and file size similarity measures generated additional matches which enabled the discovery of 82 other documents from this collection.

The number of metadata associations discovered from the Desktop dataset was 108, an average of 6 associations per file identified from the keyword search. Although not all files generated that many associations, the average value is indicative of the relationships exhibited between the documents in that collection. The number of metadata associations discovered from the user's Internet activity dataset was 132, an average of 8.25 associations per file identified from the keyword search.

7.5.1.3 Conclusions

We have thus demonstrated the use of the metadata association model to determine files related to a particular matter of interest that were not readily discovered using the keyword search technique.

7.5.2 Document Analysis

When I analyze large collections of documents, it is useful to determine characteristics such as the total number of authors, the number of single author documents, the number of authors who appear in exactly one file, the largest number of documents authored by a single individual and so on. Typically, classification techniques can identify these characteristics and as discussed in Chapter 6, each classification process uses unique parameters to determine the classes that exist. However, during analysis, it is also necessary to identify documents related to those found in a particular class. For instance, if we were to classify all documents into Microsoft Word documents, PowerPoint slides and Excel spreadsheets, how do we determine all the co-authors of a particular set of documents

- i. who have authored single-author Word documents; and
- ii. who have co-authored PowerPoint slides or Excel spreadsheets?

If such authors exist, then are the set of co-authors identical or different? Some other questions that can be posed during analysis include how do we determine which PowerPoint slides were created, modified, used or downloaded along with a Word document and how many Excel files were used during the time that the document was edited? By their very nature, these questions

necessitate one to study the relationships that exist in the documents, a task that requires content analysis, usually by an individual. Traditional forensic tools offer little help in identifying such critical information when analyzing document collections. If we were able to propose an automated approach to identify such relationships and group the documents, it can save significant human effort. I have identified 12 characteristics for document collections and propose the use of the Metadata Association Model to determine these characteristics using the metadata.

7.5.2.1 Method

On each dataset, I provided the documents from the collection to the AssocGEN analysis engine which traversed the documents and identified multiple homogeneous sources⁵⁵. The parsed metadata determined metadata matches leading to source relationships, existence relationships and structural similarity relationships. In my work, I have focused on the metadata that pertain to the source and ownership metadata families. This focus stems from the need to attribute documents and determine related documents and individuals when conducting the analysis. Naturally, the characteristics identified for this experiment use metadata like ‘Author’, ‘Organization’, ‘Filename’, ‘Filesize’ and so on. I have also included the application metadata family to identify documents created using the same software application with identical or different release versions. The results are tabulated in Table 7.5 and discussed in the sequel.

Characteristic No.	Dataset Characteristics	Desktop (976)	Digital Corpora (2970)
	Number of association groups	108	1892
	Association index (<i>ai</i>)	0.21	0.004
	Grouping efficiency η	0.856	0.293
1	<i>No. of distinct authors</i>	158	3300
2	<i>Most number of documents by one author</i>	170	228
3	<i>No. of authors who have authored more than one document</i>	126	2599
4	<i>Most number of documents similarly named by a single author</i>	36; <i>Stefan</i>	17; <i>J. Scott Peterson</i> ⁵⁶
5	<i>Most number of documents of</i>	98;	9; <i>Jon Heal</i>

⁵⁵ Documents belonging to the same application of the identical software version were treated as coming from a single homogeneous source, differentiated based on document MIME type.

⁵⁶ Since all the files in this repository were renamed (and named similarly) after they were downloaded by Simson Garfinkel, this value is merely the single largest similarity pocket based on ‘Author’.

	<i>similar file size belonging to one author</i>	<i>Stefan</i>	
6	<i>Most number of Organizations single author is affiliated with</i>	4; <i>Stefan</i>	2 ⁵⁷
7	<i>No. of distinct Organizations</i>	71	1098
8	<i>Most number of documents generated within the same Organization</i>	79; <i>QUT</i>	50; <i>US Dept of Agriculture</i>
9	<i>Most number of authors from a single Organization</i>	13; <i>QUT</i>	11; <i>US Dept of Agriculture</i>
10	<i>No. of Organizations generating multiple documents</i>	27	336
11	<i>No. of distinct application names</i>	16	20
12	<i>No. of distinct document titles</i>	207	1703

Table 7.5 Results from determining dataset characteristics for the two datasets

7.5.2.2 Observations

For the desktop dataset, besides the metadata ‘Author’ and ‘Organization/Company’, the metadata ‘Filesize’ and ‘Filename’ generated the largest number of metadata matches. After combing the overlapping similarity pockets, I discovered 108 association groups. In addition to this, there were 32 documents that were removed to the unclassified list as they lacked sufficient metadata. Such files were individually analyzed by examining the forensic image under FTK. Consequently, the efficiency $\eta = 1 - \frac{(108+32)}{976} = 0.856$. This implies that for this collection, more than 85% of the documents are associated with one or more documents and is indicates that several documents were created, accessed or modified under similar contexts. Besides this, providing a reduction in the number of independent documents for further analysis can help a forensics examiner to triage the dataset and quickly focus on a smaller set of documents.

For the Digital Corpora dataset, there were not many common points with regard to where the documents were downloaded from and, therefore, it resulted in a much larger set of association groups. Metadata ‘Author’ and ‘Organization/Company’ generated the largest number of matches amongst their documents. Filesize matches, although present, had few other metadata matches and resulted in a small number of association groups. In all, I determined 1892 association groups and 209 documents in the unclassified list. Therefore, the efficiency for this dataset is computed as $\eta =$

⁵⁷ More than one author was affiliated with 2 organizations. Since there is multiplicity, no name is specified.

$1 - \frac{(1892 + 209)}{2970} = 0.293$. Since these documents were downloaded from the Internet from diverse

sources, the relative association factor was, expectedly, low. Notwithstanding, metadata matches and association groups enable one to group similar documents and analyze related documents together, eliminating the need to repeated or unnecessary analysis. The characteristics defined in Table 7.5 are generic pertaining to the analysis of documents and can be applied to any collection of documents as described below.

The number of distinct authors (Characteristic 1 in Table 7.5) and number of distinct organizations (Characteristic 7) are computed by counting the value field for the 'Author' and 'Company' metadata tags respectively. Since the author field is also multi-valued and a document can have more than one author, each unique author is counted when this is the case. Wherever multiple authors from the same organization are discovered, the individual similarity pockets are merged into association group(s). Thus, I integrate multiple association groups and the size of the largest similarity pocket for metadata tag 'Company' provides the organization generating the largest number of documents. The largest multi pocket generated from the similarity pockets for 'Author' and 'Company' provide the values for Characteristics 2 and 8 in Table 7.5. The number of non-singleton similarity pockets identified for 'Author' and 'Company' provide the values for Characteristics 3 and 10.

By grouping the pockets for 'Author' and 'Filename' similarity the size of the largest multi pocket provides the values for Characteristic 4 in Table 7.5. When I substitute the similarity pockets generated by 'Filename' with those by 'Filesize', then the largest multi pocket thus formed provides the values for Characteristic 5 in Table 7.5. Superimposing the similarity pockets obtained from the 'Author' and 'Company' metadata then reveals the set of authors who share the same organization affiliation. The largest multi pocket formed by superimposing the similarity pockets for 'Author' with the ones for 'Company' provides the values for Characteristic 9 in Table 7.5. Characteristics 11 and 12 are determined in the same manner as Characteristics 1 and 2 using the metadata Application-Name and Title.

7.5.2.3 Conclusions

We have thus demonstrated the application of the MAM to generate metadata associations using the exhaustive mode to determine critical parameters to document analysis. The approach is amenable to automation by virtue of the ubiquity of metadata and the generation of associations leads to the identification of source and ownership relationships during analysis.

In this thesis, I have hitherto established the existence of associations between files based on metadata value matches and demonstrated them for collections of digital image files (refer Chapter 6) and word processing documents (in this chapter). However, I showed in Chapter 2 how the establishment of associations based on timestamps, particularly across files stored across heterogeneous sources can contribute to interpretation challenges. I developed the Provenance Information Model in Chapter 4 to address this challenge and developed a prototype toolkit called UniTIME in Chapter 5. In the sequel, I demonstrate the utility of the model using UniTIME by applying it to two hypothetical case studies.

7.6 Generating Unified Timelines Using PIM

We present here two case studies that use the UniTIME digital time-lining tool; the first case study is based on the DFRWS forensic challenge 2008 [51] which contains four distinct homogeneous sources, maintaining different time references and the second case study is based on synthetic user documents based on a FAT32 file system to detect timestamp inconsistencies by comparing the MAC timestamps against the document metadata timestamps.

7.6.1 Evaluation Criteria

With regard to evaluating the prototype, the following criteria were tested through my experimentation.

1. Did the tool generate a unified timeline?
2. Does the generated timeline include all the events recorded in evidence in the homogeneous sources?
3. Is the generated timeline consistent with the expected outcome?
4. Did the tool identify all timestamp related inconsistencies in the homogeneous sources?

For a successful implementation, the expected answer for all these criteria is ‘yes’. If otherwise, it could indicate either a design flaw or an incomplete implementation in software. The most common reason for the tool’s inability to provide a complete timeline was determined as the inability to complete the extraction of timestamps owing to inaccurate parsing of the timestamps from the homogeneous sources.

7.6.2 Repeatability in Generating Unified Timelines – A Case Study

In this case study, my goal was to generate a unified timeline of the events for the DFRWS Forensic Challenge 2008 [40]. The timestamp interpretation was a challenge due to the fact that the source of digital evidence being a ZIP archive of the userspace folder, the Mozilla Firefox browser history and cache logs and a packet capture, each corresponding to a distinct homogeneous sources on the source⁵⁸. In all, there are four sets of homogeneous sources,

1. a user folder (with US/Eastern time zone reference),
2. the Firefox⁵⁹ browser history (with US/Eastern time zone reference),
3. the Firefox browser cache (with US/Eastern time zone reference), and
4. a packet capture (with UTC time zone reference).

A brief case outline is provided below.

7.6.2.1 DFRWS 2008 Challenge Case Outline

Mr. Steve Vagon, an employee at Saraquiot Corporation, was suspected of smuggling confidential Saraquiot information to an outsider using the company's resources. The source was a hashed archive of the aforementioned contents. Based on the case brief, I set the location of the activities as the east coast of the United States and accordingly the Provenance Information Model for each homogeneous source, i.e., the user folder, the browser logs and the packet capture was set to UTC -0500. The activities on the source were recorded between May 2007 and December 2007. The packet capture contains a network session in December 2007 captured on the IP assigned to the user's machine. I separated the homogeneous sources and parsed the contents for analysis. The homogeneous sources were provided as inputs to UniTIME which extracted the timestamps from the metadata and interpreted them using the PIM.

The list of significant findings for this investigation was recorded by Cohen et al. [40]. A merged timeline of the events was reported by Jokerst et al. [94]. While Cohen et al. apply a fixed time zone offset using Pyflag to interpret the timestamps; Jokerst et al. did not interpret the timestamps,

⁵⁸ The original forensic image provided for the DFRWS 2008 challenge also contains a memory capture from Mr. Steve Vagon's computer. However, for the purpose of this case study, the memory capture has been omitted.

⁵⁹ Although the Firefox browser stores history and cache log timestamps internally in UTC with a local time zone offset, the timestamps were all converted into local time zone when there were compressed into the ZIP format. Besides this, the time zone information was not encapsulated within the archive and hence lost.

but merely used the locally rendered values. In their case, the source was extracted in the same time zone (US/Eastern) as that of the source, which rendered the interpretations unnecessary. However, the approach was not repeatable for corroboration of results. When I retraced the steps reported by Jokerst et al. [94] in a different time zone (AEST), it resulted in an inconsistent timeline where the network activities were found to occur after the reported source acquisition. I demonstrate how my Provenance Information Model was used to address this issue and how I implemented a repeatable solution for generating a unified time-lining using UniTIME.

7.6.2.2 Experiment

In regards to this experiment, the individual homogeneous sources were isolated and preserved along with their respective Provenance Information Model data. The user folder was provided as a forensic image with no explicit time zone bias, since this is achieved using its PIM. The browser logs and the network trace were pre-processed into XML form following which they were also added as sources into UniTIME. The tool was then executed and the generated timeline was compared against the manually enumerated sequence.

7.6.2.3 Applying PIM corrections

Since the source was provided as a compressed zip archive, the files contained within the user folder could only store one timestamp reliably, namely, the last modified timestamp and that only to a 2-second precision. The timestamps on the browser logs, however, are stored differently (UNIX timestamps) with a 1-second precision. The PIM for each homogeneous source is designed specifically for this purpose which was taken to be UTC -0500. In order to obtain the timestamps in UTC, the homogeneous sources had to be selectively corrected; i.e., only the timestamps on the user folder and the browser history and cache had to be corrected, by being shifted forward in time 5 hours. The packet capture internally records timestamps in UTC and the values were readily available. To compute the local timestamps, only the packet capture timestamps had to be shifted back in time 5 hours, while the others were readily available. Thus, each event in each of the homogeneous sources now had one timestamp in UTC and a corresponding timestamp in local time.

7.6.2.4 Verifying Timestamp Resilience

Once the PIM corrections were applied, the timestamps were checked for consistency by validating any assertions pertaining to timestamps for verifying the event sequence that an examiner may record during analysis. In this case study, the assertions pertained to validating the

file system timestamps against the packet timestamps in the network capture. Basically, the timestamps obtained from the user folder correspond to one of three file activity events (create, modify and access). The timestamps obtained from the network packet capture correspond to a packet sent/arrival time. Since it was mentioned in the case brief that the network activity was discovered from within the user folder, it implied that the network packet capture was created before the last file activity event in the user folder. In other words, the *network packet timestamps should precede the last file activity* as recorded from the user folder. The timestamps from each homogeneous source were sorted and I compared the last timestamp from the user folder against the last timestamp on the packet capture. The snapshot of the partial timeline after harmonizing the provenance information across different sources is shown in Figure 7.3. The time zone provided in the figure is in reference with the Pacific Eastern time zone (UTC -0500).

```

2007-12-16 23:06:39 SYSLOG of goldfinger dhclient: DHCPREQUEST on eth0 to 192.168.151.254 port 67
2007-12-16 23:06:39 SYSLOG of goldfinger dhclient: DHCPACK from 192.168.151.254
2007-12-16 23:06:40 SYSLOG of goldfinger dhclient: bound to 192.168.151.130 - renewal in 869 seconds.
2007-12-16 23:06:51 PCAP of Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.12) Gecko/20071020 CentOS/1.5.0.12-6.el5.centos Firefox/1.5.0.12
http://corporate.disney.go.com/corporate/conduct\_manufacturers.html
2007-12-16 23:06:52 PCAP of Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.12) Gecko/20071020 CentOS/1.5.0.12-6.el5.centos Firefox/1.5.0.12
http://hb.disney.go.com/stat/Hitboxcode.js
2007-12-16 23:08:18 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/CA145DAFD01 ( Non-ISO extended-ASCII English text, with very long lines)
2007-12-16 23:08:19 HISTORY First Browse of http://corporate.disney.go.com/environmentality/index.html
2007-12-16 23:08:19 HISTORY Last Browse of http://corporate.disney.go.com/environmentality/index.html
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/OC72616Dd01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/B652618Ad01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/56A7DF65d01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/90F04203d01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/3B2FF872d01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:20 MODIFICATION of .mozilla/firefox/n5q6tfua.default/Cache/DE92D282d01 ( JPEG image data, JFIF standard 1.02)
2007-12-16 23:08:24 MODIFICATION of .mozilla/firefox/n5q6tfua.default/cookies.txt ( Web browser cookie text)
2007-12-16 23:08:39 PCAP of Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.12) Gecko/20071020 CentOS/1.5.0.12-6.el5.centos Firefox/1.5.0.12
http://corporate.disney.go.com/environmentality/index.html
2007-12-16 23:18:48 PCAP of Agent: Mozilla/5.0 (X11; U; Linux i686; en-US) Gecko/20071126 http://en.wikipedia.org/http://en.wikipedia.org/wiki/Main\_Page
2007-12-16 23:21:08 PCAP of Agent: Mozilla/5.0 (X11; U; Linux i686; en-US) Gecko/20071126 http://en.wikipedia.org/http://en.wikipedia.org/wiki/Lee\_Smith\_%28baseball\_player%29
2007-12-16 23:24:05 SETTING (sources) change of .gconf/apps/evolution/addressbook/%gconf.xml
2007-12-16 23:24:06 SETTING (interface) change of .gconf/apps/ekiga/protocols/%gconf.xml
2007-12-16 23:24:06 SETTING (output_device) change of .gconf/apps/ekiga/general/sound_events/%gconf.xml
2007-12-16 23:24:06 SETTING (public_ip) change of .gconf/apps/ekiga/general/nat/%gconf.xml
2007-12-16 23:24:06 SETTING (input_device) change of .gconf/apps/ekiga/devices/video/%gconf.xml
2007-12-16 23:24:06 SETTING (input_device) change of .gconf/apps/ekiga/devices/audio/%gconf.xml
2007-12-16 23:24:06 SETTING (output_device) change of .gconf/apps/ekiga/devices/audio/%gconf.xml
2007-12-16 23:24:08 MODIFICATION of .gnome2/gnomemeeting ( ASCII text)
2007-12-16 23:24:09 SETTING (size) change of .gconf/apps/ekiga/general/user_interface/druid_window/%gconf.xml
2007-12-16 23:24:09 SETTING (position) change of .gconf/apps/ekiga/general/user_interface/druid_window/%gconf.xml
2007-12-16 23:24:11 SETTING (position) change of .gconf/apps/ekiga/general/user_interface/main_window/%gconf.xml
2007-12-16 23:24:22 PCAP of Agent: Ekiga http://ekiga.net/ip/
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/protocols/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/sound_events/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/nat/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/user_interface/druid_window/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/user_interface/main_window/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/user_interface/%gconf.xml ( empty)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/general/%gconf.xml ( empty)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/devices/video/%gconf.xml ( XML)
2007-12-16 23:25:04 MODIFICATION of .gconf/apps/ekiga/devices/audio/%gconf.xml ( XML)

```

Figure 7.3 Snapshot of the partial timeline obtained sing UniTIME after harmonizing the provenance information between the different sources of digital evidence

7.6.2.5 Conclusions

Once the timestamps passed the consistency checks, they are digitally time-lined and rendered. The resultant timeline was *in agreement* with the timelines produced by Cohen et al. and Jokerst

et al. [40, 94] and was found to be *repeatable* as I obtained an identical timeline by repeating this experiment for 3 other different time zones.

7.6.3 Validating Document Consistency Using Assertion Testing in PIM – A case study

For this case study, I created a synthetic user folder to detect timestamp inconsistencies on documents during analysis. While earlier research [19, 23, 171, 184] has used only file system timestamps to detect anomalies, I have considered timestamps both from the file system and document metadata. The user folder was generated on a FAT32 file system as shown in Figure 7.4. The folder contained an archive file and a directory, both called *Sample*; the directory contained multiple sub-directories and files.

7.6.3.1 Experiment

In this directory, 4 Microsoft Word document files (Doc1.docx, Doc2.docx, Doc3.docx and Doc4.docx) were created at different levels in the hierarchy and the system was set to a different time zone (the time zone changed from UTC +1000 to UTC +0500). Since FAT32 does not record the time zone, the current system time will be regarded as the source for recording these timestamps. All directories and the text files were created earlier than mid-2009 and were therefore insensitive to the time zone shift applied in June 2011. After the time zone change, the Word documents were accessed a few times. Among the Word documents, Doc1.docx and Doc4.docx were updated while the other two were merely read. This action would ensure that the document timestamps and the file system timestamps corresponding to `LAST_MODIFIED` and `LAST_ACCESS` are updated in accordance with the present system time zone. Then, I imaged (using DD) the folder and analyzed it using UniTIME.

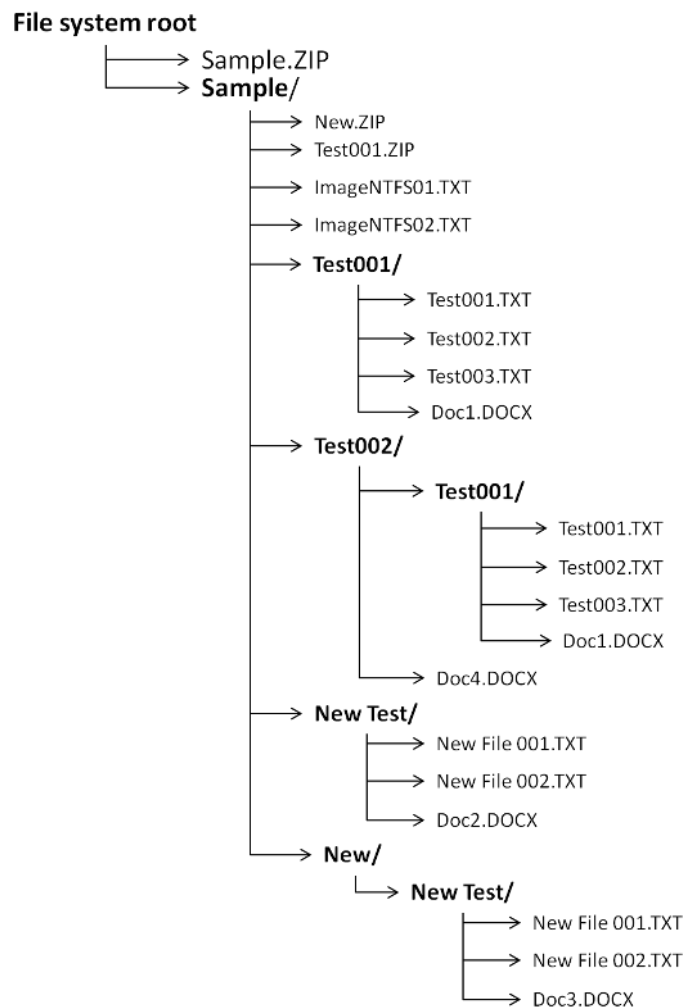


Figure 7.4 The synthetic User folder structure for detecting timestamp inconsistencies

7.6.3.2 Setting up Assertions

At the time of acquiring the source, the system time zone was UTC +0500 and was set in the Provenance Information Model. The first of my two hypothetical assertions recorded into the PIM is given below:

1. Document creation times must always precede document last modification and last access times.

Since the documents were regarded as being generated according to the current system time, it was expected that all files are first created ahead of file operations such as *file access* or *file modify*. When a document breaches this assertion, it can indicate one of two possibilities, i.e., the document was a copy (either direct or altered) of a document from some removable or network storage or the timestamps were intentionally tampered with. While the former can be a common

occurrence in a file system (usually in FAT and NTFS file systems, as noted in Footnote 60, p. 216), there is still value in identifying such documents, especially in an automated manner using the PIM. The second hypothetical assertion recorded was:

2. File system MAC timestamps on documents should be greater than or equal to the document metadata timestamps.

The application is responsible for modifying the document metadata while the operating system controls the file system metadata. Hence all modifications by the application on documents should have occurred before the operating system modified the respective MAC timestamps. When a document breaches this assertion, it can indicate timestamp tampering.

7.6.3.3 Testing Assertions to Detect Inconsistencies

The UniTIME tool traversed the file system hierarchy on the forensic image and applied the provenance information model corrections to the timestamps in metadata. All text files passed the assertions but the 4 Microsoft Word documents breached them. This is attributed to the fact that when the Word documents were modified after the time zone was changed, the LAST_MODIFIED timestamps were updated according to local system time and since the system time was temporally behind the CREATION timestamps on the respective documents. The output from the result of the breach is shown in Table 7.6. The tool identified the files in breach and flagged a message stating the type and nature of the inconsistency discovered.

Sl. No.	Artifact name	Alert TYPE	Details
1	Doc1.doc	alert for TS inconsistency	file metadata inconsistent
2	Doc4.doc	alert for TS inconsistency	file metadata inconsistent
3	Doc3.doc	alert for TS inconsistency	file metadata inconsistent
4	Doc3.doc	alert for TS inconsistency	metadata and MAC inconsistent
5	Doc2.doc	alert for TS inconsistency	file metadata inconsistent
6	Doc2.doc	alert for TS inconsistency	metadata and MAC inconsistent

Table 7.6 Output from temporal assertion testing

7.6.3.4 Conclusions

The message “file metadata inconsistent” was flagged when the tool determined that the document metadata had an inconsistency on Assertion 1, i.e., the Microsoft Word metadata “Creation-Date”, “Last-Saved-Date” and “Last-Printed-Date” were found to be out of order. The message “metadata and MAC inconsistent” was flagged when MAC timestamps (one or more) were less than the document metadata timestamps. Such inconsistencies are detected and alerted for preemptive corrective actions on the identified documents⁶⁰ during analysis.

7.7 Discussion

Metadata underlines the context to describe the *situational similarity* during the life cycle of the documents and files stored on digital sources. Document metadata store a variety of information regarding who and how a document was created and operated on such as author, organization, document format, application type, application version, MAC timestamps and document timestamps. Such information are related to who created the document and how (formatting information) it was created. Document metadata may also record information about where it was created (geo-tagging), number of pages/slides, formatting type, encoding type and so on.

Rowe and Garfinkel [165] have analyzed a large repository of documents to determine anomalous documents. They computed statistical characteristics using directory metadata and identified the top and bottom 5 percentile in the repository as outliers. These statistical characteristics are applicable to file size, number of similarly named files, related files residing in the same directory and so on. Their dataset does not pertain to a single investigation and hence the anomalies identified correspond to misnamed files and duplicate copies of files. In a similar vein, by applying the MAM to document collections, not only can one identify such anomalies (files with unusually large file sizes and similarly named files will generate their own respective similarity pockets), those files would be grouped together informing the forensic examiner of exactly how many files there are in each category and also identifying any additional metadata similarity that associate these documents. Identifying metadata associations among the documents will group documents that stand out from the rest and thus could assist an examiner in filtering out the subset

⁶⁰ The apparent inconsistency in timestamps as demonstrated in this case study can also occur when a Microsoft Word document is copied across computers, especially set to different time zones. The Microsoft Word application which manages the document metadata does not often modify these timestamps on a copy/move operation which always affects MAC timestamps. The document metadata are only changed when the application is used to operate on the file. Therefore, such an inconsistency does not always imply malicious activity. However, whenever such an inconsistency is discovered, the tool prompts the examiner who may then choose the appropriate course of action.

of documents from a large collection that require further analysis while simultaneously identifying those that can be safely excluded, reducing the number of digital artifacts requiring manual analysis. For example, if an examiner comes across an unusually large association group (for my purposes, the particular set of metadata that were involved in the generation of this association group is not relevant), the examiner may determine one of two things:

1. the documents in the association group all belong to one principal author; or
2. the documents in the association group belong to multiple different authors all of whom have co-authored with the principal author on different documents.

7.8 Chapter Summary

In this chapter I evaluated the Metadata Association Model on word processing documents. I discussed the identification of metadata pertaining to the four metadata families as identified in Chapter 5. I grouped large collections of documents based on the metadata associations identified in a manner that facilitated document analysis.

In the next chapter, I summarize the research challenges addressed and my research outcomes in this thesis. I present a discussion on the scope of this research and identify areas for future work.

This page is intentionally left blank

“An expert is one who knows more and more about less and less until he knows absolutely everything about nothing.”
- Nicholas Murray Butler

8. Conclusions and Future Work

Digital forensics concerns the analysis of electronic artifacts to reconstruct events such as cybercrimes. Rapid technological advances during the last decade have resulted in a proliferation of digital devices. Besides this, it is becoming increasingly common for individuals to own multiple digital devices; today, any individual chosen at random is likely to possess a workstation, a laptop, mobile phone, a couple of USB flash drives, and a GPS receiver, not to mention online user profiles depicting their personal information. During a digital investigation, such heterogeneous devices have to be forensically examined and analyzed. Therefore, contemporary digital forensics is forced to contend with such heterogeneity. Two major challenges surface as a result, viz., *diversity* and *volume*. To address the diversity and the volume challenges, I identified associations among digital artifacts across heterogeneous sources of digital evidence that represent the syntactic and semantic relationships. I used metadata as the instrument to determine these associations (based on metadata value matches).

This research produced a framework to support forensic analyses by identifying associations in digital evidence using metadata. It showed that metadata based associations can help uncover the inherent relationships between heterogeneous digital artifacts thereby aiding in the reconstruction of past events by identifying artifact dependencies and time sequencing. It also showed that metadata association based analysis is amenable to automation by virtue of the ubiquitous nature of metadata across forensic disk images, files, system and application logs and network packet captures. The results prove that metadata based associations

can be used to extract meaningful relationships between digital artifacts, thus potentially benefiting real-life forensics investigations.

8.1 Research Objectives & Contributions

We developed a framework for automatically identifying source metadata-based associations in digital evidence and for grouping the related artifacts. I have shown experimentally that my approach can be used for answering the six forensic questions of *who*, *what*, *when*, *where*, *how* and *why* posed by Casey [32].

8.1.1 Objectives of this Research

In accordance with the goals and objectives stated in Chapter 1, the following objectives were targeted in this research.

1. To develop an understanding for the treatment of metadata in digital evidence by different forensic and analysis tools and to integrate the functionalities of different existing tools for analyzing digital evidence.
2. To develop an understanding of how to generate metadata associations based on syntactic and semantic relationships between digital artifacts, using metadata matches across arbitrary types of digital artifacts.
3. To develop an understanding for the semantics linked to metadata associations and their interpretation in a forensic context, to allow us to produce intuitive groupings of digital artifacts, of both homogeneous and heterogeneous natures, for forensic analysis.

8.1.2 Contributions from this Research

Based on the understanding arising out of achieving the goals and objectives of this project, the following were the salient contributions from this research:

1. We conducted a review of contemporary forensic and analysis tools to abstract the different functionalities supported to analyze different sources of digital evidence. This review culminated in the design of the *functional Forensic Integration Architecture* which consolidated these functionalities and defined a new layer to group artifacts based on metadata associations. I developed a prototype toolkit called AssocGEN analysis engine based on the *f*-FIA architecture which spans over 20000 lines of Java code and consists of

multiple modules. The modules are pluggable at runtime and can access and parse files and folders from most common file systems such as FAT32, NTFS, EXT2, EXT3 and HFS+, web page visitation and cache logs on browser applications and network packets contained within packet captures.

2. We conducted experiments to elicit the syntax and semantics associated with metadata associations which were determined through the identification of metadata matches. I generalized my findings which resulted in the *Metadata Association Model* (MAM) for identifying metadata-based associations across the digital artifacts from multiple sources of digital evidence. I also developed the associated theory to study the formation of metadata based matches across heterogeneous sources of digital evidence and algorithms to identify specific artifact relationships that can be of interest during forensic analysis.
3. The identification of data items from homogeneous and heterogeneous sources, whether regarding files, log records or network packets, *to discover the higher-order associations or relationships via the metadata*. This was demonstrated by the successful grouping of digital image files and word processing documents belonging to different file formats and discussed the formation of association groups across multiple source classes by determining metadata matches between them.
 - a. We studied the use of the Metadata Association Model to analyze collections of digital image files and demonstrated two methods of grouping the related digital images. I illustrated the use of digital image file relationships to determine instances of image downloads and identify the origin of these downloads.
 - b. We studied the use of the Metadata Association Model to analyze collections of word processing documents and demonstrated two methods of grouping the related documents. I illustrated the use of word processing documents relationships to determine instances of document doctoring during analysis.
4. The development of the *Provenance Information Model* (PIM) to provide timestamp resilience in metadata for interpretation. I developed a prototype toolkit called UniTIME unified timelining tool based on the *f*-FIA architecture which spans over 6000 lines of Java code and consists of multiple modules. The modules are pluggable at runtime and can access and parse timestamps from files and folders on file systems such as FAT32, NTFS, EXT2, EXT3 and HFS+, web page visitation and cache logs on browser

applications and network packets contained within packet captures. I demonstrated the execution of this tool to generate unified timelines using contemporary case studies involving FAT32 file systems and ZIP file formats and validating event consistency across heterogeneous sources.

8.2 Limitations & Future Directions

The *Metadata Association Model* (MAM) developed in this thesis has explored the realms of representing the interdependencies between digital artifacts on one or more sources of digital evidence. In view of the goal of digital forensics which attempts to develop a scientific method to reconstruct past events, I indicate future directions for my research in this section.

Exploring Domain Heuristics for Computational Benefit. The metadata association model is a novel way to look at digital artifacts and their interdependencies as similarity pockets and association groups based on metadata value matches. Even though I used deterministic algorithms to elicit artifact relationships, one may have to contend with exponentially large number of associations as the volume of digital evidence increases. It is worthwhile exploring whether heuristics based on domain specific information are likely to yield computational benefits and aid in the identification of targeted analysis. This, in the author's opinion, is likely to open new vistas in digital forensics.

Uniform Representation for Digital Artifacts. In my research I explored the analysis of digital artifacts such as files, log records and network packets based on the associations identified using respective metadata present in them. It may be worthwhile to look at the development of standards to unify the creation of metadata and the identification of metadata based value matches to aid in the forensic reconstruction process. While doing so, one may develop a mapping between semantically equivalent metadata values to identify metadata associations and progressively establish an up-to-date ontology of digital artifacts across contemporary sources of digital storage.

8.3 Conclusion

Metadata represents an important component of a digital artifact and contain contextual and situational information. I showed that metadata can be invaluable during analysis and can aid in the identification of key relationships across heterogeneous sources of digital evidence. I believe that my framework and the metadata association model are inherently capable of absorbing future

growth in metadata both in relation to context and situations. I believe that, in time, my *Metadata Association Model* could be integrated into mainstream forensic toolkits for use in day-to-day investigations. I hope that my research has provided the necessary stimulus to achieve unified forensic analyses.

References

- [1] Agrawal N., Bolosky W J., Douceur J R., & Lorsch J R., (2007), A Five-Year Study of File system metadata, *ACM Transactions on Storage*, Vol. 3(3), pp. 9:1-9:32
- [2] Allen J (1991). Time and Time again: The Many Ways to Represent Time, *Intl. Journal of Intelligent Systems* Vol. 6(4), pp. 1-14.
- [3] Alink, W., Bhoedjang, R. A. F., Boncz, P. A., & de Vries, A. P. (2006). XIRAF - XML-based indexing and querying for digital forensics. *Digital Investigation, The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, 3(Supplement 1), pp. 50-58.
- [4] Alvarez P. Spl. Agt. (2004). Using Extended File Information (EXIF) File Headers in Digital Evidence Analysis, *Intl. Journal of Digital Evidence* Vol. 2(3), pp. 1-5.
- [5] Apache Software Foundation, Apache Tika – content analysis toolkit, <http://tika.apache.org/>, last retrieved on July 12, 2011
- [6] Arasteh A R and Debbabi M. (2007). Forensic Memory Analysis: From Stack and Code to Execution History, *Digital Investigations, Proceedings of the 7th Annual Digital Forensic Research Workshop (DFRWS '07)*, 4(Supplement 1), pp. S114-S125.
- [7] Arasteh A R, Debbabi M, Sakha A and Saleh M. (2007). Analyzing Multiple logs for Forensic Evidence, *Digital Investigations, Proceedings of the 7th Annual Digital Forensic Research Workshop (DFRWS '07)*, 4(Supplement 1), pp. S82-S91.
- [8] --- Are Windows file timestamps time zone aware? <http://superuser.com/questions/109922/are-windows-file-timestamps-timezone-aware>, last retrieved on July 12, 2011
- [9] Association of Chief Police Officers (ACPO) (2003). Good Practice Guide for Computer Based Electronic Evidence, *NHTCU Publications*, pp. 1-51.
- [10] Barik, M. S., Gupta, G., Sinha, S., Mishra, A., & Mazumdar, C. (2007). Efficient techniques for enhancing forensic capabilities of Ext2 file system. *Digital Investigation*, 4(Supplement 1), pp. 55-61.
- [11] Bayram S., Sencar H. T. and Memon N. (2008). Classification of Digital Camera-Models Based on Demosaicing Artifacts, *Digital Investigation*, Vol. 5(1), pp. 49-59, 2008.
- [12] Beebe, N. L., & Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2), pp. 147-167.
- [13] Beebe, N. L., & Clark, J. G. (2007). Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. *Digital Investigation*, 4(Supplement 1), pp. 49-54.

- [14] Berghel H. (2007). Hiding Data, Forensics and Anti-Forensics, *Communications of the ACM* Vol. 50(4), pp. 15-20.
- [15] Bogen A C and Dampier D A. (2005). Preparing for Large scale Investigations with case Domain modeling, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.
- [16] Bohm K. and Rakow T. C. (1994)., Metadata for Multimedia Documents, *In Proceedings of ACM SIGMOD RECORD 1994*, Vol. 23(4), pp. 21-26.
- [17] Boutell M. and Luo J. (2004). Photo Classification by Integrating Image Content and Camera Metadata, *ICPR, 17th International Conference on Pattern Recognition (ICPR'04)*, Vol. 4, pp.901-904.
- [18] Boutell M. and Luo J. (2005). Beyond pixels: Exploiting Camera metadata for Photo Classification, *Pattern Recognition, Image Understanding for Photographs*, June 2005, Volume 38(6), pp. 935-946, ISSN 0031-3203, DOI: 10.1016/j.patcog.2004.11.013
- [19] Boyd C, & Forster P., (2004), Time and Date Issues in Forensic Computing – A case study, *Digital Investigations*, Vol. 1(1), pp. 18-23.
- [20] Brand A, Daly F., & Meyers B., (2003). Metadata Demystified, *The Sheridan and NISO Press*, http://www.niso.org/standards/resources/Metadata_Demystified.pdf, ISBN: 1-880124-59-9, pp. 1-19.
- [21] Brinson A., Robinson A. and Rogers M. (2006). A Cyber-Forensics Ontology: Creating a new approach to studying Cyber Forensics., *Digital Investigations, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, 3(Supplement 1), pp. S37-S43.
- [22] Buchholz F and Spafford E H. (2004). On the Role of System metadata in Digital Forensics, *Digital Investigations*, 1(1), pp. 298-309.
- [23] Buchholz F., & Tjaden B., (2007), A Brief History of Time, *Digital Investigations, Proceedings of the 7th Annual Digital Forensic Research Workshop (DFRWS '07)*, Vol. 4S (2007), pp. S31-S42.
- [24] Calhoun W C. and Coles D. (2008). Predicting the Types of File fragments, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, Vol. 5(1), pp. S14-S20.
- [25] Carrier B D., (2003), Sleuthkit, <http://www.sleuthkit.org/sleuthkit/>, last retrieved on July 12, 2011
- [26] Carrier B D. (2005). File system Forensic Analysis, *Addison Wesley Publishers*, ISBN 0-32-126817-2
- [27] Carrier, B. D., (2003). Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence (IJDE)*, Vol. 1(4), pp. 1-12.
- [28] Carrier B D and Spafford E H. (2003). Getting Physical with the Digital Investigation Process, *Intl. Journal of Digital Evidence* Vol. 2(2), pp. 1-20.

- [29] Carrier, B. D., & Spafford, E. H. (2004). An Event-based Digital Forensic Investigation Framework, *Paper presented at the 4th Annual Digital Forensic Research Workshop (DFRWS '04)*.
- [30] Carrier, B. D., & Spafford, E. H. (2006). Categories of digital investigation analysis techniques based on the computer history model. *Digital Investigation, The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, 3(Supplement 1), pp. 121-130.
- [31] Case A, Cristina A, Marziale L, Richard G G and Roussev V. (2008). FACE: Automated Digital Evidence Discovery and Correlation, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, 5(Supplement 1), pp. S65-S75.
- [32] Casey E. (2011). Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, *Academy Press Publications 3/e*, ISBN 978-0-12-374268.
- [33] Casey E. (2007). What does “forensically sound” mean? *Digital Investigations (Editorial)*, Vol. 4(1), pp. 49-50.
- [34] Casey E., (2009), Timestamp Misinterpretations in File systems, <http://blog.cmdlabs.com/tag/timestamps/>, last retrieved on July 12, 2011
- [35] Castiglione, A., De Santis, A., & Soriente, C. (2007). Taking advantages of a disadvantage: Digital forensics and steganography using document metadata. *Journal of Systems and Software, Component-Based Software Engineering of Trustworthy Embedded Systems*, Vol. 80(5), pp. 750-764.
- [36] Carvey H., (2005), Windows Forensics and Incident Recovery, *Addison Wesley Publishers*, ISBN 0-321-20098-5.
- [37] Ciardhuain S O. (2004). An Extended Model for Cybercrime Investigations, *Intl. Journal of Digital Evidence* Vol. 3(1), pp. 1-22.
- [38] Cleverdon, C. W., Mills, J. & Keen, E. M. (1966). Factors determining the performance of indexing systems. Cranfield, UK: *Aslib Cranfield Research Project*, College of Aeronautics. (Volume 1:Design; Volume 2: Results)
- [39] Cohen M I. (2008). PyFlag – An Advanced Network Forensic Framework, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, 5(Supplement 1), pp. S112-S120.
- [40] Cohen M I., Collet D J., & Walters A., (2008), Submission for Forensic Challenge 2008, *Forensic Challenge 2008*, I Place, http://sandbox.dfrws.org/2008/Cohen_Collet_Walters/, last retrieved on July 12, 2011
- [41] Cohen M. I., Garfinkel S. & Schatz B. (2009). Extending the Advanced Forensic Format to accommodate Multiple Data Sources, Logical Evidence, Arbitrary Information and Forensic Workflow, *Digital Investigations, Proceedings of the 9th Annual Digital Forensic Research Workshop (DFRWS '09)*, Vol. 6 (2009), pp. S57-S68.
- [42] Combs G. (1998). Wireshark – Network Protocol Analyzer, <http://www.wireshark.org/about.html>,

last retrieved on July 12, 2011

- [43] Common Digital Evidence Storage Format Working Group (CDESF-WG) (2006). Standardizing Digital Evidence Storage, *Communications of the ACM*, Vol. 49(2), pp. 67-68.
- [44] Common Digital Evidence Storage Format Working Group (CDESF-WG) (2006). Survey of Disk Image Storage Formats, *Paper presented at the 6th Annual Digital Forensic Research Workshop (DFRWS '05)* pp. 1-18.
- [45] --- Converting NTFS timestamps to FAT Timestamps, <http://stackoverflow.com/questions/2247339/converting-ntfs-timestamps-to-fat-timestamps>, last retrieved on July 12, 2011
- [46] Chow K., Law F., Kwan M., & Lai P., (2007), The Rules of Time on NTFS file system, *In Proceedings of the 2nd International Workshop on Systematic Approaches to Digital Forensic Engineering*, April 2007.
- [47] Denecke K., Risse T. and Baehr T., (2009)., Text Classification Based on Limited Bibliographic Metadata, *In Proceedings of the Fourth IEEE International Conference on Digital Information Management, ICDIM 2009*, ISBN 978-1-4244-4253-9, pp.27-32
- [48] Denning. P. J., (1981). ACM president's letter: smart editors. *Communications of the ACM*, Vol. 24(8) (August 1981), pp. 491-493.
- [49] Denning P. J., (1981). Experimental Computer Science: Performance analysis: Experimental computer science as its best, *Communications of the ACM*, Vol. 24(11), pp. 725–727, Nov 1981.
- [50] DFRWS Technical Committee. (DFRWS) (2001). A Road map for Digital Forensic Research: DFRWS Technical Report, *DTR - T001-01 FINAL*
- [51] DFRWS 2008 Forensic Challenge. (DFRWS), *8th Annual Conference*, <http://www.dfrws.org/2008/challenge/submission.shtml>, last retrieved on July 12, 2011
- [52] Digital Imaging Group Inc., (2001), DIG35 Specification – Metadata for Digital Images, Version 1.1 April 16th 2001 Working Draft, *Digital Imaging Group Inc.*, 2001-04-16.
- [53] Ding X. and Zou H. (2011)., Time Based Data Forensic and Cross Reference Analysis, *In Proceedings of the ACM Symposium on Applied Computing 2011, TaiChung, Taiwan*, ISBN: 978-14503-0113-8, pp. 185-190.
- [54] Dolan-Gavitt B. (2008), Forensic analysis of Windows Registry in Memory, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, 5(Supplement 1), pp. S26-32.
- [55] Dyreson C. E. & Snodgrass R. T., (1993). Timestamps semantics and representation, *Journal of Information Systems*, Vol. 18(3), pp. 143-166.
- [56] Eckstein K and Jahnke M. (2005), Data Hiding in Journaling File Systems, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.

- [57] Eckstein K. (2004)., Forensics for Advanced UNIX File Systems, *In Proceedings of the 2004 IEEE Workshop on Information Assurance, West Point, New York*, ISBN: 0-7803-8572-1, pp. 377-385.
- [58] EXIF Specification Document (2002), JEITA CP-3451, *Standard of Japan and Information Technology Association, Exif Version 2.2*, <http://www.exif.org/Exif2-2.PDF>, last retrieved on July 12, 2011
- [59] Fathi, M., Adly, N., and Nagi, M. (2004)., Web Documents Classification Using Text, Anchor, Title and Metadata Information, *In Proceedings of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications*, pp. 1-8.
- [60] Fei, B. K. L., Eloff, J. H. P., Olivier, M. S., & Venter, H. S. (2006). The use of self-organising maps for anomalous behavior detection in a digital investigation. *Forensic Science International 17th Triennial Meeting of The International Association of Forensic Sciences 2005, Hong Kong*, Vol. 162(1-3), pp. 33-37.
- [61] Feitelson D. G., (2006). Experimental Computer Science: The Need for a Cultural Change.
- [62] Fernandez, E., Pelaez, J., & Larrondo-Petrie, M. (2007, Jan). *Attack Patterns: A New Forensic and Design Tool*. Paper presented at the Digital forensics; Advances in digital forensics III: IFIP International Conference on Digital Forensics, Orlando, FL.
- [63] Zhangjie Fu, Xingming Sun, Yuling Liu, Bo Li, Forensic investigation of OOXML format documents, *Digital Investigation*, Volume 8(1), July 2011, Pages 48-55.
- [64] Garfinkel S L. (2006). AFF: A New format for Storing Hard Drive Images, *Communications of the ACM* Vol. 49(2), pp. 85-87.
- [65] Garfinkel S L. (2006). Forensic Feature Extraction and Cross Drive Analysis, *Digital Investigation* 3(Supplement 1), pp. S71-S81.
- [66] Garfinkel S L., Malan D., Dubec K., Stevens C and Pham C. (2006). Advanced Forensic Format: An Open Extensible Format for Disk Imaging, *Proceedings of the Second Annual IFIP WG 11.9 International Conference on Digital Forensics, Advances in Digital Forensics II*, M. Olivier and S. Sheno (Eds.), Springer, Boston, 2006. (ISBN: 0-387-36890-6) pp. 17-31.
- [67] Garfinkel S L. (2009). Digital Forensic Research: The next 10 years, *Digital Investigations, In Proceedings of the 10th Annual Conference on Digital Forensic Research Workshop (DFRWS '10)*, Vol. 7(2010), pp. S64-S73.
- [68] Garfinkel S L. Farrell P., Roussev V. and Dinolt G. (2009), Bringing Science to digital forensics with standardized forensic corpora, *Digital Investigation, In Proceedings of the 9th Annual Conference on Digital Forensic Research Workshop (DFRWS '09)*, Vol. 6, pp. S2 – S11.
- [69] Garfinkel S., (2009), Automating Disk Forensic Processing with Sleuthkit, XML and Python, *In Proceedings of the 2009 Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2009)*, Berkeley, California, ISBN: 978-0-7695-3792-4, pp. 73-84.

- [70] Garfinkel S., (2009)., Digital Corpora – Govdocs1, <http://digitalcorpora.org/corpora/files>, last retrieved on July 12, 2011
- [71] Garfinkel S., (2009)., Digital Corpora – Govdocs1 – Simple statistical report, <http://digitalcorpora.org/corpora/files/govdocs1-simple-statistical-report>, last retrieved on July 12, 2011
- [72] Gehani, A., & Reif, J. (2007, Jan). *Super-Resolution Video Analysis for Forensic Investigations*. Paper presented at the Digital forensics; Advances in digital forensics III: IFIP International Conference on Digital Forensics, Orlando, FL.
- [73] Geiger M. (2005). Evaluating Commercial Counter Forensic Tools, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.
- [74] Gerber, M., & Leeson, J. (2004). Formalization of computer input and output: the Hadley model. *Digital Investigation*, Vol. 1(3), pp. 214-224.
- [75] Gilligan J. (2001). Beating the daylight savings Time bug and getting the correct File Modification times, *Code Project –Date and Time*, <http://www.codeproject.com/KB/datetime/dstbugs.aspx>, last retrieved on July 12, 2011
- [76] Gladyshev, P., & Patel, A. (2005), Formalizing Event Time Bounding in Digital Investigations, *Intl. Journal of Digital Evidence*, Vol. 4(2), pp. 1-14.
- [77] Gladyshev, P., & Patel, A. (2004). Finite state machine approach to digital event reconstruction. *Digital Investigation*, Vol. 1(2), 130-149.
- [78] Gloe T., & Bohme R., (2010), The Dresden Image database for benchmarking digital image forensics, *In Proceedings of the ACM Symposium on Applied Computing 2010 (SAC 2010)*, ISBN 978-1-60558-639-7
- [79] Gloe T. (2010). The Dresden Image Database gallery, <http://forensics.inf.tu-dresden.de/ddimgdb/positions/index>, last retrieved on July 12, 2011
- [80] Gupta M R, Hoeschele M D and Rogers M K. (2006). Hidden Disk Areas: HPA and DCO, *Intl. Journal of Digital Evidence* Vol. 5(1) pp. 1-8.
- [81] Hamilton E. (1992). JPEG Specification Document, *C-Cube Microsystems, Version JFIF 1.02*, <http://www.w3.org/Graphics/JPEG/jfif3.pdf>, last retrieved on July 12, 2011
- [82] Hargreaves C, Chivers H and Titheridge D. (2008). Windows Vista and Digital Investigations, *Digital Investigations*, Vol. 5(1), pp. 34-48.
- [83] Harms K. (2006). Forensic Analysis of System Restore points in Microsoft Windows XP, *Digital Investigations, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, 3(1), pp. 151-158.
- [84] Harry Potter Films – Downloads, Warner Bros (WB) Entertainment, <http://harrypotter.warnerbros.com/harrypotterandthedeathlyhallows/mainsite/index.html>, last

retrieved on July 12, 2011

- [85] Hildreth, C. R. (2001). Accounting for users' inflated assessments of on-line catalogue search performance and usefulness: an experimental study. *Information Research*, Vol. 6(2), Available at: <http://InformationR.net/ir/6-2/paper101.html>
- [86] Hosmer C and Hyde C. (2003). Discovering Covert Digital Evidence, *Paper presented at the 3rd Annual Digital Forensic Research Workshop (DFRWS '03)*.
- [87] Hosmer C. (2006). Digital Evidence Bag, *Communications of the ACM* Vol. 49(2), pp. 69-70.
- [88] Huang H-C, Fang W-C, Chen S-C, (2008). Copyright Protection with EXIF Metadata and Error Control Codes, Security Technology, *International Conference on, 2008 International Conference on Security Technology*, pp. 133-136.
- [89] --- How to prove that the date of a Word document was not tampered?
<http://superuser.com/questions/266787/how-to-prove-that-the-date-of-a-word-document-was-not-tampered>, last retrieved on July 12, 2011
- [90] Ives, K., Update Date/Time stamps on ZIP files Version 1.4,
<http://www.freevbcode.com/ShowCode.asp?ID=2510>, last retrieved on July 12, 2011
- [91] Jeyaraman S and Atallah M J. (2006). An Empirical Study of Automatic Event Reconstruction Systems, *Digital Investigations, Proceedings of the 6th Annual Digital Forensic Research Workshop (DRFWS '06)*, 3(Supplement 1), pp. S108-S115.
- [92] Jiang X., Walters A., Xu D., Spafford E, Buchholz F. and Wang Y. (2007)., Provenance-Aware Tracing of Worm Break-In and Contaminations: A Process Coloring Approach, *In Proceedings of the 24th IEEE International Conference on Distributed Computing Systems, (ICDCS 2006)*, Lisbon, Portugal, ISBN: 0-7695-2540-7, pp. 38.
- [93] Johnston A and Reust J. (2006). Network Intrusion Investigation – Preparation and Challenges, *Digital Investigations* 3(1), pp. 118-126.
- [94] Jokesrst R M., Kouskoulas Y A., Saur K J., Snow K Z., & Whipple B M., Recreating Malicious Network User Activity, *Submission to the Forensic Challenge 2008*, II Place,
http://sandbox.dfrws.org/2008/JHU_APL/, last retrieved on July 12, 2011
- [95] Jones, K J., Bejtlich R, & Rose C W., (2004), Real Digital Forensics – Computer Security and Incident Response, *Addison Wesley Publishers*, ISBN 0-321-24069-3
- [96] Katz P., (2007). .ZIP file format Specification document, *Appnote.txt Version 6.3.2, PKWARE.Inc*,
<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>, last retrieved on July 12, 2011
- [97] Kee E. and Farid H. (2010). Digital Image authentication from Thumbnails, *In Proceedings of the SPIE Symposium on Electronic Imaging*,
- [98] Kee E, Johnson M. K. and Farid H. (2011), Digital Image Authentication from JPEG headers, *IEEE Transactions on Information Forensic and Security (In Press)*, 2011.

- [98a] Kessler G. (2003)., "Computer Forensics: An Investigative Overview" *Annual meeting of the Vermont Chapter of the FBI National Academy Associates*, March 2003, Rutland, VT.
- [99] Khan, M. N. A., Chatwin, C. R., & Young, R. C. D. (2007). A framework for post-event timeline reconstruction using neural networks. *Digital Investigations*, 4(3-4), pp. 146-157.
- [100] Koen R., & Olivier M., (2008), The Use of File Timestamps in Digital Forensics, *In Proceeding of the Information Security of South Africa (ISSA 2008)*, pp. 1-16.
- [101] Kornblum J D. (2008). Using JPEG Quantization Tables to Identify Imagery Processed by Software, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS' 08)*, Vol. (5), pp. S21-S25.
- [102] Kornblum J D. (2006). Identifying Almost Identical Files Using Context Triggered Piecewise Hashing, *Digital Investigations, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, Vol. 3(Supplement 1), pp. S91-97.
- [103] Kwon H., Kim Y., Lee S. and Lim J. (2008)., A Tool for the Detection of Hidden Data in Microsoft Compound Document File Format, *In Proceedings of the International Conference on Information Science and Security, ICISS 2008*, ISBN:0-7695-3080-X
- [104] Lalis S, Karypidis A and Savidis A. (2005). Ad-hoc Composition in Wearable and Mobile Computing, *Communications of the ACM* Vol. 48(3), pp. 67-68.
- [105] Lamport, L (1978). Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of ACM*, Vol. 21(7): pp. 558-565
- [106] Lee S, Shamma D A and Gooch B. (2006). Detecting False Captioning Using Common Sense Reasoning, *Digital Investigations, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)* 3(Supplement 1), pp. S65-S70.
- [107] Leighland R and Krings A W. (2004). A Formalization of Digital Forensics, *Intl. Journal of Digital Evidence* Vol. 3(2), pp. 1-32.
- [108] Liebrock, L. M., Marrero, N., Burton, D. P., Prine, R., Cornelius, E., Shakamuri, M., et al. (2007). A Preliminary Design for Digital Forensics Analysis of Terabyte Size Data Sets. *Paper presented at the Symposium on Applied Computing (SAC '2007)*, Seoul.
- [109] Liu X., Zhang L., Li M., Zhang H., and Wang D., (2005)., Boosting Image Classification with LDA-based Feature Combination for Digital Photograph Management, *In Journal of Pattern Recognition, Elsevier Science Publications*, ISSN: 0031-3203, Vol. 38(6), pp. 887-901.
- [110] Lerman K., Plangprasopchok A., and Knoblock C. A. (2006)., Automatically Labeling Inputs and Outputs of Web Services, *In Proceedings of the National Conference on Artificial Intelligence*, (AAAI 2006), Menlo Park, California, pp. 1363-1368.
- [111] --- log2timeline.net, <http://log2timeline.net/>, last retrieved on July 12, 2011
- [112] Lyle, J. R. (2006). A Strategy for Testing Hardware Write block devices. *Paper presented at the*

6th Annual Digital Forensic Research Workshop (DFRWS '06) 3(Supplement 1), pp. S3-S9.

- [113] Maly K. J., Zeil S. J. and Zubair M. (2007)., Exploiting Dynamic Validation For Document Layout Classification During Metadata Extraction, *In Proceedings of the IADIS International Conference on World Wide Web and the Internet (WWW/Internet 2007)*, ISBN: 978-972-8924-44-7, pp. 261-268.
- [114] McKemmish R. (1999). What is Forensic Computing? *Australian Institute of Criminology: Trends and Issues in Crime and Justice*, ISBN 0-642-24102-3, No.188, pp.1-6.
- [115] Mead S. (2006). Unique File Identification in the National Software Reference Library, *Digital Investigations* Vol. 3(1), pp. 138-150.
- [116] Mee, V., Tryfonas, T., & Sutherland, I. (2006). The Windows Registry as a forensic artefact: Illustrating evidence collection for Internet usage. *Digital Investigation*, Vol. 3(3), pp. 166-173.
- [117] Mercuri R T. (2005). Challenges in Forensic Computing, *Communications of the ACM* Vol. 48(12), pp. 17-21
- [118] Metadata Working Group (2010), Guidelines for Handling Metadata, *Version 2.0*, http://www.metadataworkinggroup.org/pdf/mwg_guidance.pdf, last retrieved on July 12, 2011
- [119] Microsoft, File Properties Task, CodePlex Open Source Community, <http://fileproptiestask.codeplex.com/SourceControl/changeset/view/41095#417780>, last retrieved on July 12, 2011
- [120] Microsoft Developer Network Library, SYSTEMTIME Structure, *MSDN Microsoft Corporation*, [http://msdn.microsoft.com/en-us/library/ms724950\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms724950(v=VS.85).aspx) last retrieved July 12, 2011
- [121] Microsoft Developer Network Library, TIME_ZONE_INFORMATION Structure, *MSDN Microsoft Corporation*, [http://msdn.microsoft.com/en-us/library/ms725481\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms725481(v=VS.85).aspx), last retrieved July 12, 2011
- [122] Microsoft Developer Network Library, DYNAMIC_TIME_ZONE_INFORMATION Structure, *MSDN Microsoft Corporation*, [http://msdn.microsoft.com/en-us/library/ms724253\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms724253(v=VS.85).aspx), last retrieved on July 12, 2011
- [123] Microsoft Developer Network Library, File Times, *MSDN Microsoft Corporation*, [http://msdn.microsoft.com/en-us/library/ms724290\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms724290(v=VS.85).aspx), last retrieved on July 12, 2011
- [124] Microsoft Developer Network Library, Local Time, *MSDN Microsoft Corporation*, [http://msdn.microsoft.com/en-us/library/ms724493\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms724493(v=VS.85).aspx), last retrieved on July 12, 2011
- [125] Microsoft Developer Network Library, DateTime.ToUniversalTime Method, *MSDN Microsoft Corporation*, <http://msdn.microsoft.com/en-us/library/system.datetime.touniversaltime.aspx>, last retrieved July 12, 2011
- [126] Microsoft Support, Time stamps change when copying from NTFS to FAT, *Article ID 127830*, *Microsoft Corporation*, <http://support.microsoft.com/kb/127830>, last retrieved on July 12, 2011

- [127] Microsoft Support, Description of NTFS date and Time stamps for file and folders, *Article ID 299648, Microsoft Corporation*, <http://support.microsoft.com/kb/299648>, last retrieved on July 12, 2011
- [128] Microsoft Support, Interpreting timestamps on NTFS file systems, *Article ID 158558, Microsoft Corporation*, <http://support.microsoft.com/kb/158558>, last retrieved on July 12, 2011
- [129] Minack E., Paiu R., Costache S., Demartini G., Gaugaz J., Ioannou E., Chirita P-A, and Nejdil W., (2010), Leveraging personal metadata for Desktop Search: The Beagle ++ System, *Journal of Web Semantics: Science, Services, and Agents on the WWW, Elsevier Science Publications*, ISSN: 1570-8268, Vol. 8(1), pp. 37-54.
- [130] Mocas S. (2004). Building theoretical underpinnings for digital forensics research. *Digital Investigation*, Vol. 1(1), pp. 61-68.
- [131] Mohay G. M, Anderson A, Collie B, de Vel O and McKemmish R. (2003). Computer and Intrusion Forensics, *Artech House Publications*, ISBN 1580533698, 9781580533690
- [132] Murphey R. (2007). Automated Windows Event Log Forensics, *Digital Investigation, Paper presented at the 7th Annual Digital Forensic Research Workshop (DFRWS '07)*, Vol. 4(Supplement 1), pp. S92-S100.
- [133] Myers M and Rogers M. (2004). Computer Forensics: A need for Standardization and Certification, *Intl. Journal of Digital Evidence* Vol. 3(2), pp. 1-11.
- [134] National Institute of Justice (NIJ). Electronic Crime Scene Investigation Guide: A Guide for First Responders, *National Institute of Justice, Department of Justice (DoJ) 2001*. <http://www.ncjrs.gov/pdffiles1/nij/187736.pdf>
- [135] National Library of Australia, Thumbnail image Specifications, *Digital Collections*, <http://www.nla.gov.au/digicoll/pictures.html>, last retrieved on July 12, 2011
- [136] Netherlands Forensics Institute (NFI), Snorkel Java library, <http://www.holmes.nl/NFIlabs/Snorkel/index.html>, last retrieved on July 12, 2011
- [137] NISO. (2004). Understanding Metadata, *NISO Press*, <http://www.niso.org/publications/press/UnderstandingMetadata.pdf>, ISBN: 1-880124-62-9, pp. 1-20.
- [138] NIST. (May 2002). Hard Disk Hardware Write Block Tool Specification. *Unpublished manuscript*.
- [139] NIST. (May 2003). Hard Disk Software Write Block Tool Specification. *Unpublished manuscript*.
- [140] NIST. (Nov 2001). General Test Methodology for Computer Forensic Tools. *Unpublished manuscript*.
- [141] NIST. (Oct 2001). *Disk Imaging Tool Specification*. Unpublished manuscript.
- [142] Noakes A. (2008). Drew Noakes' Image Gallery, <http://drewnoakes.com/>, last retrieved on July 12,

- [143] Olievier M S. (2008). On metadata context in Database Forensics, *Digital Investigations*, Vol. 5(1), pp.1-8.
- [144] Olson J., & Boldt M., (2009). Computer forensic timeline visualization tool, *Digital Investigations, In proceedings of the 9th Annual Digital Forensic Research Workshop (DFRWS '09)*, Vol. 6 (2009), pp. S78-S87.
- [145] Pal A, Sencar H T and Memon N. (2008). Detecting File Fragmentation Point Using Sequential Hypothesis Testing, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, Vol. 5(Supplement 1), pp. S2-S13.
- [146] Pan L and Batten L M. (2005). Reproducibility of Digital Evidence in Forensic Investigations, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.
- [147] Park B, Park J and Lee S. (2009). Data Concealment and Detection in Microsoft Office 2007 Files, *Digital Investigation*, Vol. 5 (3-4). pp. 104-114.
- [148] Park J. and Lee S., Forensic investigation of Microsoft PowerPoint files. *Digital Investigation*, Vol. 6 (1-2) (2009), pp. 16-24.
- [149] Peisert S and Bishop M, (2007). How to Design Computer Security Experiments, *Proceedings of the Fifth World Conference on Information Security Education (WISE)*, pp. 141-148, West Point, NY, June 2007.
- [150] Petroni, J., Nick L., Walters, A., Fraser, T., & Arbaugh, W. A. (2006). FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory. *Digital Investigation*, Vol. 3(4), pp. 197-210.
- [151] Pollitt M M. (2007). An Ad-hoc review of Digital Forensic Models, *IEEE Publication, In Proceedings of the Second Intl. Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE '07)*.
- [152] Popescu A. C. and Farid H. (2004). Statistical Tools for Digital Forensics, in *Proc. Sixth Inf. Hiding Workshop*, May 2004.
- [153] Raghavan S., Clark A J., and Mohay G. (2009). FIA: An Open Forensic Integration Architecture for Composing Digital Evidence., *Forensics in Telecommunications, Information and Multimedia, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2009, Volume 8(1), pp. 83-94, DOI: 10.1007/978-3-642-02312-5_10
- [154] Raghavan S, & Raghavan S V., (2009). Digital Evidence Composition in Fraud Detection, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2010, Volume 31(1),1-8, DOI: 10.1007/978-3-642-11534-9_1
- [155] Reith M, Carr C and Gunsch G. (2002). An Examination of Digital Forensic Models, *Intl. Journal of Digital Evidence* Vol. 1(3), pp. 1-12.

- [156] Reyes, A., O'Shea, K., Steele, J., Hansen, J. R., Jean, B. R., & Ralph, T. (2007). Digital Forensics and Analyzing Data, Cyber Crime Investigations. In Burlington: Syngress, pp. 219-259.
- [157] Richard III, G. G., and Roussev, V. (2005). Scalpel: A Frugal High performance File Carver, *Paper presented at the 5th Annual Digital Forensics Research Workshop (DFRWS '05)*.
- [158] Richard III, G. G., and Roussev, V. (2006). Next-Generation Digital Forensics, *Communications of the ACM* Vol. 49(2), pp. 76-80.
- [159] Richard III, G. G. and Roussev, V.(2006). File System support for Digital Evidence Bags, *Proceedings of the Second Annual IFIP WG 11.9 International Conference on Digital Forensics, Advances in Digital Forensics II*, M. Olivier and S. Shenoj (Eds.), Springer, Boston, 2006. (ISBN 13: 9780-387-36890-6) pp. 29-40.
- [160] Richard III, G. G., Roussev, V., & Marziale, L. (2007). Forensic discovery auditing of digital evidence containers. *Digital Investigation*, Vol. 4(2), pp. 88-97.
- [161] Roesch M. (1999)., Snort – Lightweight Intrusion Detection for Networks, *In Proceedings of the 13th USENIX LISA Conference - System Administration Conference*, Seattle, Washington, pp. 229-238.
- [162] Ross, A., Time and Timestamps, <http://digfor.blogspot.com/2008/10/time-and-timestamps.html>, last retrieved on July 12, 2011
- [163] Rossev V, Chen Y, Bourg T and Richard III G G. (2005). md5Bloom: Forensic Filesystem Hashing Revisited, *Paper presented at the 5th Annual Digital Forensics Research Workshop (DFRWS '05)*.
- [164] Roussev, V., Richard III, G. G., & Marziale, L. (2007). Multi-resolution similarity hashing. *Digital Investigation*, Vol. 4(Supplement 1), pp. 105-113.
- [165] Rowe N. C. and Garfinkel S. (2011)., Finding anomalous and suspicious files from directory metadata on a large corpus, *to appear In Proceedings of the Third International Conference on Digital Forensics and Cyber Crime, ICDF2C 2011*, Dublin, Ireland 2011.
- [166] Sanderson, P. (2006). Identifying an existing file via KaZaA artefacts. *Digital Investigation*, Vol. 3(3), pp. 174-180.
- [167] Sarmoria C G and Chapin S J. (2005). Monitoring Access to Shared Memory Mapped Files, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.
- [168] Scientific Working Group on Digital Evidence, (2009). Technical Notes on Microsoft Windows Vista, *SWGDE Technical Notes*, February 8th 2009, pp. 1-25.
- [169] Scientific Working Group on Digital Evidence, (2010). Technical Notes on Microsoft Windows 7, *SWGDE Technical Notes*, May 15th 2010, pp. 1-20.
- [170] Schatz B L and Clark A J. (2006). An Open Architecture for Digital Evidence Integration, *Proceedings of the AusCERT R&D Stream, AusCERT 2006*, pp. 15-29.

- [171] Schatz, B., Mohay, G., & Clark, A. (2006). A correlation method for establishing provenance of timestamps in digital evidence. *Digital Investigation, The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, Vol. 3(Supplement 1), pp. 98-107.
- [172] Schuster, A. (2006). Searching for processes and threads in Microsoft Windows memory dumps. *Digital Investigation, The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, Vol. 3(Supplement 1), pp. 10-16.
- [173] Schuster, A. (2007). Introducing the Microsoft Vista event log file format. *Digital Investigation*, Vol. 4(Supplement 1), pp. 65-72.
- [174] The Sedona Conference Working Group, The Sedona Principles: Best Practices Recommendations & Principles for Addressing Electronic Document Production (Second Edition) (2007)., http://www.thesedonaconference.org/content/miscFiles/TSC_PRINCP_2nd_ed_607.pdf, last retrieved on July 12, 2011
- [175] The Sedona Conference Working Group, The Sedona Conference Glossary: E-Discovery & Digital Information Management (Third Edition) (2010)., www.thesedonaconference.org/dltForm?did=glossary2010.pdf, last retrieved on July 12, 2011
- [176] The Sedona Conference Working Group, The Sedona Conference: Commentary on ESI Evidence & Admissibility (2008)., http://www.thesedonaconference.org/dltForm?did=ESI_Commentary_0308.pdf, last retrieved on July 12, 2011
- [177] Sencar H. T. and Memon N. (2009)., Identification and Recovery of JPEG Files with Missing Fragments, *Digital Investigation*, Vol. 6(4), pp. S88-98, 2009.
- [178] Sencar H. T. and Memon N. (2008)., Overview of State-of-the-art in Digital Image Forensics, *Part of Indian Statistical Institute Platinum Jubilee Monograph series titled Statistical Science and Interdisciplinary Research*, WORLD SCIENTIFIC PRESS, 2008.
- [179] Shankaranarayanan G and Even A. (2006). The Metadata Enigma, *Communications of the ACM* Vol. 49(2), pp. 88-94.
- [180] Standards Australia. (2003). *HB171 - Guidelines for the management of IT Evidence*.
- [181] State Library of New South Wales, Guidelines for digitizing images in NSW public libraries, http://www.sl.nsw.gov.au/services/public_libraries/docs/digital.pdf, last retrieved on July 12, 2011
- [182] Strauss A and Corbin J, Basics of Qualitative Research Techniques and Procedures for Developing Grounded Theory (1st edition), Sage Publications: London (1990)
- [183] Steele, J. (2007). Digital Forensics and Analyzing Data: Alternate Data Storage Forensics. In (pp. 1-38). Burlington: Syngress.
- [184] Stevens M W. (2004). Unification of relative Time Frames for Digital Forensics, *Digital Investigations*, Vol. 1(1), pp. 225-239.

- [185] TCPDUMP, Command-line packet analyzer, <http://www.tcpdump.org/>, last retrieved on July 12, 2011
- [186] Tichy W. F., (1998). "Should computer scientists experiment more?", *Computer* Vol. 31(5), pp. 32–40, May 1998
- [187] --- Timeline analysis part 3, log2timeline, <http://thedigitalstandard.blogspot.com/2010/03/timeline-analysis-part-3-log2timeline.html>, last retrieved on July 12, 2011
- [188] Thomas D. R., (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data, *American Journal of Evaluation*, Vol. 27(2) pp. 237-246.
- [189] Toyama K., Logan R., Roseway A. and Anadan P. (2003)., Geographic Location Tags on Digital Images, *In Proceedings of ACM Multimedia 2003, Berkeley, California*, ISBN: 1-58113-722-2, pp. 156-166.
- [190] Turner, P. (2005). Unification of digital evidence from disparate sources (Digital Evidence Bags). *Digital Investigation*, Vol. 2(3), pp. 223-228.
- [191] Turner, P. (2005). Digital provenance - interpretation, verification and corroboration. *Digital Investigation*, Vol. 2(1), pp. 45-49.
- [192] Turner, P. (2006). Selective and intelligent imaging using digital evidence bags. *Digital Investigation, The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, Vol. 3(Supplement 1), pp. 59-64.
- [193] van Baar R B, Alink W and Van Ballegooji A R. (2008). Forensic Memory Analysis: Files Mapped in Memory, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, Vol. 5(Supplement 1), pp. S52-S57.
- [194] Venter, J., de Waal, A., & Willers, C. (2007, Jan). *Specializing CRISP-DM for Evidence Mining*. Paper presented at the Digital forensics; Advances in digital forensics III: IFIP International Conference on Digital Forensics, Orlando, FL.
- [195] Volatility – Volatile memory artifact extraction utility framework, Volatile Systems, <https://www.volatilesystems.com/default/volatility>, last retrieved on July 12, 2011
- [196] Wang, S.-J., & Kao, D.-Y. (2007). Internet forensics on the basis of evidence gathering with Peep attacks. *Computer Standards & Interfaces*, Vol. 29(4), pp. 423-429.
- [197] Wang W and Daniels T E. (2005). Network Forensic Analysis with Evidence Graphs, *Paper presented at the 5th Annual Digital Forensic Research Workshop (DFRWS '05)*.
- [198] W3Schools.Com, Operating System Platform Statistics, http://www.w3schools.com/browsers/browsers_os.asp, last retrieved on July 12, 2011
- [198a] Webster's English Dictionary, <http://www.webster-dictionary.org/>, last accessed July 12, 2011
- [199] Weil M C., (2002), Dynamic Time and Date stamp analysis, *Intl Journal of Digital Evidence*, Vol.

1(2), pp. 1-6.

- [200] Willassen S. (2008). Finding Evidence of Antedating in Digital Investigations, *In Proceedings of the Third International Conference on Availability, Reliability and Security*, ARES 2008, pp. 26-32.
- [201] Windows Hardware Development Center, *Microsoft Extensible Firmware Initiative FAT32 File System Specification*, rev. 1.03, Dec 6, 2000, Microsoft Corporation, <http://msdn.microsoft.com/en-us/windows/hardware/gg463080>, last retrieved on July 12, 2011
- [202] Zander S., Nguyen T. and Armitage G. (2005)., Self-Learning IP Traffic Classification Based on Statistical Flow Characteristics, *In Proceedings of the Sixth ICST Conference on Passive Active Measurement*, (PAM 2005), LNCS 3431, Springer-Verlag Publishers, Berlin, pp. 325-328.