AssocGEN: Engine for Analyzing Metadata Based Associations in Digital Evidence

Sriram Raghavan
Secure Cyber Space (*www.securecyberspace.org*)
Email: sriram.raghavan@securecyberspace.org

S V Raghavan
Department of Computer Science & Engineering,
IIT Madras, Chennai INDIA
Email: svr@cs.iitm.ernet.in

Abstract— Traditionally, sources of digital evidence are analyzed by individually examining the various artifacts contained therein and using the artifact metadata to validate authenticity and sequence them. However, when artifacts from forensic images, folders, log files, and network packet dumps have to be analyzed, the examination of the artifacts and the metadata in isolation presents a significant challenge. Ideally, when a source is examined, it is a valuable task to determine correlations between the artifacts and group the related artifacts. Such a grouping can simplify the task of analysis by minimizing the need for human intervention. By virtue of the value that metadata bring to an investigation and its ubiquitous nature, metadata based associations is the first step in realizing such correlations automatically during analysis.

In this paper, we present the AssocGEN analysis engine which uses the metadata to determine associations between artifacts that belong to files, logs and network packet dumps, and identifies metadata associations to group the related artifacts. A metadata association can represent any type of value match[1] or relationship that is deemed relevant in the context of an investigation. We have conducted preliminary evaluation of AssocGEN on the classical ownership problem to highlight the benefits of incorporating this approach in existing forensic tools.

Keywords— Metadata association, similarity pocket, similarity group, association group

I. INTRODUCTION

During the analysis of digital evidence, there is a need to ascertain the nature of the artifacts contained and how they corroborate with each other in relevance to an investigation. However, the establishment of such associations and the discovery of relationships continue to remain largely manual. As the heterogeneity of digital evidence continues to grow with advances in technology, we are faced with newer digital devices, newer file and log formats from which such associations must be discovered. While keyword search continues to remain the dominant technique in the discovery of related artifacts from which some associations can be discerned, there are two challenges to this approach:

1. *keywords used for search are not extensive to cover all aspects of associations between the artifacts in the digital evidence*
2. *unless the keyword is known a priori, the grouping cannot be established, even when the associations already exist in evidence*

Therefore, there is a need for a more comprehensive and automated method to identify the associations and group the related artifacts without human involvement.

Current forensic tools like Encase, FTK and Sleuthkit and analysis tools like PyFlag, Wireshark, Volatility and log2timeline provide the functionality for keyword search in addition to classification and filtering which allow the artifacts to be grouped in a particular way that is then examined for patterns. Often, an examiner may need to classify the digital evidence repeatedly in different ways before a pattern becomes apparent. The challenge with this approach is that the sequencing of the classifications and the attributes may be crucial to identifying the relationships. However, in the absence of complete knowledge about the digital evidence, arriving at the optimal set of classifications can be a challenge. In lieu of having to provide the keywords for searching and the attributes according to which the digital evidence must be classified to discern all patterns, it is necessary to utilize the attributes of the artifacts, already present in the evidence. Besides, in order to be exhaustive so as not to miss any potentially valuable association or relationship, an automated method is needed.

Metadata in digital evidence store information regarding the creation, usage and context pertaining to the artifacts they are associated with, which is valuable to forensic analysis [3]. Metadata can be treated as a vector of values pertaining to the artifact. Extending the principle of a keyword search, we use the metadata and group the resulting artifacts for each value. Each group describes a specific metadata association. Then all the groups containing overlapping artifacts are consolidated into a larger group. Sets of such groups hold non-overlapping sets of digital artifacts which can be used to prime the search for relevant evidence. The rest of this paper is organized as follows: In Section 2, we present the analysis engine architecture and describe its implementation. In Section 3, we illustrate its benefit using a classical ownership resolution problem and present preliminary results of the performance study comparing completion times against FTK 3.2. In Section 4, we conclude with a brief summary and provide scope for future work.

[1] AssocGEN uses a value match for identifying a metadata association

Sponsored by IEEE Louisville Chapter.
978-1-4799-4061-5/13/$31.00 ©2013 IEEE

II. THE ASSOCGEN ANALYSIS ENGINE

The AssocGEN is our research prototype[2] implementation of a metadata based approach introduced in FIA [11] to integrate different sources of digital evidence and unify the analysis by identifying metadata matches between them. The AssocGEN architecture is shown in Fig 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.

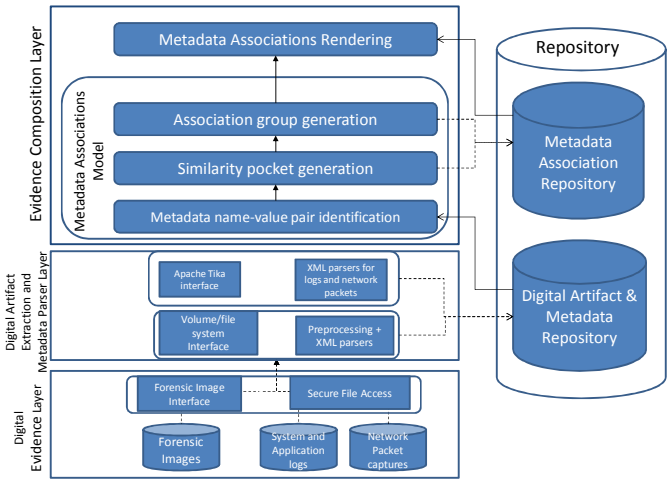

Fig 1. The AssocGEN architecture

A. Design Rationale

AssocGEN was primarily designed with the view of abstracting current technological support extended to heterogeneous sources of digital evidence. The architecture is inspired by the hypothesis-based review of forensic and analysis tools [12] and incorporates 3 basic layers of abstraction, viz., the digital evidence layer, the digital artifact traversal and metadata parser layer and the evidence composition layer. The Digital evidence layer provides binary-stream support 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* [10] are potential candidates to provide this support.

Among these, *Sleuthkit* 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 challenge. The *ewflib* has similar concerns, best taken advantage of using commercial forensic toolkits like Encase which encapsulate using proprietary binary stream interfaces. The *afflib* accessed a source of digital evidence as inodes which store the attributed related to the contents. The abstractions modeled in the *afflib* library were more conducive to raw binary data and stream processing rather than the discrete digital artifact abstraction which is the focus of our work. *Afflib* was therefore, restrictive in terms of being able to define a generic metadata structure to determining metadata matches. The *Snorkel* library, on the other hand, provided the necessary abstractions to handle the digital artifacts are read-only nodes with metadata. Besides, developed in Java, can readily integrate with other Java libraries for log parsing and network packet analysis which can be automated. Hence, *Snorkel* was chosen to implement the 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*, *fiwalk* and the *apache tika* [2] libraries. Of these, the *libextractor* was completely built in C and had (at that time) limited file metadata support to word processing documents. Besides, the memory requirements to handle the 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, and their 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 our 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, *apache tika* was chosen to implement the metadata parsers in AssocGEN.

To process the log records and network packets individually, we processed logs and network traces and translated 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 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. We 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.

B. Digital Evidence Layer

The Digital Evidence layer was built using the snorkel library[3] which is responsible for providing raw binary access through a forensic file system interface. The snorkel library mirrors the functionality of the *fiwalk* tool [9]. 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 provided regulated bit-stream access to the various different digital evidence sources from the upper layers. The layer allowed unidirectional data flow ensuring read-only access to forensic images, file systems, Internet browser logs and network packet

[2] Source can be downloaded from http://sourceforge.net/projects/assocgen/

[3] Netherlands Forensics Institute (NFI), Snorkel Java library, http://www.holmes.nl/NFIlabs/Snorkel/index.html

captures implemented by the snorkel forensic image interface. The snorkel interface allowed traversing multiple forensic images without compromising data integrity.

C. Digital Artifact Traversal and Metadata Parser Layer

The digital artifact extraction and metadata parser layer was composed of third party applications that we designed to traverse the digital artifacts and parse the metadata. This layer was implemented using the Apache tika metadata extractor library[4] to parse metadata from files and log analyzers to traverse log records and network packets and parse their attributes. We extracted the metadata from files based on the file MIME type. The MIME type for a file was identified by determining its encoding type in conjunction with its magic numbers identifying the file beginnings and endings.

The browser logs were initially processed by a third party application (Nirsoft browser analyzer[5]) into XML which was then read by our parsers to extract the attributes for individual browser events. The browser history was 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 were regarded as its metadata. In our prototype, we have developed parsers for history and cache logs for the Internet Explorer and Mozilla Firefox browser applications. The network packet captures were similarly processed into XML using Wireshark and then interpreted by our parser to extract packet related attributes. These lists of hash tables extracted from all the different digital artifacts across all the sources of digital evidence were stored into the repository.

D. Evidence Composition Layer

The Evidence composition layer comprised of algorithms that seek metadata matches between the various digital artifacts and group them. These groupings are merged and presented to an examiner for analysis. The AssocGEN was configured to prioritize based on metadata matches determining the *source*, *ownership* and *timestamps* of digital artifacts; for instance, all digital images captured using Canon Powershot A70 by the user on Sept 11 2011.

Between two or more digital artifacts, a single metadata match led to a set of digital artifacts that have an identical value for that metadata tag name. Such a set was termed a *similarity pocket*. It was identified by the metadata tag name. 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 were merged into a *similarity group*. The number of digital artifacts that overlap between two or more similarity pockets can be either partial or complete. In the case where it is complete, each component similarity pocket will contain the same set of artifacts for different metadata name. When such similarity groups match across multiple sources, these were merged into an *association group*. Merging the overlapping similarity pockets continues until all transitive overlaps are accounted for. When multiple similarity pockets were merged into a similarity group and multiple similarity groups into an association group, the individual similarity pockets and similarity groups in the repository were 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, metadata tags take names of individuals, metadata tags that take the values of applications, metadata tags that take timestamps and so on. Such equivalence relations were 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. The naïve algorithm to determine metadata matches across a set of N sources of digital evidence is described in Algorithm 1.

association grouping algorithm

GIVEN: S: set of all sources of digital evidence $\{S_1, S_2, S_3, \ldots, S_N\}$
A: set of all digital artifacts across the set S $\{a^i_j \mid a^i_j$ is an artifact of source $S_i\}$
M: set of all metadata across all digital artifacts in set A $\{m^i_j \mid m^i_j$ is an artifact of source $a^i_j\}$

To do:
For each S_i in set S, do
For all digital artifacts a^i_j in source S_i, do
$SP \leftarrow$ list of all similarity pockets sp^i_t for all metadata m^i_j in M without repetition
$SG_i \leftarrow$ Set of all $sg^i_t = \{\bigcup_t sp^i_t\}$ where there is at least one digital artifact in common without repetition
End *for*
End *for*
$AG \leftarrow$ Set of all $ag_t = \{\bigcup_t sg_t\}$ where at least one metadata tag is equivalent $\forall(S_i, S_j)$ where $i \neq j$
End algorithm

Algorithm 1. Association grouping algorithm

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

1) Metadata Equivalence in AssocGEN

[4] Apache Software Foundation, Apache Tika – content analysis toolkit, http://tika.apache.org/

[5] http://www.nirsoft.net/web_browser_tools.html

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

1. *Between 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; and*
3. *IP addresses and domain names from DNS lookups in browser logs and network packet captures*
4. *Filesize 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*
7. *'Source' in packet captures with 'Domain' in browser logs*

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

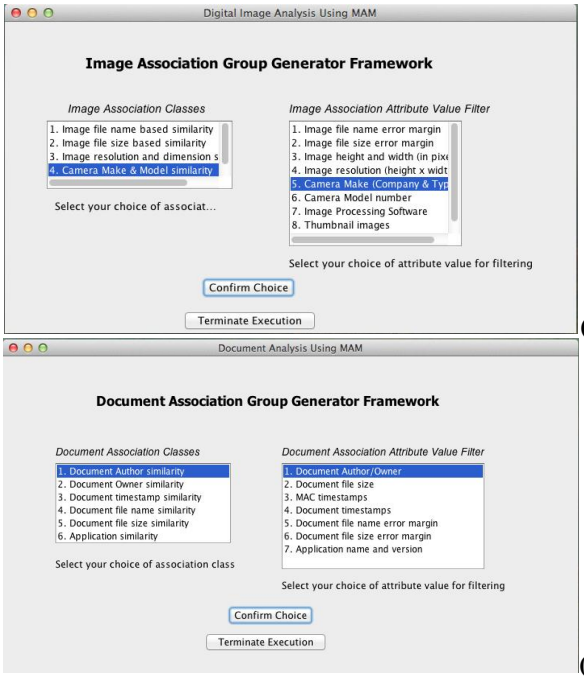

Fig 2. The AssocGEN customized to analyze files from file systems

2) Configuring and Controlling Metadata Associations Using AssocGEN

Typically, AssocGEN extracted all metadata from each digital artifact and groups artifacts according to the inherent metadata matches. As this can be an exhaustive approach with significant computation complexity, an alternate AssocGEN also allowed a user to specify a subset of metadata from the digital artifacts, often based on their application type, in order to contain the number of metadata matches found and hence contain 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 shown in Fig 2.

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 such digital images that are associated rather than containing images captured with the same digital camera.

In lieu of an exhaustive approach described in Algorithm 1, the AssocGEN engine also allows a user to select an artifact at random and generates a list of all artifacts on the same source or across all sources (as configured) by determining its associated artifacts. This method uses a user chosen digital artifact as a seed and determined all other artifacts that are metadata associated with that artifact. This process is then repeated in turn for each digital artifact in the identified set until transitive closure is achieved. Each digital artifact is only listed once. Such an iterative approach adopts an incremental search method and the algorithm to determine such associated artifacts is given in Algorithm 2.

incremental association builder algorithm

GIVEN: S: set of all sources of digital evidence $\{S_1, S_2, S_3, \ldots, S_N\}$
A: set of all digital artifacts across the set S $\{a^i_j \mid a^i_j$ is an artifact of source $S_i\}$
M: set of all metadata across all digital artifacts in set A $\{m^i_j \mid m^i_j$ is an artifact of source $a^i_j\}$
SEED: digital artifact a^i_j on some source S_i with metadata m^i_j

To do:

loop 1: For all $m^i_j \in a^i_j$, do
$sp^i_t \leftarrow$ all a^i_j for single metadata matches in m^i_j
End *for*
$sp^i \leftarrow \{\bigcup_t sp^i_t\}$ across all metadata in m^i_j without repetition
For each a^i_j in sp^i
repeat *loop 1* across all sources S_i in S
$SG_t \leftarrow \{\bigcup_i sp^i\}$ without repetition
End *for*
$AG \leftarrow \{\bigcup_t SG_t\}$
End algorithm

Algorithm 2. Incremental association builder algorithm

3) Comparing AssocGEN against Contemporary Forensic Toolkits and Architectures

The AssocGEN implementation supports forensic disk images, Internet browser logs on three browsers and PCAP packet capture files. We present a comparison of our AssocGEN implementation against contemporary forensic toolkits and architectures in Table I. Our primary motivation was the identification of metadata matches across digital artifacts and therefore the tool only deals with well-defined items such as files, log records and network packets. In order to distinguish Internet browser logs from other files on a file system, we separate them as logs which are analyzed to corroborate a user's Internet activity against any resource exchanged or downloaded to the user's file system during the same.

TABLE I. TABULATING THE COMPARISONS BETWEEN ASSOCGEN AND CONTEMPORARY FORENSIC TOOLKITS

	Digital Evidence access	**Digital Artifact Traversal**				**Metadata Parsing & Extraction**	**Evidence Composition**	
	Binary abstraction to digital evidence	*File system examination*	*Log examination*	*Network capture examination*	*Text indexing and Search*		*Multiple sources of digital evidence (examination and analysis)*	*Identify correlations*
Encase	√	√	×	×	√	Only file system metadata	Only file system images (examination)	×
FTK	√	√	×	×	√	Only file system metadata	Only file system images (examination)	×
Sleuthkit	√	√	×	×	√	Only file system metadata	Only file system images (examination)	×
PyFlag	√	√	√	√	√	Only file system metadata	Only examination	×
OCFA	√	√	×	×	√	Only file system metadata	Only file system images (examination)	×
AssocGEN	√	√	Only Internet browser logs (on Internet Explorer, Firefox and Safari)	Only PCAP	√	File system and application metadata in files and browser log and packet capture attributes	Traversal and search on forensic disk images, Internet browser logs and PCAP packet captures	Identification of metadata matches + grouping of *related* digital artifacts

On files, we parse both the file system and application metadata and determine metadata matches where a one-to-one correspondence on metadata can be established readily. This is usually feasible for files belonging to the same file format such as Microsoft Word 97, Microsoft Word 2003, JPEG, and TXT. Where the file format types differ, we determine similarities by identifying those metadata that are likely to take the same type of value for comparison.

On Internet browser logs, we parse the log record attributes that pertain to the nature of web page visit, the domain server, the page attributes, the resources accessed and their attributes and so on. On network packet captures, attributes pertaining to the packet such as the source and destination IP address, the OS on the source, the protocol, the packet timestamp, the packet size, and attributes corresponding to resources exchanged. Across source types, we identify those metadata that are likely to take similar type of value and establish metadata equivalence between them by configuring AssocGEN.

III. EVALUATION

Buchholz and Spafford [3] have analyzed the significance of file system metadata and argued that when sufficient metadata is recorded, it can aid in determining answers to the 6 questions. The AssocGEN extends this approach by determining metadata associations across artifacts to discover artifact relationships for analysis and evidence corroboration.

A. File Ownership Problem

Consider the problem where the owner of a document is under investigation. We have adapted this problem to illustrate the benefits of using AssocGEN to examine multiple sources of digital evidence 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 posed to an examiner is who is responsible for file F? We interpret this question as who is the author of the contents found in file F? We illustrate this file ownership problem in Fig 3. From the description of the problem, we 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.

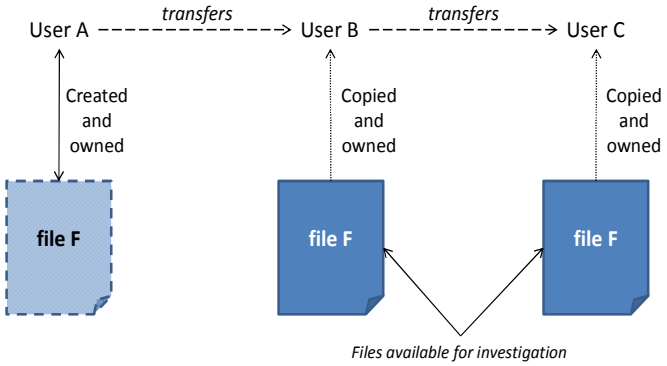

Fig 3. Illustrating the file ownership problem

1) 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 a standard forensic tool. Buchholz and Spafford observed that in the presence of *only* file system metadata, and particularly, the owner of the document, *this problem cannot be solved*. From a digital forensics standpoint, they advocate that if an examiner is uncertain as to who (or which process) is responsible for an output, 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 should have been seized from User B and User C, but not User A. Despite the ability 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.

Using AssocGEN, the sources acquired from User B and User C can be examined together allowing an examiner to corroborate the sources of digital evidence. This will also allow the extraction of the document metadata in addition to the file system metadata from the two copies of file F. The document metadata from the two copies of file F, from User B and User C, will generate metadata matches to form association groups that include the metadata 'Author' and 'filesize'. An simple examination of this association group *identifies* User A, who remained undiscovered previously. Once a copy of User A's file F is also examined, the metadata matches generated with this file and a timeline to establish the chronology of file events across the three copies of file F will establish User A as the owner.

2) Automatic Corroboration of Evidence Using AssocGEN

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 groups them into similarity pockets. 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 user*

Once the groupings are presented, the user 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 user examines the groupings, one will find 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. Application metadata in tandem with file system metadata will also generate multiple matches that correspond to information that correspond to who, when, where and how, leading to association groups that characterize the similarity of the two copies of file F. The involvement of User A can be fully established if an evidence source is seized from User A and the contents of file F is compared against the other two copies of the file from B and C. Moreover, the timestamps recorded on the application metadata, which will predate the MAC timestamps discovered on copies of file F with B and C, will establish User A as the original author of file F.

B. Performance

Table II shows the results of performance studies that we conducted using AssocGEN on different datasets. The datasets contained files taken largely from file systems, albeit containing temporary internet files such as script files, HTML, XML files in addition to digital image files and word processing documents. Dataset #1 and #5 contained high-resolution digital photographs and each image file > 1.3 MB. Dataset #2 was primarily photographs of indoor scenes and the images ranged in file size from 200 KB to 2.3 MB. Dataset #3 contained digital photographs, edited photographs, digital generated composites, downloaded image files and thumbnail images and belonged to an individual personal collection. Dataset #4 was largely edited image files and thumbnail image files downloaded in response to Google search queries. In this dataset, very little application metadata were available. Dataset #6 contained a mixture of image files, documents and text and Unicode files which were also downloaded in response to Google queries.

TABLE II. TABULATING THE RESULTS OF PERFORMANCE STUDY CONDUCTED ACROSS DIFFERENT DATASETS

				Time taken to traverse the dataset	
Serial No.	**Dataset Volume**	**Nature of files**	**Number of files in the dataset**	**using AssocGEN**	**using FTK 3.2**
1	126 MB	Carved raw photographs	52	51 secs	1 min 26 secs
2	374 MB	Digital photographs	126	1 min 42 secs	2 min 30 secs
3	1.6 GB	Assorted image files	491	2 min 24 secs	3 min 20 secs
4	6.8 GB	Downloaded image files	2157	23 min 28 secs	44 min 20 secs
5	49.3 GB	Hi-resolution Digital photographs	16386	2 hrs 16 min	3 hrs 38 mins
6	33.2 GB	Downloaded files (assorted)	29700	3 hrs 24 min	6 hrs 12 mins

During each traversal, the metadata for each file traversed was parsed and printed on the console. The times reported are averaged over 10 runs of the traversal on each dataset. Since FTK, by design, traverses the BLOB (binary large object, i.e., forensic image) first before identifying the metadata for each digital artifact and extracting them, it was always slower than AssocGEN which only seeks the logical self-contained artifacts. Besides, FTK only seeks file system metadata, while AssocGEN extracts both file system metadata and the application metadata. Interestingly, in dataset #1, our tool was only able to extract metadata from 34 complete image files, while FTK was able to extract at least partial file system metadata from all the 52 images files; however, it took more time. In datasets #2, #3, #4 and #5, AssocGEN identified all edited and digital generated image files and grouped the edited images with its corresponding digital photograph where available.

IV. CONCLUSIONS & FUTURE WORK

In this paper, we presented the design of a forensics and analysis tool that extracts metadata from files, log records and network packets and identifies metadata associations to group them. The tool uses metadata to access and analyze multiple artifacts across one or more sources of digital evidence together. The grouping process also does not require constant user monitoring or input unlike current tools. We illustrated the features of this tool in comparison with existing forensic tools and showed how metadata can be used to corroborate information across sources in order to solve the classical file ownership problem. This demonstrated the use of metadata associations can attribute the ownership to the correct individual which is an important aspect of digital forensic analysis. We also discussed some preliminary results on performance across datasets containing digital image files and word processing documents.

In the future, we hope to apply this tool to the analysis of usage discovery across multiple file systems, log files and network packet traces in a corporate setting to discern data exfiltration and IP theft. Evidence corroboration as discussed in Section II. D can be exercised for email transactions and system logs, and we are presently updating AssocGEN to implement them. The updated version is expected to support Windows registry, syslog on UNIX, Outlook, Thunderbird and iMail clients.

REFERENCES

[1] 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, Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06), 3*(Supplement 1), pp. 50-58.

[2] Apache Software Foundation, Apache Tika – content analysis toolkit, http://tika.apache.org/, last retrieved on July 12, 2011

[3] Buchholz F and Spafford E H. (2004). On the Role of System metadata in Digital Forensics, *Digital Investigations,* 1(1), pp. 298-309.

[4] 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.

[5] Carrier B D., (2003), Sleuthkit, http://www.sleuthkit.org/sleuthkit/, last retrieved on July 12, 2011

[6] 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).*

[7] 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.

[8] 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.

[9] Garfinkel S., (2009), Automating Disk Forensic Processing with Sleuthkit, XML and Python, *In Proceedings of the 2009 Fourth International IEEE Workshop on Systemmatic Approaches to Digital Forensic Engineering* (SADFE 2009), Berkeley, California, ISBN: 978-0-7695-3792-4, pp. 73-84.

[10] Netherlands Forensics Institute (NFI), Snorkel Java library, http://www.holmes.nl/NFIlabs/Snorkel/index.html, last retrieved on July 12, 2011

[11] 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

[12] Raghavan S. and Raghavan S. V. (2013). A Study of Forensic and Analysis Tools*, In Proceedings of the 2013 8th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE)***,** IEEE 978-1-4799-4061-5, Hong Kong, China*,* Nov 21-22, 2013.