An Automated Link Analysis Solution Applied to Digital Forensic Investigations

Fergal Brennan(✉), Martins Udris, and Pavel Gladyshev

University College Dublin, Dublin, Ireland
{fergalbrennan, martins.udris}@gmail.com,
pavel.gladyshev@ucd.ie

Abstract. The rapid growth of computer storage, new technologies, anti-forensics and hacking tools, as well as cheaper and easily accessible powerful computing equipment, has led to digital crimes becoming more frequent and often more sophisticated. These challenges have led to digital examinations becoming increasingly time-consuming and laborious, resulting in an urgent need for the automation of digital forensic analysis. In addition to in-depth analysis of particular digital devices, it is often necessary to establish that two devices and hence their owners are linked. This need arises, for example, when a suspect is apprehended and the investigator needs to establish grounds for the detention of a suspect. This paper proposes a methodology and a software solution to automate the detection of information linkage between two or more distinct digital devices.

Keywords: Forensic tools · Link analysis · Social network analysis · Software engineering · Automation · Profiling · Visualisation · Keywords

1 Introduction

Technological advances have led to an increasingly networked society, with criminal and terrorist networks thus becoming more sophisticated and increasingly difficult to analyse. Social network analysis tools can provide a quantitative or qualitative understanding of the performance of networks and their ability to meet their goals, to identify networks characteristics and key individuals, as well as establish how quickly information flows within networks.

Jacob L. Moreno is widely credited for being the founder of what we now know as social network analysis [1]. Moreno's 1934 publication *Who Shall Survive?* [2] developed a methodology known as Sociometry, which measures social relationships between individuals and the impact of these relationships on small groups. Although there were various articles published prior to Moreno's work, such as Almack [3] and Bott [4], which explored network structures, it was Moreno who first began to explore the use of graphs as a visualisation tool to depict social networks [5]. Wassermann and Stanley's excellent publication Social Network Analysis: Methods and Applications [1] provide an excellent overview of the subject as well as an in-depth exploration of social network analysis.

P. Gladyshev et al. (Eds.): ICDF2C 2013, LNICST 132, pp. 189–206, 2014.
DOI: 10.1007/978-3-319-14289-0_13

The time-consuming nature of manual social network analysis had limited applicability to criminal or terrorist investigations [6]. Due to the benefits of social network analysis, the need for the automation of this process to address the challenges faced by investigators was urgently required [7]. This led to a number of tools being developed that were capable of interrogating large data sets and automatically producing graphical representations of the entities and relationships within a network. These tools included functionality for filtering based on entity type and they employed the spring embedder algorithm [8]. The most popular of these tools were Netmap [9] and early versions of both COPLINK [10] and Analyst's Notebook [11].

This initial wave of new analysis tools was a breakthrough in the field of automated social network analysis. It led to the development of advanced analytical functionality that can determine important network characteristics in tools such as Analyst's Notebook, COPLINK and the recently released Maltego [12] which leverages Open Source Intelligence [13] to discover relationships between entities based on public information. These tools provide the investigator with the ability to determine principles such as centrality, betweenness, closeness, patterns of interaction and the ability to identify individuals of most importance in a social network [1]. However, these tools rely on structured relational data already in place within an organisation or data that is publicly available. Therefore, to prove or disprove the existence of a relationship between various individuals potentially involved in a crime remains a time consuming and challenging task.

Traditional forensic tools like EnCase [14] or XWays [15] are designed to allow investigators to manually traverse the file structure of a forensic image in order to discover relevant digital evidence. Additionally, certain forensic artefacts require bespoke extraction and presentation using a variety of tailored forensic tools. Performing analysis in this fashion particularly in a multiparty case where each artefact repository would ordinarily have to be examined independently, the findings then manually correlated requires significant manual effort. This may lead to crucial data links connecting certain parties being overlooked.

The possibility of automating the discovery of relational data among forensic artefacts, on the other hand, may lead investigators to crucial evidence – for example, by searching the computers of a criminal and their suspected accomplice for common data items, such as email addresses, documents, etc., the investigator can quickly and automatically determine that the two persons know each other, which may be sufficient grounds for the detention of the suspect for further questioning. Automated discovery of common data items, such as a specific email addresses, keywords, or internet history entries on multiple devices may allow the investigators to identify criminal rings and focus their investigation on the individuals whose devices have the most relational data linking them to other devices in the case.

To the authors' knowledge there are currently no publicly available open source forensic tools that would provide such functionality. This paper summarises the results and the lessons learned from a project aimed at the development of a prototype of such a forensic tool.

2 Solution Overview

The solution provides investigators with a platform to compare data from multiple data sources for purposes of comparison and presentation with the goal of revealing any relationships that exist between the digital forensic evidence sets. To facilitate this fundamental objective, the solution provides a number of libraries that enable the acquisition, normalisation, comparison and presentation of forensic artefact data from the Windows and Linux platforms. While the solution provides a considerable amount of functionality, it also facilitates an extensible platform with which investigators can develop libraries to meet their needs by logically separating the core functions of the solution into three tasks, as described below and presented in Fig. 1.

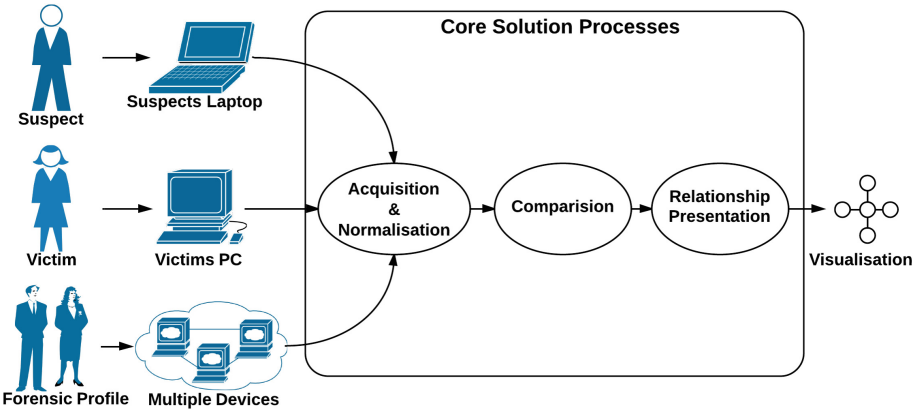

Fig. 1. External user interaction and presentation of solutions core processes separated by each processes' singular functional concern.

1. The acquisition and normalisation of forensic data.
2. The comparison of forensic data.
3. The presentation of forensic data.

If a case involves a device that the solution does not currently support the acquisition of data from, but the investigator wishes to compare its data to the rest of their case data, they can write their own library which can be integrated into the solution for use.

Due to the increasing number and complexity of cases investigators are required to handle, a logically structured intuitive user interface is available where investigators can organise their current and historical case data. A number of search and sorting functions are available to allow easy access to relevant case data and real time statistics. The solution's interface follows a logical digital forensic case paradigm whereby data sources and evidence can be attributed to individuals within a case.

This structure also enables the creation of digital profiles which allows investigators to logically group acquired forensic artefacts. For example, if an investigative department seizes disk images from a number of convicted paedophiles, the investigator can generate a profile based on the entirety or subsections of these disk images. The investigator can then run the data associated with a single individual or multiple individuals against a digital profile which will highlight any relationships that exists between them and the created paedophile profile, as well as potentially identifying new leads or individuals of interest in the case. Given enough processing power, a police department could build up a profile repository similar to that of a finger-print/DNA database currently in use throughout the world.

The user interface allows the investigator to execute any of the available libraries. In the event of long running tasks, the solution's interface provides an acceptable level of responsiveness and up to date information regarding the status of those tasks. On completion of appropriate libraries to acquire and compare the forensic artefacts associated with a case the resulting data is presented in a graphical visualisation.

3 Technology Decisions

All technologies used in the development process are Open Source and freely available, allowing the solution to be distributed without the need for licensing. Python formed the core programming language used throughout the development process. It was chosen because it is a freely available, interpreted and fully object-oriented language, which will allow for continued modular development of the solution. Python has a simple and clear syntax which increases development speed, supports multithreading and multiprocessing, has excellent memory management and supports exception handling. PostgreSQL is the database engine provider that is utilized to persist and store normalised forensic data. It is a highly scalable object-relational Open Source database and runs on all major operating systems. PyQt was chosen for the first development phase of the user interface. PyQt is a set of Python wrappers for Nokia's Qt application framework, it provides a rich graphical widget toolkit, has extensive drawing capabilities and runs on all major operating systems.

The solution architecture, as presented below in Fig. 2, is distributed among a number of logical tiers to promote ease of continued development and testing.

4 Functional Design Implementation

Six core features were identified to create an intuitive environment to allow digital forensic artefact correlation. These are: 1. Create an investigative case, 2. Create a digital profile, 3. Acquire information sources, 4. Extract forensic artefacts, 5. Compare individuals and profiles and 6. Visualise the results.

To support the identified features, a number of plugin libraries were developed. Plugin libraries are the functional building blocks registered for use within the solution based on their classification, which is defined by their role (acquisition, extraction, comparison or visualisation). The solution structure is logically separated by a number

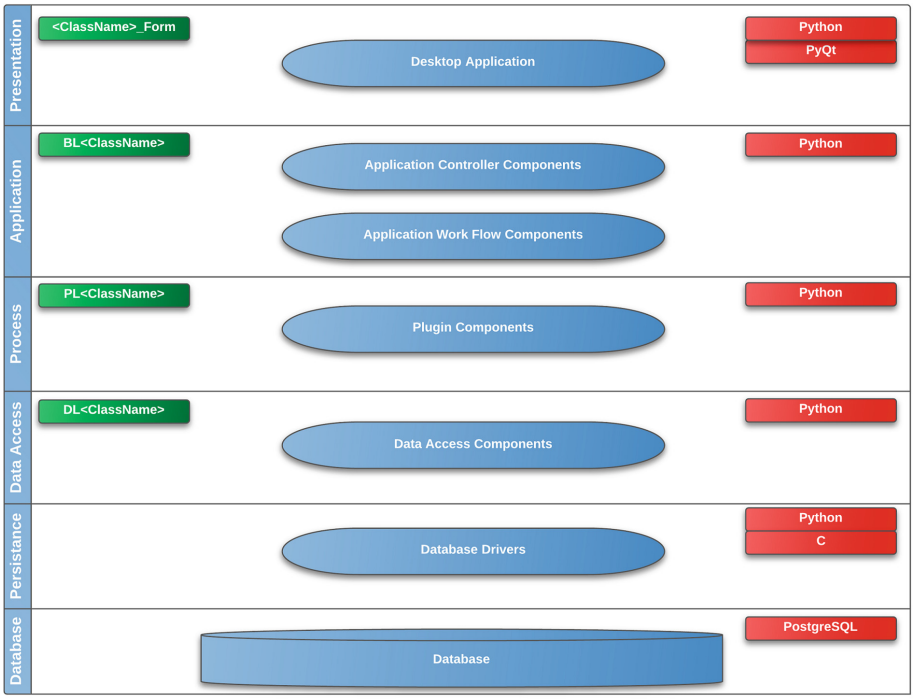

Fig. 2. High level solution architecture enabling flexibility and reusability, extending or replacing an entire layer can be done in isolation without affecting other components.

of packages. When the investigator develops a library to perform the required function, it can be then copied to the relevant package, based on the library's classification. When executed each Plugin library is launched in a separate process using the Python multiprocessing package allowing the solution to be fault tolerant. In the event of a crashing or faulting library, the solution will function as normal and no critical data will be affected. The solution monitors the newly spawned process, reporting the progress of the library back to the main application, so the user is informed at all times of its status. Once the library has completed successfully with no errors, the data generated is persisted to the database.

4.1 Investigative Case

An Investigative Case is a concept which allows investigators to logically organise their forensic case data under a structure with which they are already be familiar with. An Investigative Case consists of four components:

1. Case: The parent object underneath which all other case data is organised. Contains the case reference, overall description and relevant timestamp data.

2. Individual: The representation of a person directly involved in a case. Contains first name, surname, individual reference and relevant timestamp data.
3. Evidence: Represents the physical evidence which has been seized that is associated with an individual. Contains reference, description and timestamp data.
4. Source: Represents the location of information within the piece of seized evidence. The source contains the full path to the data directory, encase evidence file or image file that is to be acquired by the solution, as well as the operating system from which it is to be acquired.

4.2 Digital Forensic Profile

A Digital Forensic Profile allows the investigator to build up a repository of digital forensic artefacts that are logically grouped. A potential approach is to base a profile on already attained data from convicted criminals. A profile has no association with an Investigative Case. Therefore, individuals who are part of an Investigative Case can be compared directly to a predefined Digital Forensic Profile. A Digital Forensic Profile consists of three components:

1. Profile: Represents a logical grouping of digital forensic artefacts. Contains reference, description and relevant timestamp data.
2. Evidence (as above).
3. Source (as above).

4.3 Acquiring a Data Source

Once an Investigative Case structure has been created, data sources can then be acquired. The solution currently supports the acquisition of undeleted data from Windows as well as data from encase evidence and image files while running the solution on the Linux platform. When a user chooses to acquire a Source, the relevant acquisition library is executed. The location property of each Source object is used by the acquisition library to recursively extract undeleted file and folder information, which is then saved by the solution for further use by other components.

4.4 Extracting Forensic Artefacts

Once data has been acquired, libraries can be executed to extract forensic artefacts from it. The solution currently classifies forensic artefacts into three types, as displayed in Table 1. This is in order for the artefacts extracted from an application to be compared against those from a similar application such as comparing Skype to Messenger data.

Table 1. Forensic artefact classification

Type	Description
Keywords	Keywords that can be searched across a data source
Action	Artefacts extracted based on actions performed by the user. i.e. started an application, connected an external device, opened a URL, sent an IM, etc.
System	Operating system artefacts. i.e. Windows version, programs installed, etc.

The solution currently supports the extraction of Keywords, Firefox, Internet Explorer and Skype digital forensic artefacts from acquired Sources using already developed extraction libraries. Each extraction library creates a generic forensic Artefact object for each artefact discovered. This generic object contains priority, type and time stamp information and a reference to all other data associated with that artefact type.

4.4.1 Action Artefacts

Artefacts such as Internet Explorer, Mozilla Firefox and Chrome history are classified as action artefacts. For each browsing history entry, web form entry or a cookie object created by these applications, an equivalent Artefact Action object is created. This object is used to identify the type of action artefact, as well as to determine when the action began and ended. This normalised object structure, as presented in Fig. 3 allows the application to for example, compare the browsing history of Chrome to the browsing history of Firefox regardless of the data format that either application utilises. For each Artefact Action object created a referenced Artefact Action Payload object is also created which contains key data associated with each artefact.

4.4.2 Keyword Artefacts

Keyword searching is a powerful technique already used in many forensic investigations which involves the systematic searching of file systems for occurrences of specified keywords. The nature of keyword data allows it to be normalised with ease from popular and proprietary platforms. The ability to compare and visualise keyword data from multiple sources is a primary feature of the solution due to already established benefits of keyword comparison.

To acquire keyword data from the Windows platform the solution utilises a programme called strings.exe [16] (packaged with solution), while acquiring keyword data from the Linux platform utilises the Linux command *strings*. Each acquisition library searches through a set of data specified by the location property of a Source object for word strings. Each string is compared against an already predefined set of false positives attained from clean operating system installs to filter out any redundant keywords. If the string passes the false positive test an Artefact Keyword object is generated. This object represents each keyword extracted from a given location, as well as the number of occurrences within that location.

4.5 Comparing Individuals and Profiles

Once the investigator has created the necessary case structure, acquired data sources and extracted forensic artefacts using their own or solution-provided libraries, they can begin to compare the forensic data obtained. The solution currently supports the comparison of Keyword and Action artefact objects.

4.5.1 Keyword Comparison

The keyword comparison library evaluates each keyword forensic artefact associated with an individual or a profile and attempts to create a link to an artefact of the same

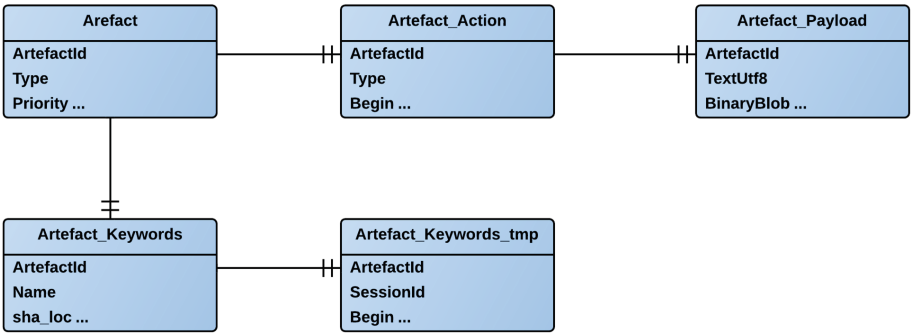

Fig. 3. Entity relationship diagram detailing the relationships between the Artefacts objects used to generate the desired architecture to facilitate the comparison of forensic artefacts.

type associated with another individual or profile by the use of a number of SQL inner joins. If there is a matching hit an Edge object is created.

4.5.2 Action Comparison

Action artefact comparisons are performed in a similar manner. The difference is that action artefacts of the same type can be compared directly against each other, regardless of the application that the artefact originated from, for example, an instant message is compared against an instant message. Based on the artefact action payload data, an attempt to create a link between each action artefact associated with an individual or a profile and another individual or a profile is performed through a number of SQL joins. If a match is found, an Edge object is created.

4.5.3 Edge

An Edge object represents a relationship and its significance that exists between two individuals or profiles, based on a digital forensic artefact they have in common. This process is repeated until all combinations have been discovered and therefore all relationships have been created as Edge objects. When an Edge object is created, a reference to the artefact which caused the creation of the Edge object is generated. Linking each relationship to a digital forensic artefact in this manner provides visibility to how each relationship was established.

4.5.4 Weighting

Artefact matches that are discovered are weighted to emphasise their importance in the context of the data set being analysed. Due to the large volumes of data that can be analysed there is potential for information overload if a large number of insignificant links are discovered and are presented to an investigator to be of equal importance. The first stage is to perform an aggregation process to discover actions or keywords with the same properties among compared parties which facilitates the calculation of the overall artefact weight.

Each action artefact discovered during extraction is created with a priority weight, this allows for the weighting of artefacts extracted by a plugin to be higher than those

of others. The priority weight of an action artefact is inserted by the running plugin either in code or through the user interface in accordance with the weight that the investigator deems appropriate. Additionally, a configurable action artefact type weight is applied which is not inserted by a plugin when an artefact is extracted, but by the solution to differentiate between various action artefacts. For example, if a plugin extracts Skype forensic data, each artefact discovered may have a priority weight of 2 as applied by the plugin, the action artefact type weight applies to the particular type of Skype artefact extracted such as a Skype contact or a Skype instant message. This allows weighting granularity of particular types of artefacts extracted using one plugin. The significance of action artefacts is calculated as follows:

((Individual1.OccurrenceOfMatch * ArtefactPriorityWeight) + (Individual2.
OccurrenceOfMatch * ArtefactPriorityWeight)) *
(Individual1.OccuranceOfMatch + Individual2.OccuranceOfMatch) *
ArtefactsTypeWeight

When a keyword match is discovered a count of the number of files that the keyword was discovered in is taken into account to determine its significance, the more a keyword appears the more significant it is. The total count of each keyword occurrence is multiplied with the number of files it has been discovered in. The keyword weighting formula is as follows:

(Individual1.OccurancesOfKeywordMatch +
Individual2.OccurancesOfKeywordMatch) *
(Individual1.NumberOfFilesKeywordFoundIn +
Individual2.NumberOfFilesKeywordFoundIn)

The presented weighting scheme assumes that the more occurrences of an artefact that individuals have in common the more significant it is. This may not be the case depending on the context and scope of the investigation. The solution can be extended to override the presented weighting scheme by abstracting the weighting calculation into a runnable plugin library. This provides flexibility allowing users to create their own weighting libraries or use already established scientific weighting schemes.

4.6 Comparison Visualisation

Once the data sets of individual's have been compared, they can be visualised in order to provide a quantitative understanding of it. The initial execution of the default visualisation plugin displays an overview of the comparison data. This is a collection of all of the Edge objects that have been created between individuals. Yellow nodes represent individuals while blue nodes represent the relationships between them, an example of which is displayed in Fig. 5.

4.6.1 Filtering Visualisation

Visualisations can become cluttered if a large numbers of relationships are discovered. However nodes which have a high degree of Edge objects associated with one another will be drawn closer together indicating a strong relationship. Additionally, users can filter artefact nodes by their weight and individual nodes based on the total weight of all artefacts between individuals which will display nodes with a higher degree of artefact relations.

5 Test Case Execution

Given below is a summary of a simulated case, created by Dr. Cormac Doherty, Dr. Pavel Gladyshev, and Mr. Ahmed Shosha for the 1st Interpol/UCD Summer school in cybercrime investigation, which was held in Dublin, Ireland in July/August 2011. It consists of 4 disk images and associated documents.

Simulated case synopsis. John Smith, who is a postdoctoral researcher in Digital Forensics Investigation Research Laboratory (DigitalFIRE) developed and patented a highly successful algorithm, which earned him millions, received a ransom letter for his son, Toby Smith. Toby is eighteen and a computer science undergraduate student at UCD. John Smith paid the requested ransom using his personal laptop without contacting An Garda Síochána. Toby was not released and An Garda Síochána were contacted regarding the case. John's laptop's HDD was then imaged. One of the prime suspect's was Mr. Paeder Patterson, John Smith's manager. The Gardaí searched the studio apartment of Paeder Patterson in UCD Residences at Belfield, where two laptops were discovered and both HDD's imaged. Mr. Patterson was not available for questioning. An Garda Síochána were then called to an address where an individual fitting the description of Toby Smith had been seen. When officers called at the address, a male ran from the building, leaving behind a laptop. The laptop was seized and the HDD imaged.

A new case with each appropriate object was created in the tool as represented below in Fig. 4. Due to the fact that Mr. Patterson was unavailable for questioning, ownership could not be established of the laptops found at his address. Therefore, two individuals marked Patterson - Image 2 and Patterson - Image 3 were created to encompass this data. One Source object is created for each evidence item. The location property of each Source object points to the user directory of each image associated with an individual.

Each Source was acquired using the appropriate acquisition library, the results of which are displayed below in Table 2.

Skype Extraction, Comparison and Visualisation. The Skype forensic artefacts' extraction library was executed against each acquired Source associated with each individual to facilitate comparison. Displayed in Table 3 are the results of the Skype artefact extraction process for each individual.

Three of the four disk images contained Skype artefacts. These artefacts can be compared to determine if there are any matches between the data sets. Displayed in Table 4 are the Skype comparison results of the three applicable individuals.

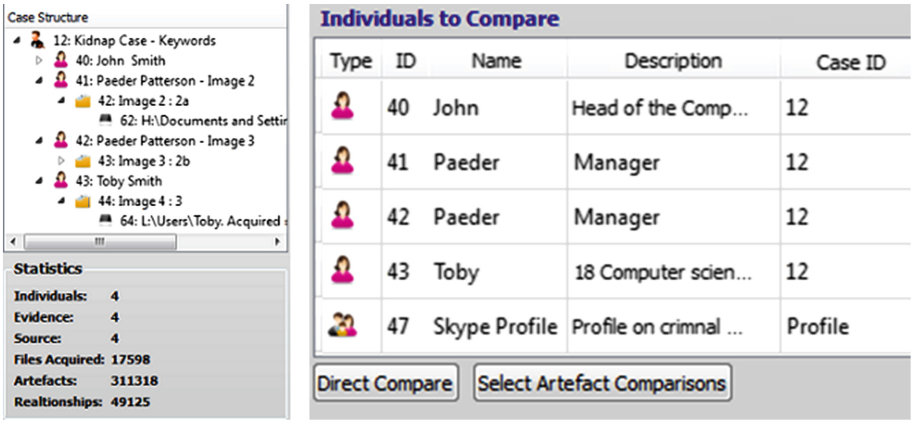

Fig. 4. Solution workbench displaying the created test case Investigative Case structure as well as important case statistics associated with the case. The compare individuals/profiles component allows users to directly compare already generated forensic data or specifically select the forensic artefact data they wish to compare.

Table 2. Acquired kidnap case data

Individual	Image name	Files acquired
John Smith	1_john	10747
Paeder Patterson – Image 2	2a	939
Paeder Patterson – Image 3	2b	3587
Toby Smith	3	2325

Table 3. Skype artefact extraction results

Individual	Image name	Skype artefacts
John Smith	1_john	61
Paeder Patterson – Image 2	2a	30
Paeder Patterson – Image 3	2b	0
Toby Smith	3	182

Once the Skype artefact data has been compared, it can be visualised using the default visualisation plugin, the results of this operation are displayed below in Fig. 5.

Each of the blue nodes displayed in Fig. 5 were expanded to display the artefacts that created the relationship, the results of which are displayed below in Fig. 6.

Firefox Extraction, Comparison and Visualisation. The Firefox forensic artefacts' extraction library was executed against each acquired Source to facilitate comparison, the results of which are displayed in Table 5.

Table 4. Skype artefact comparison results

Individual	Artefact type	Artefact	Individual	Weight
John Smith	Nickname (Skype)	novye.dengi	P Patterson – Image 2	4000
John Smith	Contact (Skype)	tobyskeeper	P Patterson – Image 2	4000
John Smith	Message sent to name (Skype)	tobyskeeper	P Patterson – Image 2	361000
John Smith	Contact (Skype)	lorna.bubbles	Toby Smith	4000
John Smith	Message sent to name (Skype)	lorna.bubbles	Toby Smith	12544000

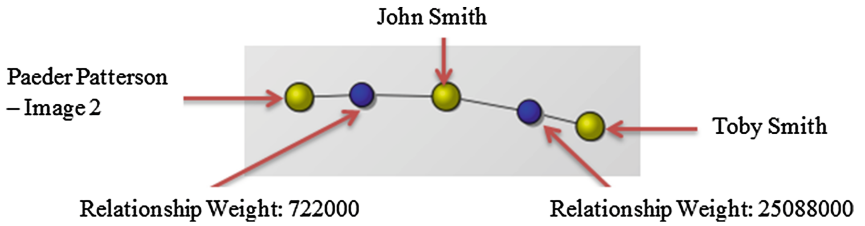

Fig. 5. Visualisation based on the Skype data that the relevant parties have in common.

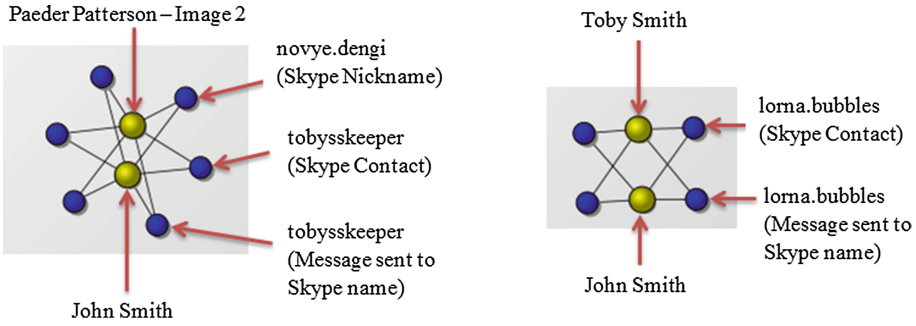

Fig. 6. Skype artefact visualisation displaying the artefact's that created the relationships.

Table 5. Firefox artefact extraction results

Individual	Image name	Firefox artefacts
John Smith	1_john	1192
Paeder Patterson – Image 2	2a	0
Paeder Patterson – Image 3	2b	0
Toby Smith	3	779

Two of the four disk images contain Firefox artefacts. These artefacts were compared to determine if any artefacts match and then visualised as displayed in Fig. 7.

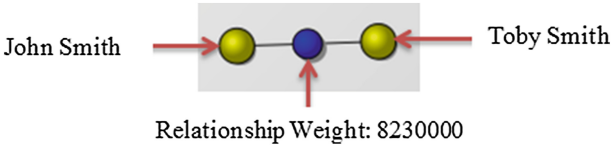

Fig. 7. Overview visualisation based on the Firefox data that both parties have in common.

The artefact relationship node is expanded to display the Firefox browsing artefacts that the individuals have in common, the results of which are displayed in Fig. 8.

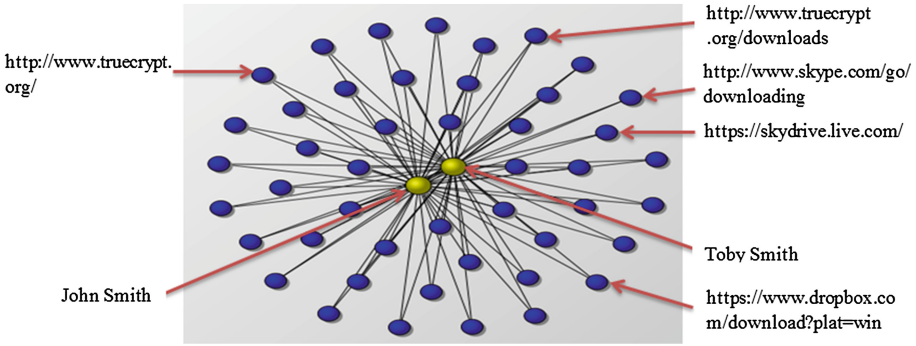

Fig. 8. Firefox artefact visualisation displaying artefact's that created the relationship (nodes of significance highlighted).

Keyword Extraction, Comparison and Visualisation. The keyword forensic artefacts' extraction library was executed against each acquired Source to facilitate comparison, the results of which are displayed in Table 6.

Table 6. Keyword artefact extraction results

Individual	Image name	Keyword artefacts
John Smith	1_john	202376
Paeder Patterson – Image 2	2a	4141
Paeder Patterson – Image 3	2b	54996
Toby Smith	3	49805

Once keyword data has been extracted and filtered against a defined set of false positives, the data associated with each individual can then be compared. Once the comparison data is generated it will be presented in the text results view of the solution

from where it can then be visualised, an example of which is displayed below in Fig. 9. The text results of these comparisons operations are displayed in Appendix A. This data has been ordered by individual and weight and additionally manually filtered to protect individual's personal information, remove IP addresses, UCD infrastructure information and irrelevant data not filtered by the false positives process to improve readability.

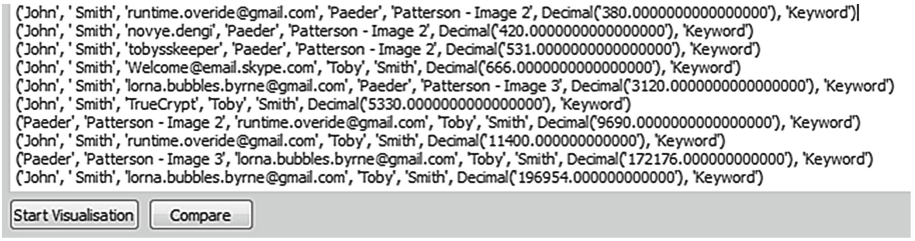

Fig. 9. Text results view of the tool based on the total comparison data of the test case.

When this data is visualised it can be easily seen that all of the individuals are connected in some way to each other, despite some of the connections being of minor relevance while others display a much greater degree of weighting significance. However if the person weight threshold filter is adjusted to 15000, Paeder Patterson – Image 2 can be seen to only have significant connections to John Smith and no connections to the other individuals in the case as displayed in Fig. 10.

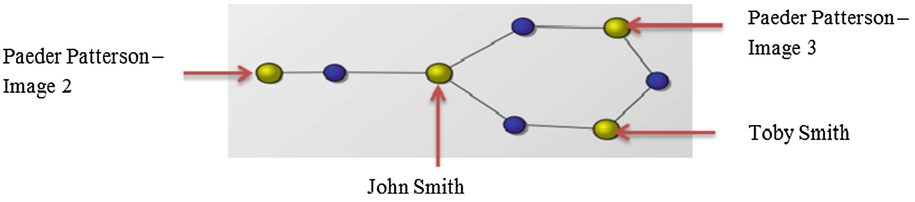

Fig. 10. Keyword overview visualisation regenerated based on increased threshold.

6 Conclusion

The tool successfully acquired, extracted and compared the various data sets associated with the individuals in this test case. The investigator is provided with compressive text data results, as well as functionality to visualise the comparison data in order to offer various perspectives of the data that would not ordinarily be available. The investigator can quickly view each relationship discovered between the various individuals and see which forensic artefact caused the relationship. Each artefact was weighted according to its significance, allowing the investigator to focus on forensic artefacts that are possibly of greater importance and thus potentially leading them to the most critical

piece of forensic data sooner. The test case results presented clearly highlight a number of significant findings.

1. No matches of any importance were discovered between artefacts associated with the individuals marked P Patterson – Image 2 and P Patterson – Image 3.
2. A keyword of runtime.overide@gmail.com with a heavy weighting was matched on the images taken from John and Toby Smith as well as the image marked 2a (Paeder Patterson – Image 2).
3. A large number of Skype messages sent between a Skype alias of novye.dengi originating from John Smith's laptop to a Skype alias of tobyskeeper originating from Paeder Patterson – Image 2.
4. John Smith is identified as a Gatekeeper entity (individual within a network which has the ability to control information between different network segments) in this social network due to a number of significant relationships being discovered between him and all other individuals involved in the case.
5. An individual with an alias of Lorna Bubbles has had contact with John and Toby Smith as well as the data present on the image marked 2b (Paeder Patterson – Image 3). This could represent a new lead in the case.
6. A large number of common browsing locations between John and Toby Smith.

Based on the findings presented by the prototype, the investigator can now narrow the scope of their investigation and focus on the areas of greater relationship density. Further detailed analysis into these results would have established:

- That the forensic image marked 2a taken from the laptop discovered at Mr. Patterson's was used by the kidnapper.
- That the forensic image marked 2a taken from the laptop discovered at Mr. Patterson's apartment was in fact planted at that address and not owned by Mr. Patterson.
- The forensic image marked 2b taken from the laptop discovered at Mr. Patterson's apartment was owned by Mr. Patterson.
- No links of any significance between the forensic images 2a and 2b taken from both of the laptops at Mr. Patterson's address.
- The Skype alias of novye.dengi found to be the Skype account that John Smith used to communicate with the kidnaper. Tobyskeeper found to be the Skype account used by the kidnaper to communicate with John Smith.
- The user of runtime.overide@gmail.com found to be a contact of Toby Smith with direct involvement in the kidnap case.

7 Evaluation and Further Research

Without performing any preliminary investigation into the test case using traditional or other forensic tools, the solution generated a substantial amount of information with which the investigator can strategically plan the rest of their investigation. This data was available within minutes, which ordinarily would have taken days if not weeks to manually generate. No special expertise to make inferences regarding the graphical data presented is required, as visual patterns are easily understood.

The prototype has the potential to save an investigator a vast amount of time and resources. This has particular relevance where many law enforcement units are under resourced and are struggling to deal with the increasing number of digital forensic cases involving huge amounts of digital evidence as well as an already substantial backlog of digital forensic cases in some instances.

The tools primary application should be as a preliminary information gathering solution prior to using traditional digital forensic tools. The results generated should be used to conduct a more targeted and focused digital forensic examination. Greater value can be attained from the results when comparing digital forensic artefact data which can be attributed to a device or an individual. The tool is of is most benefit if used to discover relationships in multiparty digital forensic cases, such as child exploitation, financial fraud, paedophile rings or cases involving a number of suspect's and victim's.

In digital forensic cases involving a single individual or a number of individuals who have no relationship in the context of the case, the prototype is less applicable. However, the support for digital profiles allows investigators to compare data attributed to a single device or individual against a predefined set of forensic artefacts. This can be of benefit if the investigator has no background knowledge of the case they are investigating and wishes to establish some initial findings. However, the discovery of no relational information in a single or multiparty case is still a valid finding and would have taken a significant amount of time to establish.

Further research and development is required to enhance the digital forensic prototype's functionality. The further addition of acquisition, extraction and visualisation libraries to process common digital forensic artefacts, as well as developing the support for comparing operating system artefacts such as programs installed, installation information and mounted devices would result in a more complete solution. Incorporating the concept of graph theory centrality into the prototype's default visualisation would provide the investigator with a greater understanding of the underlying structural properties of visualised networks. Further research into the area of artefact relationship discovery is required in order to develop the substantial benefits from automating the digital forensic investigation process.

Appendix A – Keyword Comparison Results

Individual	Keyword	Individual	Weight
John Smith	tobysskeeperTobys	P Patterson – Image 2	96
John Smith	novye.denginovye	P Patterson – Image 2	102
John Smith	tobysskeeperN7	P Patterson – Image 2	114
John Smith	runtime.overide@gmail.com	P Patterson – Image 2	380
John Smith	novye.dengi	P Patterson – Image 2	420
John Smith	Tobysskeeper	P Patterson – Image 2	531
John Smith	paeder.patterson@mobileemail.vodafone.ie	P Patterson – Image 3	294
John Smith	paeder.ucd	P Patterson – Image 3	330

(*Continued*)

(*Continued*)

Individual	Keyword	Individual	Weight
John Smith	skype.outbound.ed10.com	P Patterson – Image 3	351
John Smith	lorna.bubbles@gmail.com	P Patterson – Image 3	513
John Smith	paeder.patterson @ucd.ie	P Patterson – Image 3	22644
John Smith	paeder.ucd@gmail.com	P Patterson – Image 3	64005
John Smith	Patterson	P Patterson – Image 3	75504
John Smith	lorna.bubbles.byrne@gmail.com	P Patterson – Image 3	203194
John Smith	john.ucd@gmail.com	P Patterson – Image 3	415692
John Smith	john.smith@ucd.ie	P Patterson – Image 3	44818818
John Smith	smithtoby.smith	Toby Smith	6
John Smith	lorna.bubblesLorna	Toby Smith	6
John Smith	lorna.bubblesA	Toby Smith	6
John Smith	toby.smithToby	Toby Smith	10
John Smith	lorna.bubbles.byrne@gmail.comByrne	Toby Smith	36
John Smith	novye.dengi	Toby Smith	60
John Smith	skype.com	Toby Smith	68
John Smith	TrueCrypt.exe	Toby Smith	80
John Smith	skype.outbound.ed10.com	Toby Smith	80
John Smith	toby.smithN1	Toby Smith	92
John Smith	lorna.bubbles	Toby Smith	220
John Smith	Welcome@email.skype.com	Toby Smith	666
John Smith	toby.smith	Toby Smith	1352
John Smith	TrueCrypt	Toby Smith	5330
John Smith	runtime.overide@gmail.com	Toby Smith	11400
John Smith	lorna.bubbles.byrne@gmail.com	Toby Smith	196954
John Smith	john.ucd@gmail.com	Toby Smith	230971
John Smith	toby.paul.smith@gmail.com	Toby Smith	796500
P Patterson – Image 2	novye.dengi	Toby Smith	1176
P Patterson – Image 2	runtime.overide@gmail.com	Toby Smith	42560
P Patterson – Image 2	skype.outbound.ed10.com	Toby Smith	337
P Patterson – Image 2	lorna.bubbles.byrne@gmail.com	Toby Smith	569204

References

1. Wassermann, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, New York (1994)
2. Moreno, J.L.: Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama. Beacon House, New York (1934, 1953, 1978)

3. Almack, J.C.: The influence of intelligence on the selection of associates. Sch. Soc. **16**, 529–530 (1922)
4. Bott, H.: Observation of play activities in a nursery school. Genet. Psychol. Monogr. **4**, 44–88 (1928)
5. Scott, J.: Social Network Analysis: A Handbook, 2nd edn. Sage Publications Ltd., London (2000)
6. Xu, J., Chen, H.: Criminal network analysis and visualization: a data mining perspective. Commun. ACM **48**(6), 101–107 (2005)
7. Sparrow, M.K.: The application of network analysis to criminal intelligence: an assessment of the prospects. Soc. Netw. **13**, 251–274 (1991)
8. Eades, P.: A heuristic for graph drawing. Congressus Numerantium **42**, 149–160 (1984)
9. Verisk Analytics: NetMap. http://www.iso.com/Products/NetMap-Suite-of-Products/NetMap-Suite-visual-link-analysis-to-fight-insurance-fraud.html. Accessed 16 Apr 2012
10. IBM i2: COPLINK Accelerating Law Enforcement. http://www.i2group.com/us/products/coplink-product-line. Accessed 4 Apr 2012
11. IBM i2: Analysts Notebook. http://www.i2group.com/us/products/analysis-product-line/ibm-i2-analysts-notebook. Accessed 2 Mar 2012
12. Paterva: Maltego. www.paterva.com/web5/. Accessed 1 Mar 2012
13. Bazzell, M.: Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information, CreateSpace Independent Publishing Platform (2013)
14. Guidance Software: EnCase Forensic V7. http://www.guidancesoftware.com/encase-forensic.htm. Accessed 2 May 2012
15. X-Ways Software Technology AG: X-Ways Forensics: Integrated Computer Forensics Software. http://www.x-ways.net/forensics/index-m.html. Accessed 2 May 2012
16. Microsoft: Strings v2.5, Microsoft. http://technet.microsoft.com/en-us/sysinternals/bb897439.aspx. Accessed 15 Apr 2012