THE JOURNAL OF
DIGITAL FORENSICS,
SECURITY AND LAW

Journal of Digital Forensics,
Security and Law

Volume 10 | Number 3 Article 3

2015

On the Network Performance of Digital Evidence Acquisition of Small Scale Devices over Public Networks

Irvin Homem
Stockholm University, irvin@dsv.su.se

Spyridon Dosis
Stockholm University

Recommended Citation

Homem, Irvin and Dosis, Spyridon (2015) "On the Network Performance of Digital Evidence Acquisition of Small Scale Devices over Public Networks," *Journal of Digital Forensics, Security and Law*: Vol. 10 : No. 3 , Article 3.
DOI: https://doi.org/10.15394/jdfsl.2015.1205
Available at: https://commons.erau.edu/jdfsl/vol10/iss3/3

ON THE NETWORK PERFORMANCE OF DIGITAL EVIDENCE ACQUISITION OF SMALL SCALE DEVICES OVER PUBLIC NETWORKS

Irvin Homem
irvin@dsv.su.se

Spyridon Dosis
dossis@dsv.su.se

Department of Computer and Systems Sciences
Stockholm University
Postbox 7003, 164 07 Kista
Sweden

ABSTRACT

While cybercrime proliferates – becoming more complex and surreptitious on the Internet – the tools and techniques used in performing digital investigations are still largely lagging behind, effectively slowing down law enforcement agencies at large. Real-time remote acquisition of digital evidence over the Internet is still an elusive ideal in the combat against cybercrime. In this paper we briefly describe the architecture of a comprehensive proactive digital investigation system that is termed as the Live Evidence Information Aggregator (LEIA). This system aims at collecting digital evidence from potentially any device in real time over the Internet. Particular focus is made on the importance of the efficiency of the network communication in the evidence acquisition phase, in order to retrieve potentially evidentiary information remotely and with immediacy. Through a proof of concept implementation, we demonstrate the live, remote evidence capturing capabilities of such a system on small scale devices, highlighting the necessity for better throughput and availability envisioned through the use of Peer-to-Peer overlays.

Keywords: Digital Forensics, Digital Evidence, Remote acquisition, Proactive forensics, Mobile devices, P2P, Network performance, Availability

1. INTRODUCTION

Malevolent activities quickly adapt and evolve to align themselves with the particularities of their given environment. This is seen in that they are no longer only a confined to the physical world. They have readily adapted to the digital realm, taking up their niche markedly on the Internet. Examples of such are the Zeus banking Trojan (Stone-Gross, 2012) and the Flame malware (sKyWIper Analysis Team, 2012) stealing banking credentials and performing espionage activities respectively. They are no longer rare occurrences with mild consequences. They have permanently set up camp in intricate and surreptitious forms, taking unjust advantage over unsuspecting users going about commonplace activities on the Internet. The Regin malware (Kaspersky Lab, 2014), formally analyzed and documented in 2014 as a cyberespionage tool, is an example of this, having said to have been possibly in the wild since 2003. Today, all activities in digital realm are at the risk of being compromised by malicious actors aiming at perpetrating theft, impersonation, sabotage or to paralyze others' activities for personal benefit.

The consequences of such malicious activities for the unsuspecting user have also become more detrimental, persistent and having far reaching effects in that they are largely untraceable and easily invisible to the untrained eye. Developing novel and innovative methods that enable malicious activities to remain effectively undetected and untraceable, is the hallmark of these evildoers. They are almost always one step ahead of the pursuers. Furthermore, it is relatively easy to hide among the deluge of data that is created among communication devices that support the basic network communication on the internet. Malevolent activity in the "Digital Realm" can thus, easily become rampant and uncontrollable if there are no equally innovative methods to counter the offending actors and their activities. The rate of innovation and uptake of novel techniques by law enforcement agencies, digital forensics practitioners and incident responders must at the very least be equivalent to that of their criminal counterparts, if they are to keep up with the proliferation of crime on the Internet.

One of the foremost areas in digital crime investigations where innovative means of combatting crime are highly necessary, but largely lacking, is the *evidence capture process*. This is the initial stage of an investigation where artifacts from the scene of the crime need to be retrieved in their original form, or, in the case of digital investigations, in some form of a complete copy of the original artifact that can be proven to be devoid of any tampering (National Institute of Standards and Technology, 2004) (Scientific Working Group on Digital Evidence (SWGDE), 2006). This process needs to be performed meticulously, carefully and in many cases slowly in order to ensure that there is no potentially crucial piece of evidence left behind. This is the state of affairs in the real physical world.

However, today's crime scene is rapidly edging away from a physical reality into a more virtual one. The forms of evidence found in these "Digital Crime Scenes" have also moved from the traditional fingerprints, footprints, hair samples, blood samples or other DNA related evidence, into more digital artifacts.. Such digital forms of evidence commonly include hard-disk drives, live (RAM) memory, network traffic captures, mobile devices, RAID sets (M. Cohen, Garfinkel, & Schatz, 2009), and virtually any other form of technology that records past events of its actions; that can be captured and can be analyzed during or after the criminal event and whose integrity can be verified.

This opens the floor to almost any form of computer appliance (physical or virtual) that can be thought of. Thus arises the heterogeneity problem among devices – or simply put the seeming lack of standardization among vendors of devices that perform related tasks. Different devices may have different physical connectors, operating systems, software applications, storage formats, encoding schemes and communication protocols (CDESF Working Group, 2006). This heterogeneity makes the job of a Digital Investigator a lot more difficult because of the wide variety in which evidence could manifest itself in the wild. This greatly hampers any manual efforts of collecting evidence, even with the assistance of semi-automated tools of today such as disk imagers.

In addition to this, Electronic Crime cases today often involve more than just a single device. Several computer-like appliances including tablets, mobile phones, digital cameras, GPS devices, smart-TV's and even embedded devices such as onboard vehicle computer systems (from trucks, cars and even ships) could be seized for a single case, in order to be subjected to further investigative analysis. If we also bring in the vast realm of the Internet also into play, such evidence sources could include web application accounts, online email accounts, cloud storage facilities, network traffic captures and logs (Raghavan, Clark, & Mohay, 2009). It is not difficult to imagine that all these evidence forms could easily be part of a single case in today's world and even more so in the imminent realm of the Internet of Things. The sheer volume of data that one would have to sift through in order to investigate a single case could be in the order of Terabytes and can be a more than daunting task to perform. (Case, Cristina, Marziale, Richard, & Roussev, 2008)

Furthermore, in the realm of the Internet, composed of massively interconnected devices sharing vast amounts of highly varying data, crossing paths at high velocities, the speed of the capture of potentially evidentiary information is of essence. The same levels of meticulousness and carefulness of physical evidence acquisition may as well be sacrificed to some extent for the agility that is needed in reacting to crime in the digital world. This is because potentially evidentiary information that is not captured almost instantaneously, is likely to be lost forever in just a matter of seconds. However, this does not mean that all accuracy and care in collection of digital evidence artifacts is ignored, rather it is traded-off and reduced in favour of speed. Nevertheless, the maintenance of the chain of custody is always very important in any digital investigation. New methods of achieving similar standards of the preservation of digital evidence to those of physical evidence also need to be sought after and integrated into legal standards.

Finally, at present, investigators grapple with the problem of the relatively immature forensic tools that they are presented with. Current industry standard forensic tools such as EnCase, FTK, XRY, Volatility and Wireshark, at the moment of writing, do not cater for the highly divergent nature of digital evidence sources. Most tools focus on a single niche area such as Filesystem Data, Live Memory, Network Traffic, Mobile Devices or Log data. Some have recently begun to expand their capabilities. The latest version of Encase Enterprise v7 (Guidance Software, 2014) claims to acquire evidence from disk drives as well as RAM and some mobile devices within an organizational context. AccessData's FTK AD Enterprise (AccessData, 2014) also claims largely similar functionality. Neither deals with network traffic data yet. They are also yet to provide a comprehensive method to interface with all the variety of data present to provide a uniform investigation platform. In addition to this, current tools have rather limited capabilities for capturing potentially evidentiary data on demand over networks as well as dealing with extremely large datasets. Both EnCase and FTK products in their latest releases seem to allow for remote evidence acquisition within an enterprise network, however the performance of these are unknown and the tools proprietary, thus difficult to measure. Furthermore, most of such tools would struggle and would quickly become problematic when presented with Internet-Scale crime scenes.

In this paper, we present the architecture of a scalable, distributed, multi-component incident response and digital investigation platform aimed at dealing with large scale distributed cybercrime investigations. We name this system the Live Evidence Information Aggregator, or LEIA, in short. The LEIA architecture aims at curbing cybercrime through assisting digital forensics practitioners and law enforcement agencies in improving their digital crime response capabilities.

This is to be done through addressing several of the aforementioned problems such as the innate and growing complexity of the fast growing "Internet-of-Things" types of cases as well as dealing with the constantly growing amounts of heterogeneous data vis-a-vis the present shortage of physical resources and technical capacity within Law Enforcement. We also address the need for proactive collection of evidence from potential evidence sources on-demand over public networks, and further show the need for availability through redundancy, and faster throughput network transfers such as those seen in Peer to Peer technologies. The rest of this paper is organized as follows: In Section 2, we review related work outlining shortcomings of previous similar solutions. Section 3 describes the requirements for a comprehensive distributed digital investigation platform. The functionality of the LEIA system with particular focus on the networking component is described in Section 4. The network-focused proof of concept implementation and results are outlined in Section 5. In Section 6 and 7, we summarize the work done in this study and propose further work that may be done in this area, respectively.

2. BACKGROUND AND RELATED WORK

Several progressive efforts have been made towards improving the efficiency of the Digital Investigation process. The motivations behind these have spawned from the changing requirements of national and international legal systems, the evolution in the digital crime scene, the visible backlogs of cases overburdening law enforcement agencies and advances in technological capabilities.

Some of these efforts include: Delegation and collaboration among teams; Reduction of evidence sizes through filtering out known files; and simple automation of important but mundane, repetitive tasks (such as indexing data for subsequent searches, file carving, parsing running process in memory or TCP flows in network captures). Most of these capabilities have been implemented in current industry standard forensic tools, however, investigators and analysts still remain overburdened (van Baar, van Beek, & van Eijk, 2014). This is because of the presently abundant and steadily growing amounts of heterogeneous and disjointed datasets from multiple sources that they are tasked to collect and analyze. Methods to alleviate this problem through fully automating the remote collection and pre-processing of such data are so far either lacking in efficiency or in scalability.

Several unidirectional solutions, such as, (Almulhem & Traore, 2005) have been proposed in a bid to solve this multi-faceted problem, however, they have not been unequivocally successful. In recent times there have been initiatives to centralize evidence storage (Ren & Jin, 2005), but distribute processing among several machines (Roussev & Richard III, 2004). There has also been a push towards having the different parties, involved in solving a case to work together, even from geographically separate locations (Davis, Manes, & Shenoi, 2005), particularly with respect to technical staff in niche areas (such as filesystem forensics, network forensics, live memory forensics or mobile forensics) and the legal experts. Collaboration has been the mainstay of the attempt to get cases solved faster.

Reducing the amount of data that is needed to be collected is also a means of reducing the amount of time needed to analyze the data. This has previously been done through “Known File Filtering” as well as through scripts crafted to use heuristics (Koopmans & James, 2013). Network Security Monitoring has also been an avenue for gathering data with the help of Intrusion Detection Systems (IDS’s) assisted through data mining (Leu & Yang, 2003). However, this has been the specific mandate of the IDS, centralized or distributed, as the case may be, with terminating (end) devices or intermediary devices generally playing very minor roles in this task.

As far as is known to the author, there has not been much done, through any single initiative, in terms of expanding the scope of data captured to be the mandate of all possible devices of reasonable capability. Enabling individual devices to natively act as part of the Incidence Response System, towards the aim of collecting potential evidentiary data, has not been widely studied. Additionally, collaboration on the human processing level has been emphasized, but it has not been introduced among unrelated networked devices. These devices could possibly be harnessed to work together towards aiding in intelligent real-time capturing, filtering and processing in order to attain and retain that which could be considered as possible evidentiary data, antecedent to the event of a crime being detected. It is for these reasons that we delve into this area to explore it further.

Notable related studies include (Zonouz, Joshi, & Sanders, 2011), that describes a live network forensics system that provisions varying Intrusion Detection Systems on host machines based on their respective resource costs. It works in a virtualized environment where snapshots are taken periodically and used to revert the system back to the point before an attack began. Each system rollback results in different IDSs being deployed to collect new and possibly better information. This presupposes that the attacker re-enacts their malicious behavior in a similar way to their previous attempts, each time their efforts are thwarted by the system. Storage of the potential evidentiary information in a forensically sound manner is not particularly dealt with in this study. The aim was to understand attacks better in order to make better decisions on what kind of preventive measures to deploy.

The RAFT system (Scanlon & Kechadi, 2010) proposed an architecture for performing remote evidence acquisition from disks of computers using a live CD prepared with disk acquisition tools and networking capabilities. One of the drawbacks of this system was the inability to take live captures of the disk thus requiring the machine to be rebooted as well as needing a CD Drive or a USB port. The remote acquisition was also seen to suffer from speed deficiencies and a need was identified to improve on this. We extend on this idea but oriented towards mobile devices. (Scanlon, Farina, Khac, & Kechadi, 2014) further also describe a methodology for performing forensics on devices participating in decentralized cloud storage services such as BitTorrent Sync. They show how the default replication of data on multiple devices can help in recovering data despite it having been maliciously removed to hinder forensic analysis.

(Shields, Frieder, & Maloof, 2011), (Yu et al., 2005), (M. I. Cohen, Bilby, & Caronni, 2011), and (Moser & Cohen, 2013) describe distributed system architectures for proactive collection and summarization of evidence, with centralized data storage and processing. They are, however, particularly directed at closed domain enterprise systems, where there is some form of control and order instigated by system administrators. Participation of computer systems outside the control of the enterprise is not considered. The system being proposed in this study is aimed at being universal – applying to the entire Internet.

The work done by Redding in (Redding, 2005) is the most closely related study done in the area of pro-active and collaborative computer forensic analysis among heterogeneous systems. Redding proposes a peer-to-peer framework for network monitoring and forensics through which network security events can be collected and shared among the peers. “Analysis, forensic preservation and reporting of related information can be performed using spare CPU cycles,” (Redding, 2005) together with other spare, under-utilized, or unused resources. This system however seems to be designed to collect only network security events and not any other forms of evidence from individual host devices Furthermore it seems to be aimed towards an “administratively closed environment” under the control of some systems administrator within an enterprise. An open system that has the Internet as its domain of operation assisting in the collection of any form of computer based evidence is what is not dealt with in Redding’s work. Thus, it is this that is sought after in the current study as will be described later in this paper.

In order to facilitate uniform, seamless exchange of forensic artifacts between heterogeneous entities, some form of standardization of the transmitted evidence formats is necessary. One of the bodies that has made proposals related to this is the Common Digital Evidence Storage Format Working Group (CDESF Working Group, 2006). Other notable efforts include (Schatz & Clark, 2006) which makes use of the Resource Description Framework (RDF) from Semantic Web technologies as a common data representation layer for digital evidence related metadata, using ontologies for describing the vocabulary related to this data, and (Kahvedžić & Kechadi, 2009) where a detailed ontology of Windows Registry artifacts of interests is introduced. The Open Forensic Integration Architecture (FIA) in (Raghavan et al., 2009) and FACE (Case et al., 2008) describe methods for the integration of digital evidence from multiple evidence sources in a bid to facilitate more efficient analysis. The Advanced Forensic Format (Garfinkel, 2006), AFF4 (M. Cohen et al., 2009) and XIRAF (Alink, Bhoedjang, Boncz, & de Vries, 2006) describe annotated evidence storage formats that allow for addition of arbitrary metadata as well as interoperability among different tools.

In AFF4 (M. Cohen et al., 2009), notably, remote evidence capture, some form of availability through manually driven redundancy, and some parallelism in the evidence capture process of RAID data sets is also present. However it seems that the initiation of these processes is instigated through human intervention. They are not fully automated through machine triggers, and thus could be slow to react in acquiring evidence. The availability (fail-over) provided through redundancy is based on whether the evidence captured is required in other locations. If it is not required elsewhere, then the fail-over mechanism would not work because there would be only one copy of the evidence. The parallelism (described particularly for acquiring individual disks in a RAID set) is unclear whether it could also apply in parallelizing other potential evidence data sources such as RAM memory or NAND storage on mobile devices.

The proposed idea that this study covers is composed of several areas of specialization, namely: The Internet of Things (IoT), Intrusion Detection Systems, Peer to Peer Networks, Virtualization infrastructures, Large Scale Cloud storage and Semantic Web technologies. Most of these technologies have been previously harnessed in different capacities, singularly or in small clusters, towards the benefit of digital forensics for today's complex internetworked and intertwined cyber realm. However, to the author's knowledge, there has so far not been any work done other than our previous work seen in (Homem, 2013) and (Homem, Dosis, & Popov, 2013) that aims to merge all these technologies together, in order to provide a singular scalable solution that solves the recurring problems of large amounts of data, several sources of evidence, inability of collecting evidence efficiently over networks, heterogeneity among systems, insufficient processing power, security and privacy – that are constantly troubling digital forensic analysts and law enforcement agencies worldwide. We extend our previous work, by describing the P2P gossiping protocol further, as well as by performing tests on larger partition sizes with two more mobile devices that have slightly better computing power. More insights related to the performance and reliability of the remote evidence capture process are also realized through these tests.

3. CHARACTERISTICS OF THE DESIRED SOLUTION

Inspired by the challenges documented by Palmer at the first DFRWS conference (Palmer, 2001), we describe below a wish-list of characteristics that one would like to have in a comprehensive Digital Forensics and Incident Response system for a public open domain networked environment such as the Internet. They are aimed at complementing and updating Palmer's list in light of the current state of electronic crime and the present state of forensic tools, as described earlier.

i. *Distribution*: The ability to deal with massive amounts of distribution in terms of participants, data storage, processing and dissemination. The system needs to be able to handle the heterogeneity that may come with distributed systems as well.

ii. *Scalability*: Large scale interconnectivity, as well as the possibility of new entities joining, as well as others leaving the system dynamically and gracefully without drastic negative effects on the system. The ability to easily improve or extend the capabilities of the system through new modules is also desired.

iii. *Availability*: Providing suitable levels of functionality as and when required.

iv. *Universality*: Among the heterogeneity and lack of standardization among vendors of different systems, there needs to be some standardization and common understanding between the systems on the level of communication and storage of potential evidentiary information.

v. *Responsiveness*: The system should be able to aptly detect when a security policy has been irrecoverably violated, thus collecting information in order to pursue the perpetrators of the criminal actions. This also improves on efficiency and privacy in that the system does not have to perpetually be collecting all possible information from all possible systems.

vi. *Resource Sharing*: Today, large complex problems that are being solved through collaboration and sharing of resources as seen in Crowdsourcing, P2P networks, and cloud infrastructures. They provide on demand rapid availability of large amounts of resources from collective resource pools providing speed, efficiency and the benefits from "the wisdom of the crowd".

vii. *Integrity (Trust, Reliability & Accuracy)*: As a system facilitating law enforcement in digital crimes, the levels of trust, reliability, accuracy and integrity of the information needs to be high enough to be accepted as a veritable source of evidentiary information for a court of law. The Daubert standards and the chain of custody need to be adhered to.

viii. *Privacy & Confidentiality*: Personally identifiable and secret information must be maintained as anonymous and confidential as is reasonably acceptable, unless incriminated. Unauthorized access to such information is not to be allowed.

ix. *Security*: In addition to ensuring the security of the potential evidentiary information that it aims to collect and process, it must also take its own security into consideration – especially in terms of authentication, authorization, accountability and non-repudiation of activities undertaken.

4. LEIA: THE LIVE EVIDENCE INFORMATION AGGREGATOR

The LEIA is a 4-tiered system architecture that may be described as a combination of hypervisors, intrusion detection systems, peer to peer systems and cloud storage. It is made up of the following components:

a) The Host-based Hypervisor (HbH)

b) The Peer-to-Peer Distribution Architecture (P2P-da)

c) The Cloud-based Backend (CBB)

d) The Law Enforcement Controller (LEC)

The functionality of each of the layers of the LEIA system is briefly described in the following sections.

4.1 The Host-based Hypervisor (HBH)

The Host-based Hypervisor (HbH) system is composed of a virtualization layer managed by a hypervisor – a privileged secure platform managing the guest operating system (OS). The hypervisor contains an inbuilt host-based intrusion detection system also termed as the embedded intrusion detection system (em-IDS). Security utilities within the guest OS such as anti-malware tools and intrusion detection systems maintain their own data and logs that are accessible to the HbH. The HbH collects and assimilates the information that it gets from its own inbuilt intrusion detection system together with other information collected from the other security utilities that may exist within the guest OS. This helps in getting a better perspective of current malicious activity that may be underway.

Further to this sharing of information within a single HbH system, individual HbH systems also share their information about malicious activity they may have discovered with each other. This communication is facilitated through the Peer-to-Peer Distribution Architecture (P2P-da). This collaborative effort among the HbH systems further helps improve the accuracy of IDSs and eventually forensic data acquisition.

In order to reduce the amount of data that may need to be collected for analysis, each HbH maintains a hash list of the local files on its guest operating system (Local - Known Data Hash-List, L-KDHL). This L-KDHL is periodically cross-checked and updated against a Master – Known Data Hash-List (M-KDHL) stored at the Cloud-based Backend (CBB). This is managed by the Cloud-based Backend Differencing Engine (CBB-DE) component of the CBB. The aim of this is to quickly filter out known system data or files through matching the files on an HbH against hashes of system files that are known to be benign and have not been modified in any way.

A user data profile with its corresponding hash-lists is also created. The user-data hash-list is also maintained in a dual format – with a local copy residing on the HbH and a remote master copy being maintained at the CBB. Further to this the actual user data is backed up at the CBB. Thus, the user data hash lists are used to check which files have changed and may need to be backed up to the CBB.

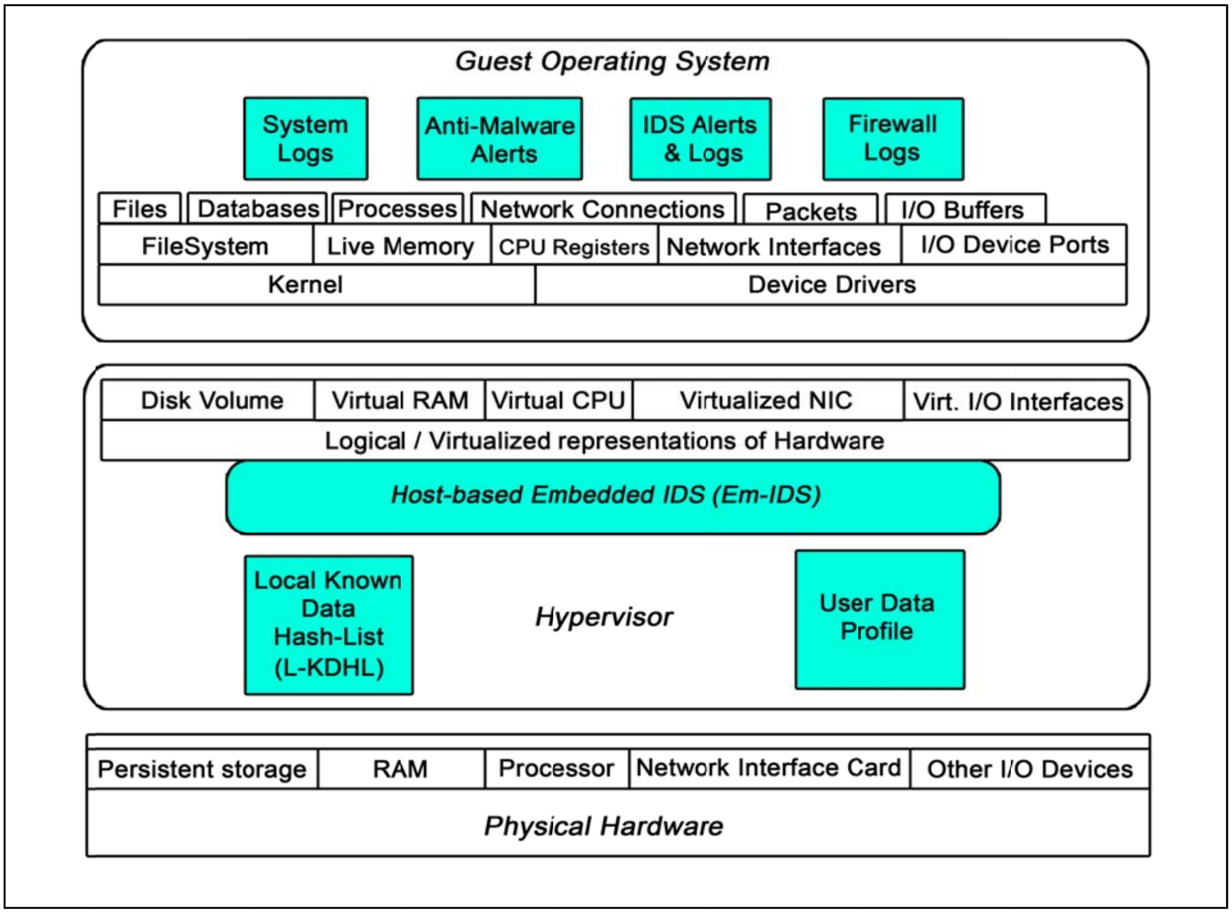


Figure 1. The components of the HbH subsystem

With respect to "Known Malicious Files" which are files that have been previously identified as having malicious content within them, a "Known Malicious File" hash list is to be maintained only on the CBB. It is not held on individual HbH systems as it can easily become large and unmanageable for a single HbH to maintain.

The hypervisor is the critical element when it comes to the collection of potential evidentiary data. Having privileged access, the hypervisor is able to directly interact with the file system, network interfaces, memory caches and other low-level resources, which are all primary sources of common evidentiary data in digital investigations. The embedded IDS's (em-IDS) also collects information mostly in the form of logs which are parsed to result in synthesized alerts. When evidentiary data from the local HbH is collected, it is transmitted towards the CBB via neighbouring HbH systems through the action of the P2P-da system (described in the next section). While such data is in transit through a neighbouring HbH system, and travelling onward to the CBB, it is always held in an encrypted form and only within temporary storage.

4.2 The Peer-to-Peer Distribution Architecture (P2P-da)

The essence of the P2P-da is to provide reliability, scalability and rapid throughput of transmitted data even in the face of high rates of "churn", that is, large numbers of nodes joining and leaving the network. In order to achieve this, a cocktail of P2P protocols are put together in order to extract the best qualities of these and also allow for each to compensate for the other's shortcomings. The particular P2P protocols that are put together in order to build the P2P-da are: Gradient overlay protocols (Sacha, Dowling, Cunningham, & Meier, 2006) Epidemic protocols (Jelasity, Voulgaris, Guerraoui, Kermarrec, & Steen, 2007), and the BitTorrent protocol (B. Cohen, 2003).

There are 3 main functionalities of the P2P-da:

I. Maintenance of the P2P Overlay

II. Dissemination and aggregation of Malicious Behaviour Information and alerts.

III. Incident response data collection

These functionalities generally correspond to the P2P protocols that make up the essence of the P2P-da. The function of the maintenance of the P2P overlay is facilitated mainly through gradient (hierarchical) overlays assisted through epidemic (gossip-based) overlays. The dissemination and aggregation of malicious behavior information is mainly facilitated by epidemic (gossip-based) overlays. Incident response data collection is mainly facilitated through an adaptation of the BitTorrent protocol. The details behind these individual functionalities are dealt with in the following sections.

4.2.1 Maintenance of the P2P Overlay

The essence of this is for the overall P2P network to maintain connectivity among neighbouring nodes as well as the larger HbH node population. Further to this, the aim here is to link HbH nodes in such a way that they are most beneficial to each other as well as to the overall communication of security events and evidence transmission aims.

In order to do this, a hierarchy is to be created among the peer nodes such that those less endowed with resources are lower in the hierarchy and those that are better endowed are higher in the hierarchy. The aim of this is to ensure that nodes that lack resources generally communicate security event information, or transmit potentially large evidence files towards more reliable and stable peers. It is assumed that nodes with more resources are more likely to be better equipped to deal with larger amounts of information and are also more likely to be online and available to be communicated with.

A gradient overlay network is suited to ensure this form of a network structure. It is built in such a way that a *utility metric* is used to determine which nodes are most suitable to connect to, and which nodes to avoid. This utility metric is determined from a combination of factors including the amount of resources available on a node, the current state of use of the node and the amount of time that it has been online. These utility metrics are shared through random node interactions, typical of "gossip-based" (epidemic) P2P protocols in order for nodes to get to know of other nodes that might be better to link to.

As gossip-based P2P protocols are known to eventually converge to a generally stable state, a hierarchy of the HbH systems is thus formed with the less endowed elements on the outer edges and the more capable elements closer towards the centre of the LEIA system (that is, the CBB).

Mechanics of the Gossiping and Gradient Overlay

Neighbour (peer) selection is an important process in maintaining both overlays. This is because it affects the performance of the gossiping overlay in its ability to communicate and converge, and thus also the gradient overlay. In order to be efficient in converging information across a network where random interaction is a key factor, each peer has to be equipped with good "local knowledge" as well as "distant knowledge". In our case this would require each peer to have knowledge of both peers that are "nearby" as well as peers that are "distant". The distance metric determining this distance could be simple Euclidean distance in terms the difference between the utility metrics of devices, or some other useful sense of distance, such as the geographic distance. Thus each peer should store information of "nearby" devices termed as the *similar set*, consisting of devices with similar utility metrics; as well as a set of devices which are distant in terms of the utility metric, termed as the "*random set*". This mix of nearby and distant peer knowledge would also assist in preventing partitioning of the network due to excessive clustering.

For the gradient overlay to be maintained, each peer should also maintain information about a set of peers that has "weaker" utility metrics and a set that has "stronger" utility metrics. This is in done in order to maintain communication between peers where the hierarchy of the gradient is needed. Such is the case when a peer needs to pass on more computationally intensive tasks, or when ensuring that evidence is transported onward to a more stable peer. We term these lists the *subordinate* list and the *superior* list, respectively.

These lists are exchanged among peers periodically and randomly. This means that a peer picks a peer either from its similar set or its random set and exchanges all or part of its subordinate and superior lists if they are different. Thresholds need to be set in order to ensure that the exchanges do not result in only a particular set of peer information being exchanged, however these thresholds are not further discussed in this study. Additionally, a list of "recently seen peers" should also be maintained in order to avoid cycling between the same peers repeatedly.

4.2.2 Dissemination and Aggregation of Malicious Behaviour Information & Alerts

This capability is necessary in order to facilitate the collaborative mechanisms needed to ensure that security event information is shared, and that potentially useful evidence information is captured efficiently and transmitted securely. Security event information known by individual HbH peers is duly shared out to others in order for the overall system to have a more informed security posture as well as to be forewarned of imminent malicious events. This includes the distribution of malicious activity signatures as well as the discovery of malicious activity originating from certain hosts. When such messages are received, only a set of the most common and recently active malicious activity signatures are maintained at the HbH. These kinds of messages are termed as “Management messages” and can be shared out to any peers that a particular HbH has address information about and that have connectivity.

The other major types of messages that are involved in this functionality are termed as “Security Incident Control messages”. These messages facilitate the reaction to the detection of a malicious event. This mainly includes the communication of procedures to initiate the evidence capture process on certain components of certain nodes as well as initiating initial pre-processing such as determining IP addresses of recently connected devices in order to extend the evidence capture process to other suspected devices.

There may be other forms of messages that might need to traverse the P2P-da; however, the 2 categories mentioned thus far are the major types.

4.2.3 Incident response data collection

This functionality is triggered by the detection of malicious events via the collective knowledge gained through collaborating HbH systems, the em-IDS and guest OS security mechanisms. For more volatile data such as network traffic and live memory, a fixed time period is chosen for which to perform the capture process (or a fixed number of snapshots of the data over a short period of time particularly for live memory) after which a decision is to be made whether subsequent captures need to be made, or whether what has been collected so far suffices. Correspondence with the Cloud-Based Backend-Differencing Engine (CBB-DE) filters out known system files through facilitating the hash comparisons. Primary analysis for IP addresses and hostnames on the data collected may result in triggering of other HbH systems to capture data also.

The actual data collection procedure involves 3 stages as described in the following sections. The diagram below (Fig. 2) depicts the data collection and transfer process of the P2P-da.

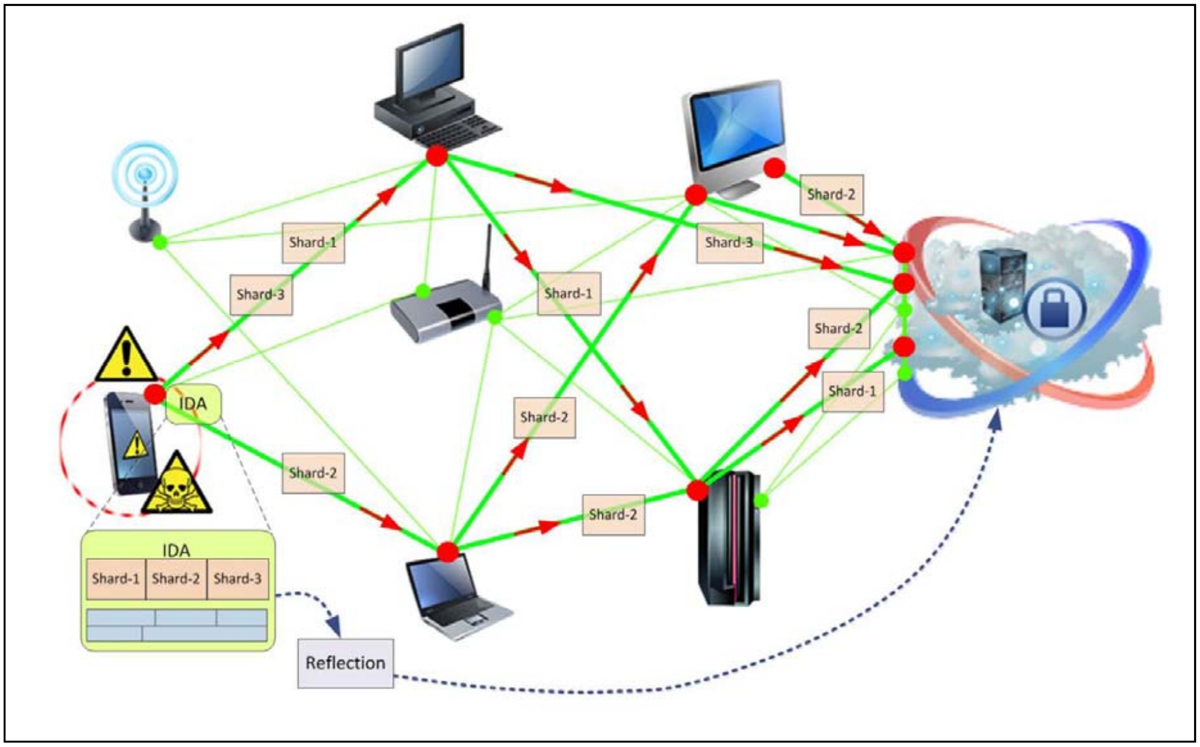


Figure 2: The P2P-da Data Transfer process

a) Data Partitioning

Different data formats (memory dumps, logs, files, packet captures, disk images) are compressed and stored temporarily on the HbH system in a modified AFF4 data structure that also contains simple RDF metadata describing the evidence. This data structure is termed as the Incident Data Archive (IDA). Each IDA data structure is partitioned in equal size pieces that will be referred to as shards. The shard is a signed and encrypted partition of the IDA analogous to the idea of a "piece" in the BitTorrent Protocol. A metadata file termed as the "reflection" (which corresponds to the BitTorrent Metadata file) is also created and sent directly to the CBB. In this way the CBB acts as the "tracker" and "leeches" IDAs from participating HbH systems in the P2P-da, thus benefiting from the high throughput of the BitTorrent protocol

b) Shard Distribution

Multiple copies of each individual shard are distributed to more capable neighbors (supporters), facilitated by the gradient overlay. Each time a shard is passed on it increases its "heat level". After a certain "heat" threshold (that we refer to as the "melting point") a HbH system is obliged to directly upload to the CBB (more specifically the HbH Master Peers of the CBB), else an election procedure is initiated to determine which previously supporting HbH should be delegated the uploading task. In order to avoid an individual node being the only "proxy" and thus a potential single point of failure, individual HbH systems are only allowed to partake in uploading a certain number of IDA shards governed by the "dependency value". This improves the overall reliability of the larger system through reducing the possibility of having a single point of failure in the transmission process.

c) Rapid fragment reconstruction

For a particular shard, downloads are initiated from all their respective supporter locations. This is done for redundancy and bandwidth maximization purposes. Similar to the BitTorrent Protocol download, priority is given to the shards that are the least commonly available, that is, those that have the fewest recorded supporters.

In order to reconstitute the IDA, individual hashes of shards are verified as they are received, against that in the reflection. Several supporters upload at the same time, thus if a shard is in error, that from another supporter is taken. Once successfully transferred, shards are deleted from supporting HbH systems.

4.3 The Cloud-based Backend (CBB)

The CBB system is a highly available, scalable, responsive, centralized back end storage service capable of storing large amounts of data in a homogeneous form. It is subdivided into 3 major components: The Storage System (SS), the Differencing Engine (DE) and the HbH Master Peers.

The Storage System (SS) is built upon the Hadoop HDFS architecture (Shvachko, Kuang, Radia, & Chansler, 2010) that provides not only the raw storage capabilities but also scalability, availability, reliability and responsiveness. The Differencing Engine (DE) filters out known files before having them stored on the CBB. This is provisioned through the MapReduce (Dean & Ghemawat, 2008) capabilities supported by Hadoop. The DE also provides a query-response mechanism to the HBH systems with information on known benign data as part of the Master Known Data Hash-List (M-KDHL). The M-KDHL contains data about known files, memory processes, protocol flows, and log entries and thus enables their removal from IDAs being prepared. This reduces the size of IDAs before being stored on the Storage System (SS) of the CBB. Versions of known data entries in the M-KDHL need also to be maintained in a scalable manner; however the mechanics of this is not discussed in this study.

The HbH Master Peers are a particular set of well-endowed peers that are directly connected to the core CBB system (that is, the SS and DE) providing an interface to the rest of the LEIA system through the P2P-da. They do not have other core functionalities unrelated to their LEIA responsibilities and are essentially the backbone of the P2P-da and ultimately the provider of connectivity of the LEIA system outwards to the other HBH systems. The HBH Master Peers also serve as the central point through which system software updates and malicious event detection heuristics are originated from and disseminated outwards to the HBH systems in the wild.

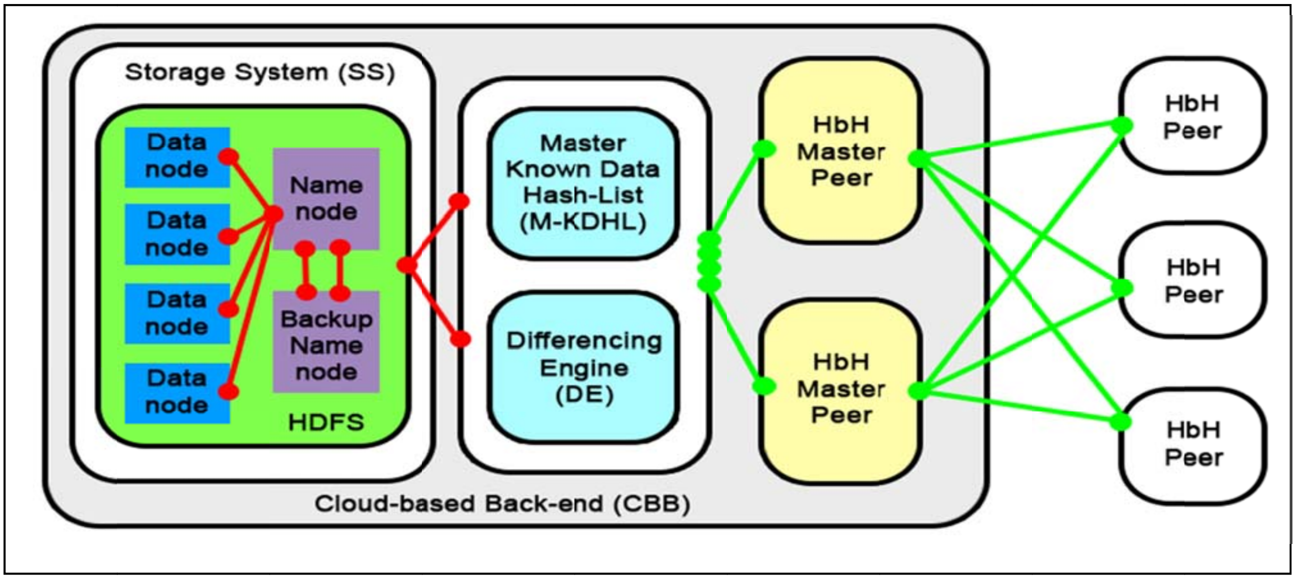


Figure 3: The Cloud-based Backend components

4.4 The Law Enforcement Controller System

The Law Enforcement Controller is the main interface that law enforcement personnel interact with in order to perform their directed analysis for a particular digital investigation case. Through it, a Law Enforcement Agent can initiate specific queries to the data sets stored on the CBB, thus retrieving detailed, structured information as well as new knowledge inferred through correlation of data originating from different sources that may help in solving a case. The aim of this is to automate otherwise manual tasks of correlating data from different heterogeneous sources in order to pose valid assertions based on the data that could assist a forensic analyst in performing their duties of making sense of digital artifacts. This functionality is described in more detail in (Dosis, Homem, & Popov, 2013).

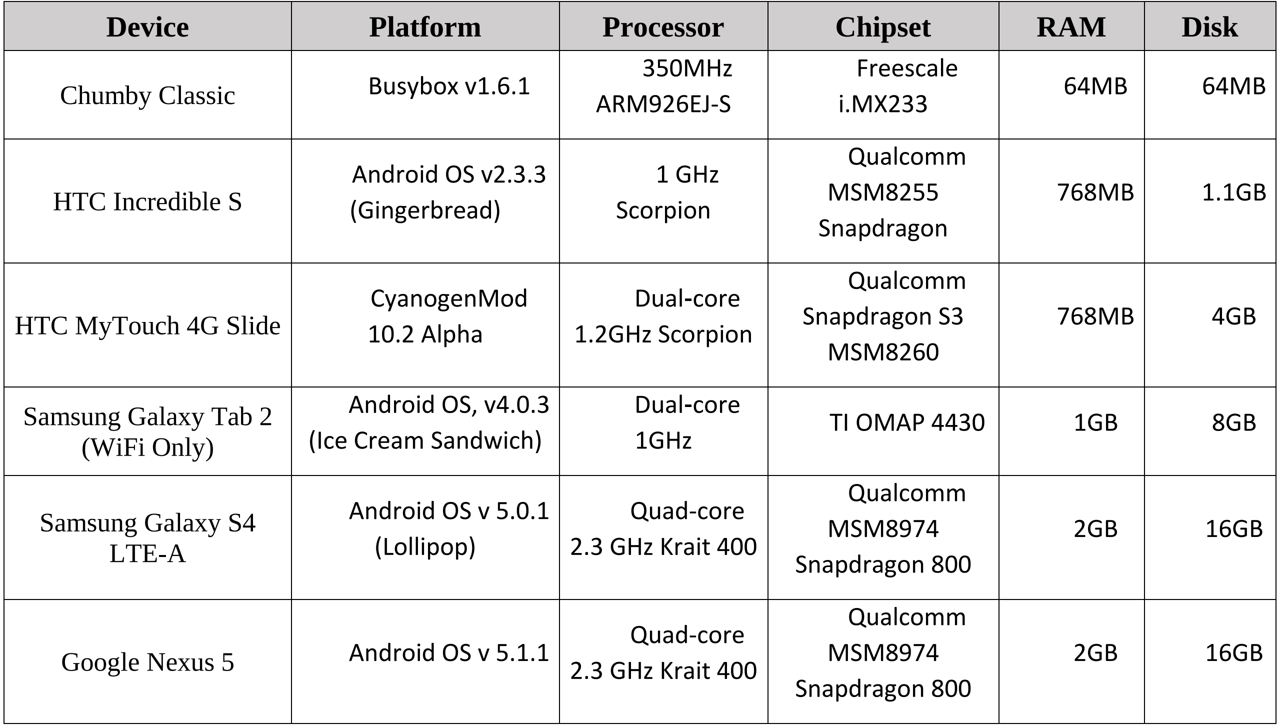
Additionally, from the new found knowledge obtained through correlation, patterns of malicious activities are to be learnt and stored. These Malicious Activity Patterns are to be used as feedback to the HbH systems in order to improve the detection capabilities of the inbuilt IDS's and thereby also improve the accuracy of collection of data of potential forensic evidentiary use.

5. PROOF OF CONCEPT EVALUATION AND RESULTS

As the first part of testing the motivations behind the designed architecture, we decided to focus on the network transmission component as it is critical in enhancing speedier evidence collection. In order to demonstrate the need for better throughput networks and the need for redundancy for availability, such as those exhibited in P2P overlays, an experiment was set up to simulate the conditions of the LEIA, however without the P2P-da component. This means that the experiment was performed with the transmission of potentially evidentiary information from an HbH system to the CBB over a traditional client-server paradigm. The experiment itself focused on the time taken to perform remote extraction, compression and transmission of increasingly larger disk images over an encrypted channel from small scale devices over the Internet and the subsequent reconstruction and storage of this data on a Hadoop HDFS cluster. The success rate of each evidence acquisition trial run was also considered to determine reliability.

It should be mentioned that for the sake of simplicity of the experiment, the actual hypervisor of the HbH system was not built, however closely similar conditions – particularly in terms of the LEIA prototype application having privileged access – were met. In order to test and measure the performance of the proof of concept application working over the client-server paradigm, six different small scale devices were used. Two rounds of testing were performed: the first round with the first 4 less powerful devices, and the second round with the 2 more powerful devices. The table below outlines the specifications of the devices being captured.

Table 1
Small scale device specifications



Device	Platform	Processor	Chipset	RAM	Disk
Chumby Classic	Busybox v1.6.1	350MHz ARM926EJ-S	Freescale i.MX233	64MB	64MB
HTC Incredible S	Android OS v2.3.3 (Gingerbread)	1 GHz Scorpion	Qualcomm MSM8255 Snapdragon	768MB	1.1GB
HTC MyTouch 4G Slide	CyanogenMod 10.2 Alpha	Dual-core 1.2GHz Scorpion	Qualcomm Snapdragon S3 MSM8260	768MB	4GB
Samsung Galaxy Tab 2 (WiFi Only)	Android OS, v4.0.3 (Ice Cream Sandwich)	Dual-core 1GHz	TI OMAP 4430	1GB	8GB
Samsung Galaxy S4 LTE-A	Android OS v 5.0.1 (Lollipop)	Quad-core 2.3 GHz Krait 400	Qualcomm MSM8974 Snapdragon 800	2GB	16GB
Google Nexus 5	Android OS v 5.1.1	Quad-core 2.3 GHz Krait 400	Qualcomm MSM8974 Snapdragon 800	2GB	16GB

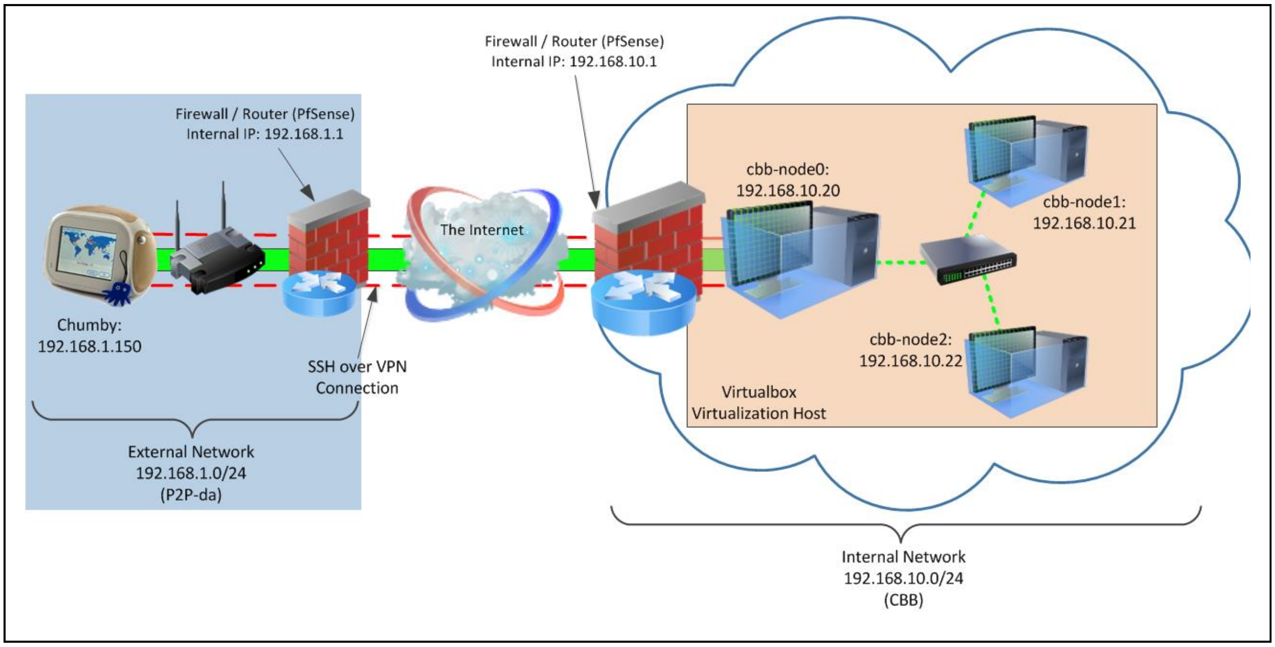


Figure 4. The experimental set up

For the testing and the performance evaluation, partitions of the various devices were filled to specific size limits with random files, including images, PDFs, music files and compressed archive files (RARs) in order to simulate normal disk usage. These devices were subsequently captured over the network.

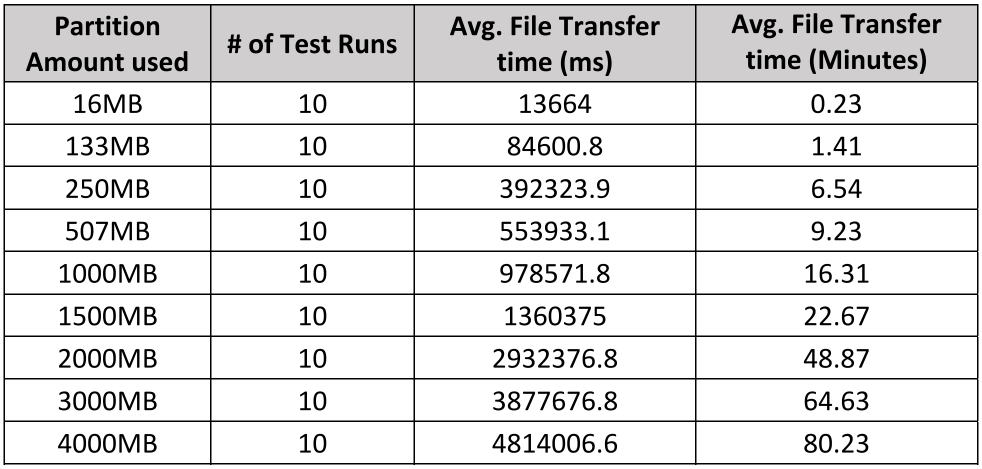
In the first round, the capture process was repeated 10 times for each individual partition size on each device in order to get the average file transfer times that each size took. The sizes measured were taken at 9 intervals with gradually increasing sizes. The maximum size of 4GB was taken as the largest size because the average capture (file transfer) times were beginning to take rather long periods (50-80 minutes) per test acquisition round. Furthermore, the maximum disk size on any of the devices available for testing was 8GB (with the rest being 4GB, 1.1GB and 64MB). A 4GB mini-SD card was also available and was used to supplement the HTC Incredible S in order to simulate a larger disk size. The Chumby Classic only had 64MB available of flash (NAND) memory, and no expansion capabilities, thus it was not included in the testing for remote data transfer performance as there was no way to increase the size of the storage capacity. It was, however, used in testing to show that the remote device capture of such a small scale device running on a Linux based platform was possible. It was also used as the main prototyping device because it had a rather small storage capacity that enabled rather quick disk acquisitions when testing the software developed.

The repetition process and the use of the averaging were done in order to compensate for the effects of random processes that could have affected network transmission times. Such random processes could include network traffic from other users of the networks being used, phone calls coming in and interfering with the I/O processes of the devices, or applications being updated on the devices, among others.

The tables below show the partition sizes used and the average times (in milliseconds) taken to perform the transfer:

Table 2

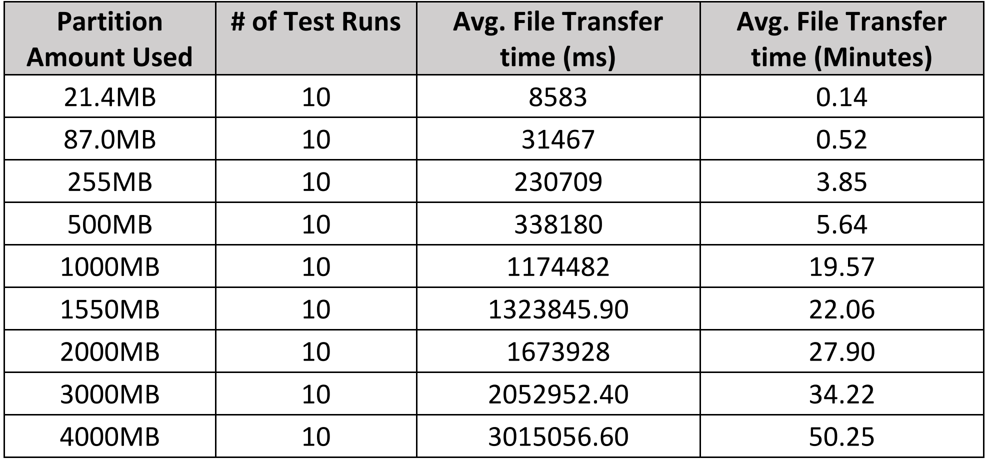
First Round Results – Tests on "HTC Incredible S"



Partition Amount used	# of Test Runs	Avg. File Transfer time (ms)	Avg. File Transfer time (Minutes)
16MB	10	13664	0.23
133MB	10	84600.8	1.41
250MB	10	392323.9	6.54
507MB	10	553933.1	9.23
1000MB	10	978571.8	16.31
1500MB	10	1360375	22.67
2000MB	10	2932376.8	48.87
3000MB	10	3877676.8	64.63
4000MB	10	4814006.6	80.23

Table 3

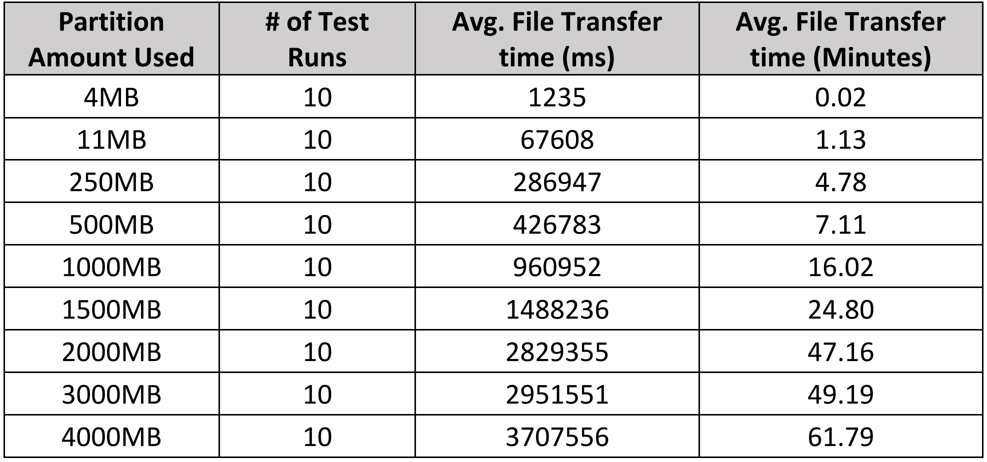
First Round Results – Tests on "HTC MyTouch 4G Slide"



Partition Amount Used	# of Test Runs	Avg. File Transfer time (ms)	Avg. File Transfer time (Minutes)
21.4MB	10	8583	0.14
87.0MB	10	31467	0.52
255MB	10	230709	3.85
500MB	10	338180	5.64
1000MB	10	1174482	19.57
1550MB	10	1323845.90	22.06
2000MB	10	1673928	27.90
3000MB	10	2052952.40	34.22
4000MB	10	3015056.60	50.25

Table 4

First Round Results – Tests on "Samsung Galaxy Tab 2"



Partition Amount Used	# of Test Runs	Avg. File Transfer time (ms)	Avg. File Transfer time (Minutes)
4MB	10	1235	0.02
11MB	10	67608	1.13
250MB	10	286947	4.78
500MB	10	426783	7.11
1000MB	10	960952	16.02
1500MB	10	1488236	24.80
2000MB	10	2829355	47.16
3000MB	10	2951551	49.19
4000MB	10	3707556	61.79

The data above from three of the four different specimen devices used in the first round was plotted on a graph in order to visualize the general trend of the file transfer time against the partition size for the client server network paradigm of remote evidence acquisition. The diagram that follows depicts the graph that was attained:

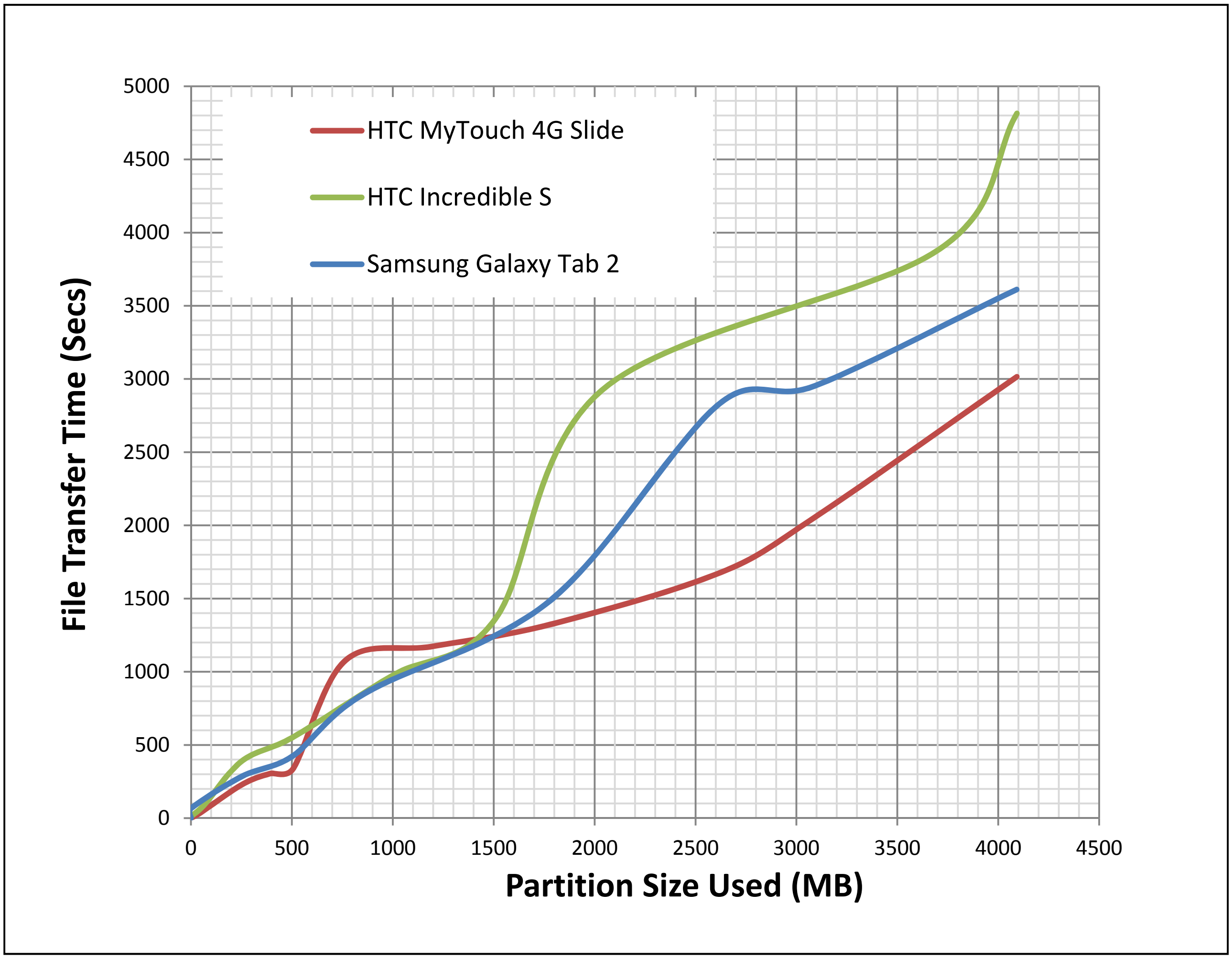


Figure 5. Remote Acquisition Performance from First Round of Tests using Client-Server paradigm

In the second round of testing and performance evaluation the two most powerful and best equipped devices were tested. The process from round one was replicated, but now with larger partition sizes since these 2 devices had larger disk storage capacity (16GB). It was noticed in the first round that capturing the larger partition sizes was not always successfully completed. In the first round, in order to collect 10 time duration values for the larger partition sizes, often more than 10 trial runs were required. Connection time-outs due to random failures either on the devices, the network, or on the server side were deemed to be the cause of failure.

Thus in the second round it was determined that strictly 10 runs would be taken and an average of these would be the resulting estimated value. Among the 10 trial runs the number of successes and failures would be recorded. This would enable determining the average success rate for ever increasing partition sizes.

The tables that follow indicate the partition size intervals and the respective time durations taken. The number of successes out of 10 trials per partition size is also indicated.

Table 5

Second Round Results – Tests on "Samsung Galaxy S4"

Partition Amount Used	# of Successful Trials	Avg. File Transfer time (Sec)	Avg. File Transfer time (Minutes)
263MB	10	275	4.58
535MB	10	535	8.92
1055MB	10	887	14.78
2168MB	10	1735	28.92
3221MB	10	2617	43.62
4304MB	9	3358	55.97
6470MB	6	5311	88.52
8637MB	6	6953	115.88
10247MB	5	8373	139.55
12778MB	4	10472	174.53
15758MB	3	13088	218.13

Table 6

Second Round Results – Tests on "Google Nexus 5"

Partition Amount Used	# of Successful Trials	Avg. File Transfer time (Sec)	Avg. File Transfer time (Minutes)
263MB	10	270	4.50
535MB	10	497	8.28
1074MB	10	820	13.67
2168MB	9	1780	29.67
3221MB	10	2757	45.95
4304MB	8	3526	58.77
6470MB	5	5412	90.20
8637MB	4	7119	118.65
10247MB	2	9369	156.15
12778MB	2	13001	216.68
15758MB	2	15117	251.95

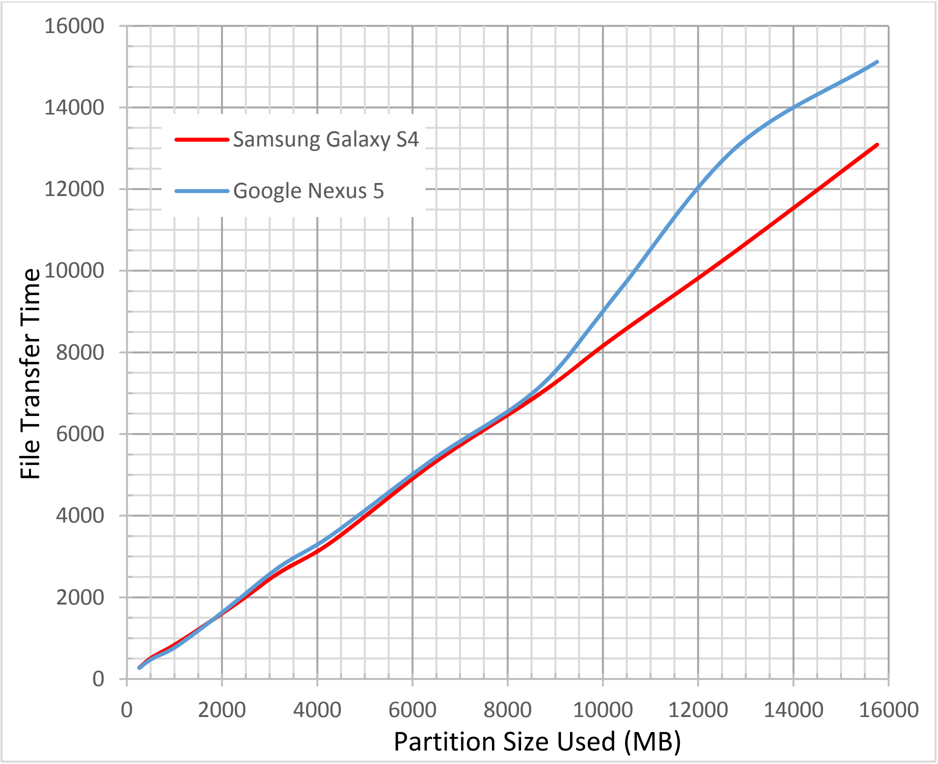


Figure 6. Remote Acquisition Performance from 2nd Round of Tests using Client-Server paradigm

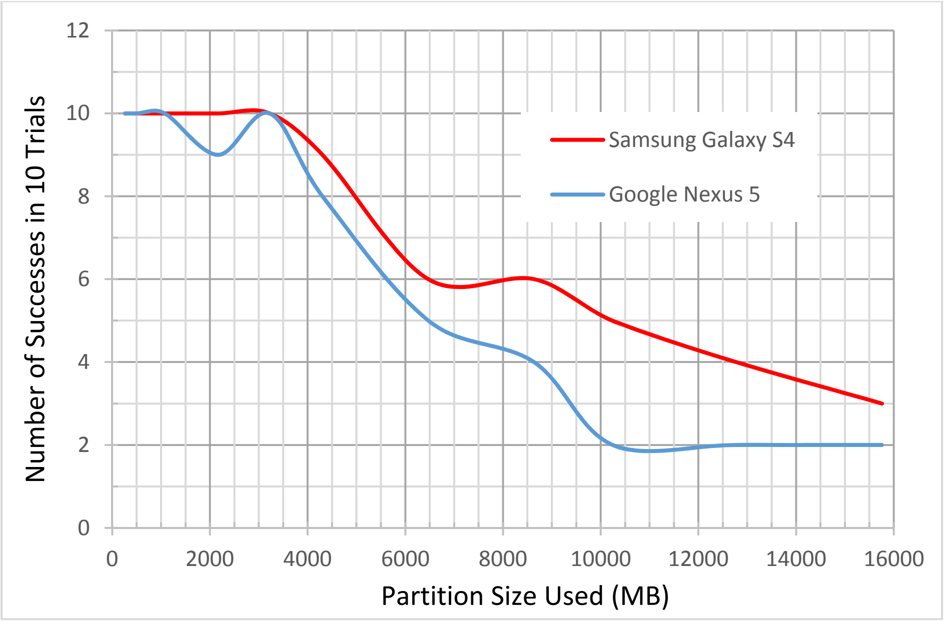


Figure 7: Remote Acquisition Successes out of 10 Trials from 2nd Round of Tests using Client-Server paradigm

6. DISCUSSION

In Figure 5, taken from the data in the first round of tests, the curves seem to start off with a linear relationship which soon seems to turn into more of an exponential relationship. The "HTC MyTouch 4G Slide" clearly portrays this characteristic, with the rest of the devices also exhibiting this however not as vividly. Overall there seems to be a more exponential relationship between the Partition Size and the File Transfer Time with respect to the larger sizes of partitions. One could posit that as the partition sizes increase, even to sizes substantially larger than those in the graph, the relationship will become ever more exponential. This means that the times taken to acquire such partition sizes would be increase in exponential magnitude and thus shows that the client-server paradigm is likely not suitable enough for the task of performing remote evidence acquisition, especially in the type of environment that the LEIA system is aimed at. This could imply the need for a more efficient network transfer paradigm for this type of activity in order to enhance the throughput.

However, in Figure 6, taken from the data in the 2nd round of tests – using more powerful devices as well as larger partition sizes – the curves seem to take a more linear shape. This could be attributed to a variety of factors including the availability of better hardware in the form of CPU capacity, higher RAM, better network cards, better I/O speeds between components, updated software, kernel modules and drivers such as the Android ION memory manager.

The major difference between the sets of devices used in the first round and those in the 2nd round is the improvement in hardware and software capabilities. It could be said that lower spec devices may not be able to scale well with large amounts of storage space to be transferred over the network. Higher spec devices may be able to scale suitably with an increase in disk storage, however, perhaps a mismatch in terms of a significantly larger difference between what is fitted into these mobile devices by default and additional external storage attached to them (E.g. through large SD cards) may cause the scaling of remote evidence acquisition to defer from the linear pattern.

As was noted in the first round of testing, and thus measured further in the second round, the success rate of the remote evidence acquisition reduced with larger partition sizes. This is seen in Tables 5 & 6 as well as in Figure 7. As the partition sizes and effectively the amount of data requiring transfer increases, the chances of a one-time success were seen to reduce. This can be attributed to the fact of the data transfer times taking rather long, thus increasing the time-window (and thus the likelihood) in which an unpredictable network interruption could cause the transfer to fail. Additionally, it was noticed that as the available disk space gradually came closer to being fully utilized, the more likely the devices were to behave erratically. The general erratic nature of suddenly freezing (requiring a reboot) while performing the remote evidence capture, as well as the presence of error notifications related to high disk usage, were observed in both devices used in the second round of tests. The lack of an expansion slot for external secondary storage in the form of an SD card on the Nexus 5 perhaps could have led to slightly worse performance compared to the Galaxy S4. It is possible that the Galaxy S4 made use of the SD card placed in its expansion slot to cache data; however this was not independently verified.

In the proof of concept set-up, partitions are captured through the "dd" tool, compressed and streamed over an encrypted channel to the Cloud-based-backend (CBB) storage site, where the actual file is reconstructed. Other than the inherent re-transmission capabilities of the underlying TCP protocol, the proof of concept does not embody its own extra recovery method for transmission failures. Thus, if a network failure interrupts the streaming, a large file being reconstructed on the CBB end could be left incomplete. In the event the device being captured also goes offline, then the evidence may be irrecoverably lost in this Client-Server scenario.

It should be noted, though, that storage failures were dealt with through the replication afforded by Hadoop HDFS that was managing the CBB backend filesystem. Thus, some form of availability and thus reliability was achieved, though only for evidence data successfully saved in storage at least once.

From the results gathered in the test experiments we discover that larger partition sizes may not scale well in remote evidence acquisition depending on the capabilities of the device need. Additionally we do see that devices with better hardware could scale well with data sizes that proved problematic for less capable devices. Furthermore we see that for large partition sizes the remote acquisition times do increase to significantly long times (over 4hrs for >14GB), thus the likelihood of a failure occurring during these large durations increases. We postulate that the use of P2P networks, between the evidence capture location and the eventual storage location, could be used to assist in providing availability through replication of data at multiple peers, as well as better throughput through using the bandwidth of multiple peers to facilitate uploading of potential evidentiary data. Certain P2P overlays such as the BitTorrent protocol are known to provide better network throughput, and thus shorter latency times between evidence capture and storage. Others are known to provide high availability through replication of data at multiple peers thus reducing the problem of a single point of failure that may be experienced with mobile devices having intermittent network connectivity. This could also potentially increase the time-window in which evidence can be gathered aiding in the need for collecting as much evidence as possible in digital investigations.

Smaller file sizes being transmitted over the network are seen to have a higher likelihood of succeeding, thus splitting up the larger partitions into smaller pieces before transmission may also help reduce failures. This is also a common trend in P2P overlay networks such as BitTorrent where large files are split into smaller pieces, thus furthering our hypothesis of using P2P overlays to help improve availability and network throughput. Another alternative paradigm that could assist in improving reliability of remote evidence capture would be to collect and transmit metadata of the activities occurring on various evidence sources (disk, memory or network) rather than transferring the entire evidence source. The metadata repositories are likely to be smaller in size and thus transferring and replicating these among peers during the remote evidence capture process, could prove to be more efficient through promoting the reliability seen in the transfer of smaller files.

In determining the success rates of the remote evidence acquisition, it should be mentioned that only 10 trial runs were done per partition size because the evidence acquisition times were getting significantly longer. More than 10 trials proved to be inconvenient and a hindrance given the time constraints in performing this study. Ten trials were deemed to be good enough to show some form of indicative result, even though the value of 10 may not represent a statistically significant sample size to generalize the results as being fully representative of the phenomenon being analyzed.

7. CONCLUSION

In this study we outlined the numerous problems that blight the digital investigation process, and law enforcement agencies at large, rendering them slow and ultimately ineffective. We proposed a comprehensive architecture of a proactive, system – that makes used of hypervisors, P2P networks, the RDF framework and cloud storage – that could help improve the digital investigation process through automation. Through a small proof of concept, we demonstrate a limited part of this system showing remote evidence acquisition of small scale devices over public networks. From the results of testing this proof of concept, we motivate the need for a network paradigm that enables more reliability and better throughput of the network transfer process. Some P2P overlays, such as the BitTorrent protocol demonstrate these characteristics and could possibly provide the solution to improving the speed and reliability of remote evidence capture.

8. FUTURE WORK

Though this architecture is promising, larger disk acquisitions need to be performed with more modern small scale devices that are equipped with larger storage capacities in order to further confirm the need for a more efficient and reliable form of network data transfer in the form of P2P communication. From the proposed architecture, several parameters within the P2P communication protocols need further optimization and testing. Additionally, a PKI infrastructure can be infused in the system in order to improve the security of the communication and storage facilities. Also, the storage capabilities of the Cloud-based Backend could be supplemented by that of participating HbH nodes in order to realize a more distributed, decentralized and independent storage solution. The concept of privacy also needs to be addressed within the scope of this solution. Finally, an experiment with a wider scope, in terms of multiple devices being tested simultaneously, would be greatly desired in order to better drive this architecture towards becoming a reality.

REFERENCES

AccessData. (2014). AccessData FTK ADEnterprise. Retrieved December 4, 2014, from http://accessdata.com/solutions/digital-forensics/ad-enterprise

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*, *3*, 50–58. doi:10.1016/j.diin.2006.06.016

Almulhem, A., & Traore, I. (2005). *Experience with Engineering a Network Forensics System. Proceedings of the 2005 international conference on Information Networking. Convergence in Broadband and Mobile Networking*. Korea: Springer Berlin Heidelberg.

Case, A., Cristina, A., Marziale, L., Richard, G. G., & Roussev, V. (2008). FACE: Automated digital evidence discovery and correlation. *Digital Investigation*, *5*, S65–S75. doi:10.1016/j.diin.2008.05.008

CDESF Working Group. (2006). Standardizing digital evidence storage. *Communications of the ACM*. doi:10.1145/1113034.1113071

Cohen, B. (2003). Incentives build robustness in BitTorrent. *Workshop on Economics of Peer-to-Peer Systems*. Retrieved from http://www.ittc.ku.edu/~niehaus/classes/750-s06/documents/BT-description.pdf

Cohen, M., Garfinkel, S., & Schatz, B. (2009). Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *Digital Investigation*, *6*, S57–S68. doi:10.1016/j.diin.2009.06.010

Cohen, M. I., Bilby, D., & Caronni, G. (2011). Distributed forensics and incident response in the enterprise. In *Digital Investigation* (Vol. 8, pp. S101–S110). Elsevier Ltd. doi:10.1016/j.diin.2011.05.012

Davis, M., Manes, G., & Shenoi, S. (2005). A network-based architecture for storing digital evidence. *Advances in Digital Forensics: IFIP International Conference on Digital Forensics*, *194*, 33–42. doi:10.1007/0-387-31163-7_3

Dean, J., & Ghemawat, S. (2008). MapReduce : Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*(1), 1–13. doi:10.1145/1327452.1327492

Dosis, S., Homem, I., & Popov, O. (2013). Semantic Representation and Integration of Digital Evidence. *Procedia Computer Science*, *22*, 1266–1275. doi:10.1016/j.procs.2013.09.214

Garfinkel, S. L. (2006). AFF : A New Format for Storing Hard Drive Images. *Association for Computing Machinery. Communications of the ACM*, *49*(2), 85–87.

Guidance Software. (2014). Encase Enterprise. Retrieved December 4, 2014, from https://www.guidancesoftware.com/products/Pages/encase-enterprise/overview.aspx

Homem, I. (2013). *LEIA : The Live Evidence Information Aggregator A Scalable Distributed Hypervisor-based Peer-2-Peer Aggregator of Information for Cyber- Law Enforcement*. KTH - The Royal Insitute of Technology.

Homem, I., Dosis, S., & Popov, O. (2013). LEIA: The Live Evidence Information Aggregator: Towards efficient cyber-law enforcement. In *World Congress on Internet Security (WorldCIS-2013)* (pp. 156–161). London. doi:10.1109/WorldCIS.2013.6751038

Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., & Steen, M. van. (2007). Gossip-based peer sampling. *ACM Transactions on Computer Systems (TOCS)*, *25*(3), 1–36. Retrieved from http://dl.acm.org/citation.cfm?id=1275520

Kahvedžić, D., & Kechadi, T. (2009). DIALOG: A framework for modeling, analysis and reuse of digital forensic knowledge. *Digital Investigation*, *6*, S23–S33. doi:10.1016/j.diin.2009.06.014

Kaspersky Lab. (2014). *The Regin Platform: Nation-State Ownage of GSM Networks.*

Koopmans, M. B., & James, J. I. (2013). Automated network triage. *Digital Investigation*, *10*(2), 129–137. doi:10.1016/j.diin.2013.03.002

Leu, F.-Y. L. F.-Y., & Yang, T.-Y. Y. T.-Y. (2003). A host-based real-time intrusion detection system with data mining and forensic techniques. *IEEE 37th Annual 2003 International Carnahan Conference onSecurity Technology, 2003. Proceedings.*, (Mid). doi:10.1109/CCST.2003.1297623

Moser, A., & Cohen, M. I. (2013). Hunting in the enterprise: Forensic triage and incident response. *Digital Investigation*, *10*(2), 89–98. doi:10.1016/j.diin.2013.03.003

National Institute of Standards and Technology. (2004). Digital data acquisition tool specification. *Draft for Comments*. Retrieved from http://www.cftt.nist.gov/Pub-Draft-1-DDA-Require.pdf

Palmer, G. (2001). A Road Map for Digital Forensic Research. In *Proceedings of the Digital Forensic Research Workshop, 2001.* Uttica, New York.

Raghavan, S., Clark, A., & 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*, *8*, 83–94. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-02312-5_10

Redding, S. (2005). Using Peer-to-Peer Technology for Network Forensics. *Advances in Digital Forensics: IFIP International Federation for Information Processing*, *194*, 141–152. doi:10.1007/0-387-31163-7_12

Ren, W., & Jin, H. (2005). Distributed agent-based real time network intrusion forensics system architecture design. In *Proceedings - International Conference on Advanced Information Networking and Applications, AINA* (Vol. 1, pp. 177–182). Ieee. doi:10.1109/AINA.2005.164

Roussev, V., & Richard III, G. G. (2004). Breaking the Performance Wall: The Case for Distributed Digital Forensics. *Digital Forensics Research Workshop*, 1–16.

Sacha, J., Dowling, J., Cunningham, R., & Meier, R. (2006). Discovery of stable peers in a self-organising peer-to-peer gradient topology. In *International Conference on Distributed Applications and Interoperable Systems (DAIS)* (pp. 70–83). Retrieved from http://link.springer.com/chapter/10.1007/11773887_6

Scanlon, M., Farina, J., Khac, N. A. Le, & Kechadi, T. (2014). Leveraging Decentralization to Extend the Digital Evidence Acquisition Window : Case Study on BitTorrent Sync. *Journal of Digital Forensics Security and Law*, *9*(December), 85–99.

Scanlon, M., & Kechadi, M. T. (2010). Online acquisition of digital forensic evidence. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering* (Vol. 31 LNICST, pp. 122–131). doi:10.1007/978-3-642-11534-9_12

Schatz, B., & Clark, A. (2006). An open architecture for digital evidence integration. In *AusCERT Asia Pacific Information Technology Security Conference* (pp. 15–29). Gold Coast, Queensland. Retrieved from http://eprints.qut.edu.au/21119/

Scientific Working Group on Digital Evidence (SWGDE). (2006). Data integrity within computer forensics. Retrieved from https://www.swgde.org/documents/Current Documents/2006-04-12 SWGDE Data Integrity Within Computer Forensics v1.0

Shields, C., Frieder, O., & Maloof, M. (2011). A system for the proactive, continuous, and efficient collection of digital forensic evidence. In *Digital Investigation* (Vol. 8, pp. S3–S13). Elsevier Ltd. doi:10.1016/j.diin.2011.05.002

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. doi:10.1109/MSST.2010.5496972

sKyWIper Analysis Team. (2012). *Skywiper (a.K.a Flame a.K.a Flamer): a Complex Malware for Targeted Attacks* (Vol. 05). Budapest. Retrieved from http://www.crysys.hu/skywiper/skywiper.pdf\npapers2://publication/uuid/1A396077-EBAB-47F8-A363-162BDAF34247

Stone-Gross, B. (2012). *The Lifecycle of Peer-to-Peer (Gameover) ZeuS*. Retrieved from http://www.secureworks.com/cyber-threat-intelligence/threats/The_Lifecycle_of_Peer_to_Peer_Gameover_ZeuS/

Van Baar, R. B., van Beek, H. M. a., & van Eijk, E. J. (2014). Digital Forensics as a Service: A game changer. *Digital Investigation*, *11*, S54–S62. doi:10.1016/j.diin.2014.03.007

Yu, J., Ramana Reddy, Y. V., Selliah, S., Reddy, S., Bharadwaj, V., & Kankanahalli, S. (2005). TRINETR: An architecture for collaborative intrusion detection and knowledge-based alert evaluation. *Advanced Engineering Informatics*, *19*(2), 93–101. doi:10.1016/j.aei.2005.05.004

Zonouz, S., Joshi, K., & Sanders, W. (2011). Floguard: cost-aware systemwide intrusion defense via online forensics and on-demand IDS deployment. In *Computer Safety, Reliability, and ...* (pp. 338–354). Naples, Italy: Springer-Verlag, Berlin, Heidelberg. doi:10.1007/978-3-642-24270-0_25