Proceedings of the Sixth International
Workshop on Digital Forensics & Incident Analysis (WDFIA 2011)

A Signature Detection Scheme for Distributed Storage

R. Hegarty, M. Merabti, Q. Shi and R. Askwith

School of Computing and Mathematical Sciences, Liverpool John Moores University, James Parsons Building, Byrom Street, Liverpool, L3 3AF, U.K.
R.C.Hegarty@2006.ljmu.ac.uk, {M.Merabti, Q.Shi, R.J.Askwith}@ljmu.ac.uk

Abstract

Cloud computing is an emerging model of computing that offers elastic scalable computing resources to many concurrent users worldwide. It provides resources that are paid for as they are consumed, dynamically scaled to suit the demands of the user, which makes it attractive to organisations that wish to consolidate resources by creating their own elastic resource platforms or outsource to obtain more flexible cost effective computing resources. The scale and dynamic nature of cloud computing creates significant challenges for their management, including investigating malicious activity and/or policy failure. Digital forensics is the practice of analysing computers for evidence of crime or breach of policy. Among the various techniques employed to forensically analyse computer systems, file signatures are commonly used. This paper identifies the barriers to applying existing signature detection techniques to the large scale distributed storage platforms provided by cloud computing. The focus of this paper is the development of a model to determine a suitable signature length for use in the forensic analysis of a large distributed set of files. By reducing the signature length we show that we can reduce the amount of data required to carry out signature detection as this is one of the constraints preventing exiting techniques from being applied to cloud platforms. Through experimentation we validate our model and show that it is possible to use shorter length signatures to accurately carry out forensic analysis if factors such as the scale of the data undergoing analysis and the scale of the signature set used for the analysis are taken into account.

Keywords

Digital forensics, cloud computing, signature detection

1. Introduction

The amount of data stored in distributed systems is growing as they move from being the domain of scientists (Huang et al. 2006) to the domain of everyday users via cloud computing and utility computing platforms (Foster et al. 2008). Signature detection is the process of comparing a set of known signatures against signatures generated from a set of files to determine if any matches are present, it entails analysis of vast repositories of data to detect the presence of known illicit files (Richard III & Roussev 2006). Current digital forensic signature detection techniques were not developed to target distributed environments and therefore do not possess the capability to process data and signatures at cloud scale. Cloud scale systems contain Exabyte's of distributed data negating the ability of a single computer to analyse the files stored within them. Therefore a distributed approach is required. The signature sets which play a key role in forensic analysis must therefore be

distributed across multiple analysis nodes within the distributed forensic analysis system. As these signature sets are themselves large in scale often containing many millions of signatures their deployment introduces storage, computation and network overheads. As the scale of the data undergoing analysis increases so too does the number of analysis nodes required to carry out signature detection. This in turn increases the signature set deployment overheads' as it increases the duplication of the signature set. To make feasible the scalable distribution of signature sets a more efficient signature creation scheme is required to reduce the burden of signature distribution placed on cloud infrastructure to make signature detection across a distributed forensic analysis system feasible.

This paper proposes a model to determine what length signature is most appropriate for use in a signature detection scheme applied to distributed storage. With the goal of minimising the burden of signature dissemination, to enable a distributed signature detection process to be developed. Section 2 provides background on cloud computing and the existing computer forensic techniques that are applied both to analyse distributed environments and utilise distributed environments to analyse single computers. Section 3 provides details of our models for signature creation and signature detection accuracy when various length signatures are used in the signature detection process. Section 4 provides details of the experiments we carried out to and result we obtained to verify our models. Section 5 contains a conclusion about the effectiveness of our approach and introduces the future work we have identified as necessary in order to enable forensic analysis of cloud computing environments.

2. Related Work

Cloud computing is a computing paradigm where users request and consume computer resources as and when they require them (Armbrust et al. 2009) in much the same way as utility companies provide access to water, electricity and gas on demand (Foster et al. 2008) (Armbrust et al. 2009). Through a combination of distributed computing and virtualisation (Barham et al. 2003) techniques resources are allocated in an elastic manner scaling up and down to suit demand (Rappa 2010). The three broad categories of cloud computing are (Rimal et al. 2009) Infrastructure as a Service (IaaS), Platforms as a Service (PaaS) and Software as a Service (SaaS). IaaS provides users with hardware and software as a service, and its resources (processing power, memory, and storage) are dynamically scalable, enabling adaptation to suit the users changing requirements and providing cost savings (Delic & Walker 2008). PaaS provides an environment in which developers can create distributed scalable applications using the API's provided by the service provider. SaaS provides users with applications hosted on a distributed architecture to be accessed using a Web browser. This enables developers to create cross platform applications and for end users to access applications from any computer with a Web browser. All of these models provide storage as a service in some form or other, storage is also offered as a service individually by providers such as Amazon via their S3 service (Amazon n.d.) and Dropbox (Dropbox n.d.) a storage service for end users. The detection of illicit data stored in these large scale storage platforms is the target of the forensic analysis techniques proposed in this paper.

While cloud computing is efficient (Delic & Walker 2008) and flexible it introduces barriers to current digital forensic analysis techniques, which typically rely on having physical access to a data storage device in order to image (create a bit by bit copy of the device) (Allen 2005) and search the device for known illicit files. Storage device access is restricted in cloud computing as the distributed storage often spans multiple data centres worldwide and is used to store multiple concurrent users' data (Reilly et al. 2010).

Various authors have identified the potential to utilise distributed computing platforms to perform forensic analysis or store images (Garfinkel 2007). In (Golden G. Richard, Vassil 2006) (Roussev & Richard III 2004) the authors identify the need to apply distributed analysis techniques to the task of analysing computer hard drives. They cite the complexity of investigations and ever increasing storage capacities as barriers to existing standalone techniques. They also identify the features of forensic analysis techniques, which can benefit from informed design choices such as avoiding unnecessary memory-to-memory copies and disk I/O particularly writes (Golden G. Richard, Vassil 2006). However ultimately they posit that the performance gains alone from well designed software will not be sufficient and a distributed approach is required.

In (Golden G. Richard, Vassil 2006) the acquisition phase of forensic analysis is identified as an area which needs improvement as capturing all possible data is costly (Golden G. Richard, Vassil 2006). A system to load entire disk images into main memory, distributed across multiple worker nodes and coordinated by a central control node is proposed in (Golden G. Richard, Vassil 2006). This allows the disk image to be read into memory once for subsequent analysis by multiple worker nodes reducing the I/O overheads imposed through multiple reads. The author also criticises the current sequential query/response/query model used by current tools for their lack of concurrency. While the paper successfully identifies the requirements for a distributed processing technique it does not implement and test any techniques nor does it address the emerging trend of large-scale distributed storage on the Internet often referred to as cloud computing.

A distributed forensic analysis architecture proposed by (Liebrock et al. 2007) utilises a database wrapper to enable workstations to request the results of a distributed analysis technique from a parallel machine. The distributed analysis technique is not described and the overall architecture appears to be simplistic. They do not consider the requirements identified by previous work in this field.

The Forweb technique proposed and tested in (Haggerty et al. 2008) utilises a more efficient block signature search technique Forsigs (Haggerty & Mark Taylor 2006) to analyse blocks from images retrieved from the Internet by a web spider. The approach was developed to analyse image files uploaded and stored in distributed storage platforms accessible via the World Wide Web using a single analysis host to target specific websites. While the technique results in accurate signature detection when searching compressed file types it is not accurate when searching for non-

compressed file types and does not scale due to its reliance on a single host for analysis.

3. Distributed Forensic Analysis

The existing work cited in the previous section identifies either the requirement to distribute the forensic analysis of single host machines or the requirement to analyse data stored in distributed platforms. However, to our knowledge to date no attempt has been made to bring together these requirements to enable distributed forensic analysis of the large scale distributed data repositories found in cloud computing. To achieve these goals we need to develop a distributed signature detection technique. This paper focuses on the first stage of this task, which is the creation of signatures with a low storage overhead for use in a distributed signature detection process. We make the following assumption about the overall system to enable this work to be carried out. We assume that multiple analysis nodes will be created within the cloud environment, each of which will receive a copy of the signature set being used to carry out the first round signature detection. These analysis nodes will receive the reduced length signature sets which are the subject of this paper and carry out comparison with signatures they create from their local subset of files. Any matches in the first round stage will result in a second round search being carried out using the corresponding full length MD5 hash values retrieved following the first round search to confirm or deny that a file signature matches. The overall aim being to reduce the quantity of data required to store and distribute the signature sets needed for forensic analysis by reducing the storage requirement per signature for the first round and reducing the number of MD5 hash values required for the second round.

Our work aims to provide a set of metrics for use in signature detection to determine the most suitable length signatures to use for signature detection based on any given set of criteria. The criteria include the availability of time, required accuracy and computational resources available. By creating and disseminating reduced length signatures from known target files forensic analysis of large distributed storage platforms can be carried out. MD5 signatures for a set of target files may be hundreds of megabytes so significantly reducing the size of these enables a distributed approach to signature detection where many analysis nodes carry out signature detection simultaneously resulting in an approximately linear reduction in the amount of time required to carry out analysis.

3.1. Signature Creation and Detection Models

Our model can determine how many signatures a signature set will contain. By taking into account the number of hashes the signatures are created from and the length of the signatures in the set. The purpose of a signature is to provide pseudo unique identification of the file or hash value they were generated from. We calculate the theoretical number of signatures that a theoretical evenly distributed signature set would hold if we input x hash values and compressed them to length m using equation 1.

Equation 1 sums the probability of each of the hash values being input resulting in a unique signature. This is determined by calculating the probability that each signature will be unique. The probability of a signature being unique is reduced as signatures are added to the set. To calculate the probability of a signature already being in the set we divide the number of signatures in the set by the capacity of the set. To determine the probability of the signature not being in the set we divide 1 over the probability that the signature is already in the set. The capacity of the set is determined by the length of the signatures. As each bit in the signature can represent 2 values the capacity of the set is 2^m. If s calculated in equation 1 is greater than the sets capacity we substitute s with the set's capacity. Thus we can determine how many unique signatures are theoretically held in the set of signatures of a given length.

$$s = \left(\frac{x}{\sum_{i=1}^{z} \frac{1}{i}}\right) 2^m$$

Equation 1. Probable number of signatures created

$$r = \frac{s}{2^m}$$

Equation 2. Probable false positive rate

$$t = (128fr) + ms$$

Equation 3. Total amount of search data for both search rounds

Symbol	Meaning
s	Number of signatures in the set
x	Number of hash values input
m	Length of the signatures being created
z	The smaller of x and 2^m
r	False positive rate as a %
f	Number of files undergoing analysis
t	Total amount of search data for both search rounds.

We analyse the impact of reduced length signatures on the accuracy of a search using reduced length signatures. The probability of a file signature matching a signature in the signature set is directly related to the percentage of the signature set that is occupied by signatures. Equation 2 of our model illustrates this relationship.

We combine both the signature creation and signature detection equations to create equation 3 which calculates the total amount of data required to represent the signatures in the first round signature set. This is added to the total amount of data required to represent the 128bit MD5 hash values used in the second round search which are required to rule out the false positives induced through the use of the shorter length signatures in the first round search.

4. Results

To verify our signature creation and detection models we employed two signature creation techniques. The first technique SPLIT takes the first *n* bits of a MD5 hash value and uses them as a signature. The second XOR technique splits an MD5 hash value into a even number of pieces and combines them using bitwise exclusive or to create a signature *n* bits in length. Both techniques where applied to sets of randomly generated unique MD5 hash values to create signature sets for the various signature lengths.

4.1. Signature Creation

We created a Python script to create and calculate the total number of signatures in each set using the SPLIT and XOR techniques to create various length signatures. The script generated 10,000 MD5 hash values and processed them to create the signature sets. The script was executed 100 times and the average number of signatures in each signature set recorded for signatures of length 4 – 40 bits. Both of the SPLIT and XOR techniques resulted in very similar results to our theoretical model as illustrated in table 1 and figure 1.

	4	8	12	16	20	24	28	32	36	40
Model	16	256	2744	9216	9952	9997	9999	9999	9999	9999
XOR	16	256	3732	9266	9951	9996	9999	10000	10000	10000
SPLIT	16	256	3720	9273	9958	9994	10000	10000	10000	10000

Table 1: Actual and theoretical signature set size

The actual results largely follow the trend of the theoretical model. Figure 1 compares the average of the actual results obtained from the XOR and SPLIT techniques with the theoretical results from our model. Initially the results are identical as the number of signatures in the signature sets is equal to their capacity. For signature sets containing signatures of lengths from 10 bits to 16 bits, there is some divergence between the theoretical and actual results. This divergence is expected as our mathematical model assumes perfectly even byte value distribution for the signatures at each length.

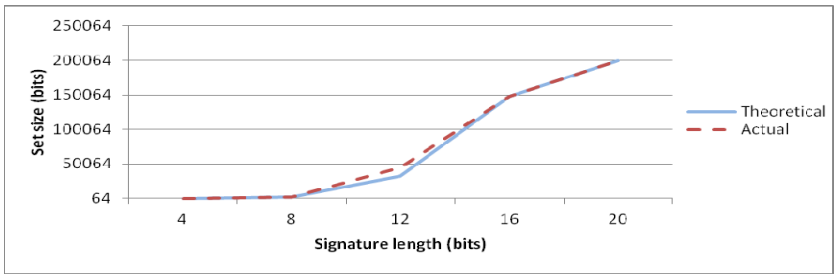

Figure 1: A comparison of the actual and theoretical signature set sizes

This is of course not the case in reality. The MD5 hash values input into the signature creation techniques have near even byte value distribution across their length. However the signature creation techniques create a shorter signature which may contain byte values which are less evenly distributed across the length of the signature. The effect of any skew in byte value distribution is not observed when the signature set is full to capacity. Likewise when the capacity of the signature set is much larger than the number of signatures in the signature set the effect is not observed as the probability of the byte value distribution skew resulting in a collision where two or more difference MD5 hash values are processed resulting in the same reduced length signature is much smaller. For signatures of length 20 bits – 40 bits the results from our experiments converge with those of our model and follow a linear trend.

4.2. Signature Detection

We created a Python script which generated two distinct sets of hash values. The first set represented the signature set and contains 10,000 hash values. The second set represents the files undergoing analysis and contains 100,000 hash values. Using the XOR and SPLIT techniques the hash values were used to create signature sets containing signatures of lengths 4,8,12...40. For each length and type of signature, comparison was carried out between the sets to determine the number of matches. As the initial hash values generated by the script formed two distinct sets containing 10,000 and 100,000 unique hash values we expect zero matches to be found therefore any matches between the sets where viewed as false positives. We executed the script 100 times and calculated the average number of false positives for each signature type at various lengths.

Signature Length	4 bits	8 bits	12 bits	16 bits	20 bits	24 bits	28 bits	32 bits	36 bits	40 bits
Theoretical	100%	100%	67.0%	14.1%	0.95%	0.06%	0.004%	0.0002%	<0.0002%	<0.0002%
XOR	100%	100.%	91.6%	14.1%	0.94%	0.06%	0.003%	0.0002%	<0.0002%	<0.0002%
SPLIT	100.%	100%	91.8%	14.2%	0.95%	0.06%	0.004%	0.0002%	<0.0000%	<0.0000%

Table 2: False positive rates when searching for 10,000 signatures in 100,000 signatures.

Table 2 illustrates the percentage false positive rate for signatures of length 4 to 40 bits determined by our model and those generated by our experiment. The results more or less follow the trend determined by our model. The false positive rate was initially 100% as the signature sets containing 4 and 8 bit signatures are filled to 100% capacity by signatures. This results in 100% of the file signatures matching signatures in the set. As the percentage of signatures in the signature sets is reduced due to the increased signature length the percentage of false positives also falls. In Figure 2 we compare the results of our experiment with our model, for signature lengths 4 – 24 bits. The false positive rates between 24 and 40 bits are either identical to the graph generated by our model in figure 2 or show negligible difference. As the results produced by the XOR and SPLIT techniques are almost identical they are represented on the graph by a single line which is the average of the two.

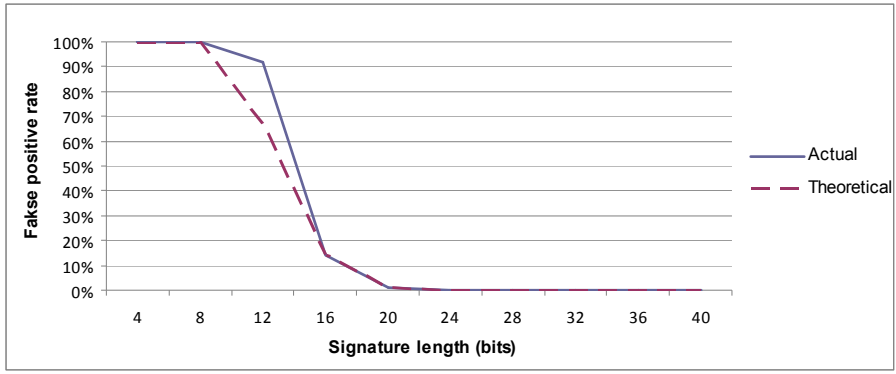

Figure 2: Difference between modelled and actual false positive rates.

The marked difference between the modelled and actual results for signature lengths 10 – 16bits is not unexpected. As explained in the previous section the possible skew in the byte value distribution induced by the SPLIT and XOR signature creation techniques is reflected in the false positive rates for signatures at these lengths. When the signature length is 16 bits or greater the signatures are more evenly distributed across the set and occupy a much smaller percentage of the signature sets capacity making collisions and therefore false positives much less probable. The results of our experiment for signatures with lengths 24 – 40 bits closely match our model with the false positive rate rapidly decaying to a small fraction of a percent.

Our model can fairly accurately determine the size of a signature set when a given length signature is used. It can also determine the accuracy of a signature detection process which utilises the signatures. This is useful as it allows decisions to be made about what length signature to use when analysing a set of files.

Using equations 1 and 2 from our model we can compare determine the false positive rate of reduced length signatures and the amount of round one search data required to carry out signature detection. Figure 3 shows the relationship between signature length and accuracy.

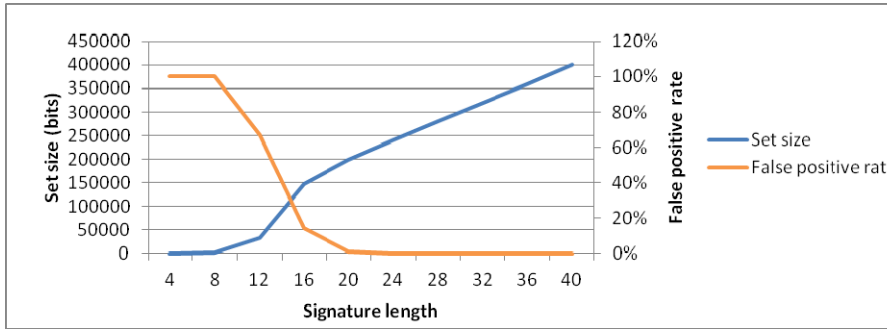

Figure 3: Comparison of signature set size and accuracy for various length round one signatures

This satisfies the requirement to reduce the size of the signature set used for forensic analysis of a distributed storage platform and enables us to determine the accuracy levels likely to be achieved if we use signatures of a given length. This allows tradeoffs to be made when determining which length signature to use for the first round of a signature detection technique, taking into account the size of the resulting signature set and the potential burden it will impose of the signature detection process.

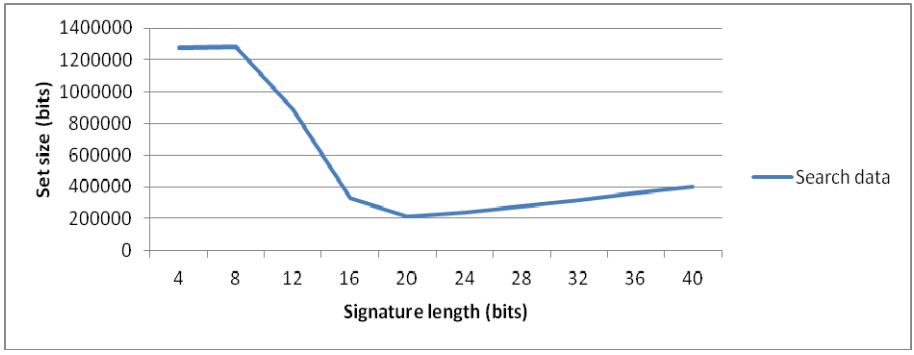

Figure 4. Total amount of data required to achieve accuracy comparable to existing techniques.

Using equation 3 from our model we calculate the total amount of data required to represent the signatures for both the first and second round searches. Taking into account the number of round two MD5 hash values that are required to rule out any false positives induced through the use of shorter signatures in the first round. The results are graphed in figure 4 which illustrate how we can achieve accuracy comparable with that of a single round search using MD5 hash values while reducing the combined size of the signature sets using our model for a two round search. In the case of our experiment using 24 bit signatures in the first round search provides a reduction in the overall signature set size from 1280000 to 240689 an 81% reduction. Obviously if files with matching signatures in the first round also match signatures in the second round there will be an increase in the size of the required search data.

5. Conclusion and Further Work

Reduced length signatures can be used to reduce the scale of signature sets and make feasible forensic analysis of large scale distributed storage platforms. We have considered the scale of the signature set and the scale of the data undergoing analysis as factors that influence the selection of signature lengths when analysing data. We can now determine an appropriate signature length for a given sized signature set and number of files. This enables us to avoid two negative situations which could occur if incorrect signature lengths where selected for use in the analysis process. The first of situation may occur if the signature length selected was too short, and the result would be many false positives in the first round of the search. Resulting in the requirement for the vast majority of the hash values generated for the round two of the search to be disseminated. This would increase the burden on both the network

and analysis nodes, reducing the ability of the analysis technique to scale. The second situation arises where the signatures selected were too long. This would again result in an increased burden on the network and analysis nodes and limit the ability of the analysis technique to scale.

While there are existing techniques such as Bloom filters (Bloom 1970) available which provide time and space efficient searches, the nature of digital forensics is such that we need to determine which signature matches rather than just that a match has occurred. Our technique provides this and enables us to look up the metadata associated with the signature to continue the investigation. For example we may want to categorise our signatures based on a set of criteria or know when a signature was first detected to enable link analysis to be carried out.

We will run further experiments using a distributed network of analysis nodes to enable us to determine the effect of distributing our signatures and take into account factors such as the availability of bandwidth and other constraints when determining an appropriate signature length to enable large scale distributed analysis.

To distribute our signature detection scheme we are working on a publish subscribe model where analysis nodes can register with a server to receive signature sets as well as command and control information to initialise, execute and report the results of our distributed signature analysis technique. When a node registers it receives a copy of the reduced size round one signature set. The files on or local to the node are processed to produce reduced size signatures which are compared with those in the set to determine if a match can be found. If a match is found, the details of the file are logged and its second round signature requested. The second round search will be executed using the MD5 hash values which correspond to the round one signatures which resulted in a match.

6. References

Allen, W.H., 2005. Computer forensics. *Security & Privacy Magazine, IEEE*, 3(4), pp.59-62.

Amazon, Amazon Simple Storage Service (Amazon S3). *http://aws.amazon.com*. Available at: http://aws.amazon.com.

Armbrust, M. et al., 2009. *Above the clouds: A berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*.

Barham, P. et al., 2003. Xen and the art of virtualization. In *OSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles*. Bolton Landing, NY, USA, p. 164.

Bloom, B., 1970. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7), p.422–426.

Delic, K. a & Walker, M.A., 2008. Emergence of the Academic Computing Clouds. *ACM Ubiquity*, 9(21).

Dropbox. *http://www.dropbox.com/*. Available at: http://www.dropbox.com/.

Foster, I. et al., 2008. Cloud Computing and Grid Computing 360-Degree Compared. *Grid Computing Environments Workshop, 2008. GCE '08*, pp.1-10.

Garfinkel, S., 2007. Commodity grid computing with amazon s3 and ec2. *Usenix*, pp.7-13..

Golden G. Richard, Vassil, R., 2006. Next-generation digital forensics. *Communications of the ACM*, 49(2), pp.76 - 80.

Haggerty, J., Llewellyn-Jones, D. & Taylor, M., 2008. Forweb: file fingerprinting for automated network forensics investigations. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*. Adelaide, Australia: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), p. 29..

Haggerty, J. & Taylor, Mark, 2006. "FORSIGS: Forensic Signature Analysis of Hard Drive Multimedia File Fingerprints." *FIP TC11 International Information Security Conference* (Vol. 232). Sandton, South Africa.

Huang, W. et al., 2006. A case for high performance computing with virtual machines. In *Proceedings of the 20th annual international conference on Supercomputing - ICS '06*. Cairns, Queensland, Australia: ACM Press, p. 125.

Liebrock, L.M. et al., 2007. A preliminary design for digital forensics analysis of terabyte size data sets. *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, p.190.

Rappa, M., 2010. The utility business model and the future of computing services. *IBM Systems Journal*, 43(1), pp.32-42.

Reilly, D., Wren, C. & Berry, T., 2010. Cloud computing: Forensic challenges for law enforcement. In *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*. London, UK, pp. 1-7.

Richard III, G.G. & Roussev, V., 2006. Digital forensics tools: the next generation. *Communications of the ACM*, p.75.

Rimal, B.P., Choi, E. & Lumb, I., 2009. A Taxonomy and Survey of Cloud Computing Systems. In *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*. Washington, DC, USA: IEEE Computer Society, pp. 44-51.

Roussev, V. & Richard III, G.G., 2004. Breaking the performance wall: The case for distributed digital forensics. In *Proceedings of the 2004 digital forensics research workshop (DFRWS 2004)*. DFRWS, pp. 1-16.