Scalable Distributed Signature Detection

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

PROTECT Research Centre for Critical Infrastructure Computer Technology and Protection, School of Computing and Mathematical Sciences, Liverpool John Moores University, James Parsons Building, Byrom Street, Liverpool, L3 3AF,U.K.
e-mail: R.C.Hegarty@2006.ljmu.ac.uk, {M.Merabti, Q.Shi, R.J.Askwith}@ljmu.ac.uk

Abstract

Storage as a Service (SaS) platforms provide users with a convenient and cost effective way to store and share data. The scale and distribution of data in SaS is such that existing signature detection techniques are not suited to the task of analysing data in these platforms. To maintain a practical and effective digital forensic capability, a new approach to the detection of target data in such platforms is required. This paper analyses the potential impact of the widespread use of SaS in particular object storage platforms, has on digital forensic investigations and identifies the key challenges to be overcome. The focus of the paper is the development of a model to distribute the signature detection process in a way that minimises the quantity of resources required to carry out signature detection while at the same time maintaining the accuracy of current techniques and achieving signature detection within appropriate temporal boundaries.

Keywords

Signature detection, Digital forensics, Object storage, Cloud computing

1 Introduction

There has been an explosive growth in the quantity of data stored in SaS platforms. Data from both fixed and mobile devices is routinely stored in "the cloud". Along with the vast quantity of data (Osborne & Slay, 2011) another characteristic of SaS is the distribution of data across many different storage devices. Distribution provides resilience and the ability to recover data in the event of a device failure. Unfortunately the existing image, analyse, present (Grobler, Louwrens, & von Solms, 2010) paradigm of digital forensics is not well equipped to process large scale distributed data (F. Anwar & Anwar, 2011). In particular, the automated signature detection process where each file on a storage device has its hash value computed and compared with a local signature library is infeasible in a distributed environment.

From a practical perspective, it is not possible to image the many thousands of devices, which make up a SaS platform. Ethically imaging these devices in a multi-tenancy environment (Naqvi, Dallons, & Ponsard, 2010) would also be questionable (Burd, Jones, & Seazzu, 2011). Another approach to the preservation, analysis and

representation of evidence is required. Fortunately, SaS environments routinely generate and check hash values to validate the integrity of the data they store. These hash values can be extracted for use in the signature detection process. However, there are still challenges to be overcome, firstly the processing requirements for analysing potentially billions of files with a signature library containing the signatures of known target files makes analysis using a single machine infeasible. Secondly, as the signature libraries used in digital forensics contain many millions of signatures. Distributing the signatures to multiple analysis nodes will become infeasible as the number of nodes required carry out signature detection within reasonable temporal constraints grows.

To achieve the scalability essential to carrying out signature detection in a large distributed environment. A model is required to reduce the burden of signature distribution while maintaining the accuracy of the search process. This paper proposes a novel model to minimise the amount of data required to carry out signature detection using multiple analysis nodes. The layout of the remainder of this paper is as follows, related work is detailed in section two followed by a description of our model in section three. Section four describes the experiments carried out to validate our model and finally section five concludes with our findings and proposes future work.

2 Related Work

Object storage services such as Amazon's Simple Storage Service (S3) described in (Palankar, Iamnitchi, Ripeanu, & Garfinkel, 2008) provide users with a flexible and efficient mechanism to store and share their data. They are examples of a subset of Infrastructure as a Service (Rimal, Choi, & Lumb, 2009) providing the storage component. These types of service have a low barrier to entry often providing a limited amount of free storage. By virtue of the scalable elastic nature of these platforms (Delic & Walker, 2008) a pay as you use model similar to that of utility companies (Foster, Zhao, Raicu, & Lu, 2008) (Armbrust et al., 2009) is offered. The detection of illicit data stored in these large-scale storage platforms is the target of the forensic analysis techniques proposed in this paper.

Current digital forensics techniques require physical access to the storage device(s) under investigation. This prerequisite exists due to the requirement to preserve evidence by imaging the device and carrying out analysis on the image (Allen, 2005). This type of approach is not feasible when analysing large-scale highly distributed storage platforms. The storage and bandwidth requirements for carrying out such a task are unrealistic and the ethicacy (Reilly, Wren, & Berry, 2010) of imaging a repository storing data belonging to many thousands of concurrent users would be questionable.

Much work has been carried out to leverage the resources of cloud computing against the task of forensic analysis (SL Garfinkel, 2007). To overcome the challenges associated with the increasing complexity of cases and capacity of modern storage device (Roussev & Richard III, 2004). The authors (Richard &

Roussev, 2006) identified the acquisition phase of the forensic analysis process as being untenable and proposed a system where multiple worker nodes carry out analysis of an image that has been read into memory once. To reduce the time required for analysis by concurrently analysing the image rather than using the sequential approach employed by conventional techniques.

Some work focussed on extending the capability of existing digital forensic techniques used to analyse storage devices to distributed storage. The Forweb search technique proposed by (Haggerty, Llewellyn-Jones, & Taylor, 2008) retrieved blocks from image files found on the world wide web using a crawler. The blocks were analysed using the Forsigs (Haggerty & Taylor, 2006) technique to determine whether a files signature matched that of a signature in the signature library. The technique was accurate but only when applied to a narrow selection of file types (typically JPG, PDF ,GIF) and the reliance on a single host limited the scalability of this approach.

Work has also been carried out to detect copyright infringing content in content delivery networks (Hui, Yin, & Lin, 2009), (Yin, Hui, Li, Lin, & Zhu, 2012). This work was also content/format specific. Similarly techniques for the forensic analysis of Eucalyptus where proposed by (F. Anwar & Anwar, 2011) with the goal of detecting evidence of an attack. While this work is related to our own, the scope is very different with a focus on auditing rather than automated signature detection.

Our previous approach reduced the requirement to transfer large amounts of data in the analysis process by analysing domain specific enhancements to the search technique. We posited that a two stage search with reduced length signatures could achieve a reduction in the required data transfer. Further analysis indicates that when using our previous scheme (Hegarty, Merabti, Shi, & Askwith, 2011) scalability could be limited if large numbers of stage two searches are required. This is due to the reliance on a single node to provide the stage two signatures or carry out stage two signature detection.

3 Distributed Signature Detection

Due to the distribution and scale of data found in SaS platforms, a distributed signature detection technique is required. To overcome the limited scalability of existing signature detection techniques that rely on a single host computer, there are some challenges to be overcome. The first of which is how to affectively distribute the signature libraries used in the signature detection process efficiently? To overcome this challenge we propose a scheme to reduce the burden imposed through replicating and distributing the signature library to each of the distributed analysis nodes.

The hash values used as signatures in conventional signature detection techniques have the ability to represent a tremendous number of unique files. With MD5(Rivest, R., 1992) capable of representing 2^{128} unique files. The probability of a hash value collision occurring is practically zero. While this collision resistance is necessary

when hash values are used as signatures in a digital forensic investigation. When there is the requirement to distribute many millions of them for use in a distributed signature detection process, the length of such signatures becomes undesirable.

We propose a model in which the hash values used as signatures are partitioned using two separate techniques. Then utilised in a two stage search to identify target files within a distributed storage environment. The model takes into account the number of analysis nodes, number of files undergoing analysis and the number of signatures in the signature library.

The general model we are proposing selects the first *n* bits of a hash value for stage one signature detection and uses the remainder of the hash value as the stage two signature. The stage one signatures are sorted and distributed to each analysis node and the stage two signatures partitioned with each node receiving an equal number of signatures as illustrated by the example in Figure 1.

	Stage One Signatures	Stage Two Signatures	
Prefix	All Nodes		
0-3	00125	10294856745920101010293949 4	Node 1
	10203	9002099200BG191010888819EFE	
	30101	FF901803715018843AB18102937	
4-7	69011	8200103801163682974628910 11	Node 2
	79202	AA91828477391275659102828 34	
	70202	B91828187291847536901827 64B	
8-B	97859	01928385729192B735678190091	Node 3
	A0191	81911837645467891271717190 1	
	B9482	2112132784782B918292813AB98	
C-F	E9101	88812093747629237364911092 9	Node 4
	E9991	12387653519882727161 9BB1982	
	F0101	EEF772181008281719127912799	
	<-20bits->	<-108bits->	

Figure 1: Signature Distribution

The partitioning of the first stage signatures results in a reduction in the signature length at the cost of accuracy. The collision rate increases due to the capacity of the signature set made up of *n* length signatures reducing along with *l* the signature length. We calculate *s* the number of stage one signatures for various signature lengths using our algorithm shown in Equation 1.

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

x = The number of hash values input

z = The smaller of number of initial hash value and 2^{l}

l = Signature length in bits

h = Length of original hash value

Equation 1: Number of Stage One Signatures

With the addition of signatures to the set the probability of a signature being unique falls. To calculate *s* the total number of signatures in the stage one signature set we sum the probability that each of the signatures added to the set is unique. For each stage one signature generated from the signature library *x* we determine the probability of each signature being unique by dividing 1 over *i* the probable number of signatures already in the set. We then multiply the total by 2^l the capacity of the signature set.

$$y = \left(\frac{x}{\sum_{i=1}^{z} \frac{1}{i}} \right) 2^{h-l}$$

x = Number of hash values input

z = The smaller of number of initial hash value and 2^{h-l}

l = Signature length in bits

h = Length of original hash value

Equation 2: Number of Stage Two Signatures

Equation 2 calculates *y* the number of stage two signatures in similar fashion. The capacity of the signature set represented by the final term 2^{h-l} in Equation 2 is calculated by subtracting *l* the stage one signature length from *h* the length of the hash value used in the original signature set.

Using the calculated number of stage one signatures it is possible to determine the rate at which matches will occur, in the stage one search due to the use of a reduced signature length. The stage two search is then used to confirm each match. It should be noted that as the model uses the first *n* bits of the original hash value as the stage one signature no false negatives can occur, only false positives. In Equation 3 we calculate *r* the probable rate of stage one matches by dividing the number of stage one signatures *s* by the capacity of the stage one signature set 2^l.

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

s = Number of stage one signatures

l = Length of stage one signatures

Equation 3: Probable Rate of Stage One Matches

By taking into account the number of stage one and stage two signatures in combination with the probable rate of stage one matches and the number of analysis nodes .We are able to calculate which stage one signature length results in the lowest overall data transfer. If we assume that, each analysis node processes an equal fraction of the files. Our algorithm shown in Equation 4 calculates the total amount of data transferred during signature detection. Using this calculation we can measure the impact adding analysis nodes has on the amount of data required for analysis.

$$d = \left(ls + \left(\frac{(h-l)y}{n} \right) + \left(\left(r\frac{f}{n}(h-l) \right) \left(1 - \frac{1}{n} \right) \right) \right) n$$

l = Length of the stage one signatures

s = The number of stage one signatures

h = Length of original hash value

y = The number of stage two signatures

n = The number of analysis nodes

r = The rate of stage one matches

f = The number of files

Equation 4: Total Data Transfer Required for Analysis

The amount of data required for the first stage search is calculated by multiplying l the signature length by s the number of stage one signatures. As the stage two signature set is evenly distributed across the analysis nodes the amount of data required by each analysis node is approximately equal. The amount of data required for the second stage search at each node is calculated as h the initial hash value length minus l the length of the stage one signatures multiplied by y the number of stage two signatures, divided by n the number of nodes. Combining the quantity of stage one and stage two data required gives the total amount of data required by each node.

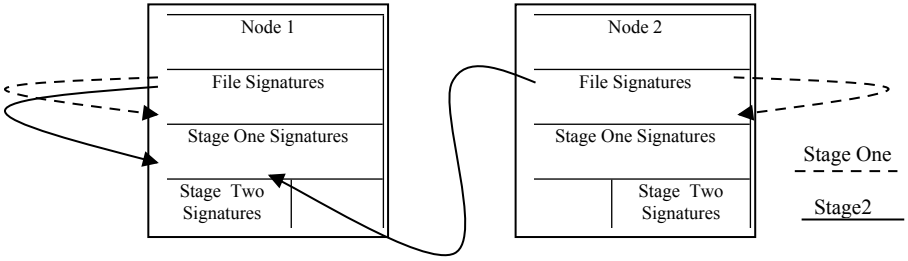

Figure 2: Analysis Node Collaboration

The benefit of the stage two signature set being distributed evenly across the nodes is a reduction in the duplication of data distributed to each of the analysis nodes. The drawback is that further data transfer is required when a stage one match occurs at a node that does not hold the corresponding stage two signatures. This is illustrated by the example in Figure 2 where two nodes carry out analysis, the left hand node does not require collaboration, but the right hand node does resulting in extra data transfer.

To calculate the amount of data transferred between analysis nodes to facilitate stage two signature detection. We calculate the probable number of stage one matches by multiplying r the rate of stage one matches by f/n the number of files at each analysis node. We then factor in the probability that the second stage search must take place at a node other than the node that detected the first stage match. Calculated as 1-1/n

where n is the number of nodes used in the analysis process. Obviously as the number of nodes increases so does the probability that the signatures required for the second stage node will reside at a different node. The additional data transferred per node is added to the previous total amount of data required per node. The result of which is multiplied by n the number of nodes to calculate d the total amount of data transferred during analysis by all nodes.

4 Experimental Results & Analysis

To validate our model we created a Python script, which conducted two stage signature detection using the optimal signature lengths calculated by our model for various numbers of files, signatures and analysis nodes. We then compared the resulting amount of data required for analysis with that predicted by our model illustrated in light grey on each of the graphs.

The application generated two unique sets of MD5 hash values from random data. The first represented the signature library being used in the analysis process and the second the files undergoing analysis at a single analysis node. Stage one and stage two signature sets where created from the signature library. The stage one signatures where created by selecting the first n bits of each MD5 hash value as indicated by our model. The remaining bits of each hash value were used as stage two signatures. The number of stage two signatures was limited to the fraction of the stage two signature set that each analysis node would contain. The total amount of data required for analysis at each node was calculated by measuring the number of elements in the first and second stage signature sets and multiplying them by their respective signature lengths.

Signature detection was carried out by processing each file signature to produce a stage one signature of the length specified by our model. The stage one file signature was compared with the stage one signature set. If a match occurred the corresponding stage two signature file signature was generated and compared with the stage two signature set.

To account for the additional data transfer required when a second stage file signature needed to be transferred to a different analysis node. Each time a second round search was required an additional $\left(1-\frac{1}{n}\right)\times(h-l)$ bits where added to the total amount of data required for analysis. The total amount of data required by all analysis nodes was then calculated by multiplying the total amount of data per node by the number of analysis nodes in use.

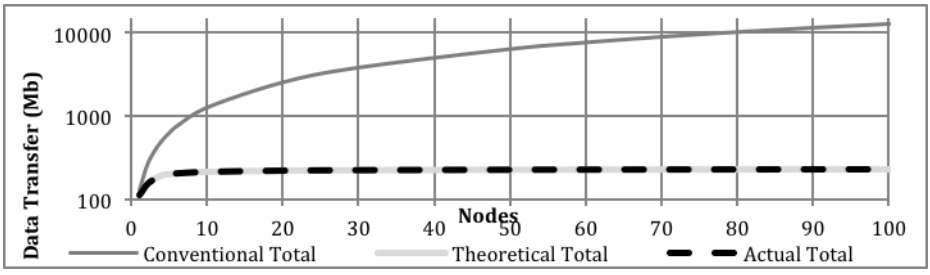

Figure 3: Data Transfer Required for Analysis with a 1:1 Signature to File Ratio

In the first experiment we searched 1 million files for 1 million signatures in order to determine how closely our model reflects the actual outcome and identify a trend in the data. Both the first experiment and subsequent experiments were repeated 100 times to enable an average to be calculated. The results are illustrated in Figure 3 note the logarithmic scale. In each experiment the total amount of data required to distributed the entire MD5 signature set to each node was calculated and used as the conventional total for comparison.

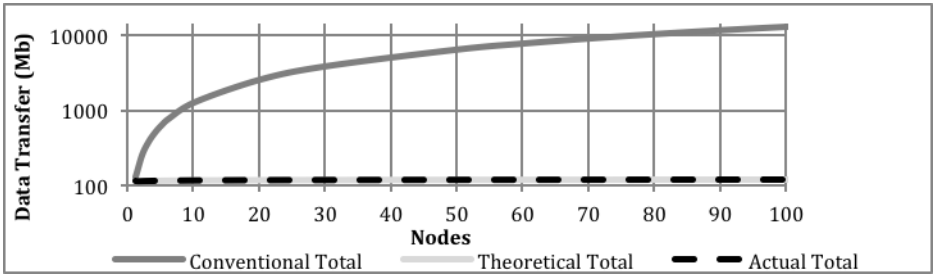

Figure 4: Data Transfer Required with a 1:100 File to Signature Ratio

Two further experiments were carried out using different signature to file ratios to illustrate the impact this variation had on the accuracy and scalability of our model. Figure 4 illustrates the total amount of data transferred when 10,000 files were compared with 1 million signatures. The result was a much flatter curve, with the total amount of data required quickly reaching a plateau where the addition of more analysis nodes resulted in a negligible increase in the amount of data required for analysis. The converse was true when the file to signature ratio was increased as illustrated by Figure 5. The curve took much longer to reach a plateau indicating that the addition of analysis nodes drove up the amount of data required more sharply. This was due to the increased number of files being analysed at each node leading to the requirement to use longer stage one signatures, to reduce the false positive rate of the first stage search. Resulting in an increase in the quantity of data transferred to each node. These signatures where duplicated in the distribution process requiring a large amount of data transfer making the curve more pronounced. The curve

eventually flattens with the addition of more analysis nodes, reducing the required length of the stage one signatures.

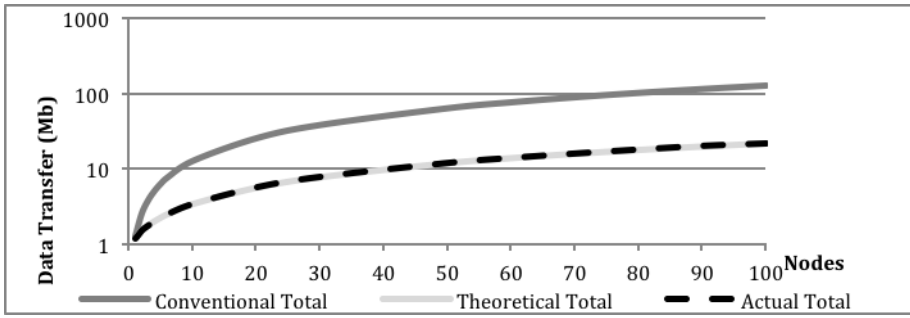

Figure 5: Data Transfer Required with a 100:1 File to Signature Ratio

There is still a considerable reduction in the quantity of data required for analysis in all cases in comparison with the conventional approach. Our experiments were carried out using data that resulted in zero second stage matches; there would be additional data transfer above what the model predicts when matches are present between the file set and signature library. However, the total amount of data transfer would still be lower than if the full MD5 hash values where distributed to each node.

5 Conclusions & Further Work

The data from our experiments indicates that our model can accurately quantify the total amount of data that will be transferred when distributed two stage signature detection is carried out using varying numbers of analysis nodes.

We overcome the limitations of our previous scheme (Hegarty et al., 2011), by removing the reliance on a single node to provide the second stage signatures. As this restricted the scalability of the approach, particularly when large numbers of stage two searches were required. The main contribution this paper makes is a technique that reduces and quantifies the amount of data required for distributed signature detection. This enables informed decisions to be made about the number of analysis nodes to employ in the signature detection process. As the time for data transfer is proportional to the amount of data transferred, our model can quantify the time required for transfer if the availability of bandwidth is known.

Further investigation is required to calculate the time complexity of the two stage search technique utilised in this paper. The focus on data quantity was deliberate as it is likely that the network component of the platform will be the bottleneck in the data intensive task of distributed signature detection.

6 References

Allen, W. H. (2005). Computer forensics. Security & Privacy Magazine, IEEE, 3(4), 59-62. doi:10.1108/09565690610677463

Anwar, F., & Anwar, Z. (2011). Digital Forensics for Eucalyptus. 2011 Frontiers of Information Technology (pp. 110-116). IEEE. doi:10.1109/FIT.2011.28

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., et al. (2009). Above the clouds: A berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28. Citeseer.

Burd, S. D., Jones, D. E., & Seazzu, A. F. (2011). Bridging Differences in Digital Forensics for Law Enforcement and National Security. 2011 44th Hawaii International Conference on System Sciences (pp. 1-6). IEEE. doi:10.1109/HICSS.2011.87

Delic, K. a., & Walker, M. A. (2008). Emergence of the Academic Computing Clouds. Ubiquity, 2008(August), 1-1. doi:10.1145/1459229.1414664

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud Computing and Grid Computing 360-Degree Compared. 2008 Grid Computing Environments Workshop (pp. 1-10). IEEE. doi:10.1109/GCE.2008.4738445

Garfinkel, S.L. (2007). Commodity grid computing with amazon s3 and ec2. Usenix, 7-13.

Grobler, C. P., Louwrens, C. P., & von Solms, S. H. (2010). A Multi-component View of Digital Forensics. 2010 International Conference on Availability, Reliability and Security (pp. 647-652). IEEE. doi:10.1109/ARES.2010.61

Haggerty, J., & Taylor, M. (2006). "FORSIGS: Forensic Signature Analysis of the Hard Drive for Multimedia File Fingerprints." in IFIP International Federation for Information Processing, 232(New Approaches for Security, Privacy and Trust in Complex Environments). Sandton, South Africa. doi:10.1007/978-0-387-72367-9_1

Haggerty, J., Llewellyn-Jones, D., & Taylor, M. (2008). FORWEB: File Fingerprinting for Automated Network Forensics Investigations. Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop (e-Forensics '08) (p. 29). Adelaide, Australia: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Hegarty, R., Merabti, M., Shi, Q., & Askwith, R. J. (2011). A Signature Detection Scheme for Distributed Storage. 6th International Annual Workshop on Digital Forensics & Incident Analysis (WDFIA 2011). London.

Hui, W., Yin, H., & Lin, C. (2009). Design and deployment of a digital forensics service platform for online videos. Proceedings of the First ACM workshop on Multimedia in forensics - MiFor '09 (p. 31). New York, New York, USA: ACM Press. doi:10.1145/1631081.1631089

Naqvi, S., Dallons, G., & Ponsard, C. (2010). Applying Digital Forensics in the Future Internet Enterprise Systems - European SME's Perspective. 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering (pp. 89-93). IEEE. doi:10.1109/SADFE.2010.28

Osborne, G., & Slay, J. (2011). Digital Forensics Infovis: An Implementation of a Process for Visualisation of Digital Evidence. 2011 Sixth International Conference on Availability, Reliability and Security (pp. 196-201). IEEE. doi:10.1109/ARES.2011.36

Palankar, M. R., Iamnitchi, A., Ripeanu, M., & Garfinkel, S. (2008). Amazon S3 for science grids. Proceedings of the 2008 international workshop on Data-aware distributed computing - DADC '08 (pp. 55-64). New York, New York, USA: ACM Press. doi:10.1145/1383519.1383526

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

Richard, G. G., & Roussev, V. (2006). Next-generation digital forensics. Communications of the ACM, 49(2), 76. doi:10.1145/1113034.1113074

Rimal, B. P., Choi, E., & Lumb, I. (2009). A Taxonomy and Survey of Cloud Computing Systems. 2009 Fifth International Joint Conference on INC, IMS and IDC (pp. 44-51). Washington, DC, USA: IEEE. doi:10.1109/NCM.2009.218

Rivest, R., L. (1992). RFC 1321 - The MD5 Message Digest Algorithm.

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

Yin, H., Hui, W., Li, H., Lin, C., & Zhu, W. (2012). A Novel Large-Scale Digital Forensics Service Platform for Internet Videos. IEEE Transactions on Multimedia, 14(1), 178-186. IEEE. doi:10.1109/TMM.2011.2170556