Intruder Detection through Pattern Matching and Provenance Driven Data Recovery

Anthony Chapman
Computing Science
University of Aberdeen
Aberdeen, UK
Email: r01ac14@abdn.ac.uk

***Abstract*—Intruder detection and recovering tampered data is challenging enough without the added complexity of the cloud or the forthcoming EU General Data Protection Regulation (GDPR), which will put greater pressure on companies to strengthen their cyber security or potentially face large fines. Intruder breach reporting and forensic analysis needs to drastically improve in order to avoid these potentially catastrophic fines. We conducted a conceptual exploration of intruder detection and data recovery methods. This paper aims to encourage further research for effective cloud security assurance with a focus on increased protection from tough legislation, such as complying with the forthcoming GDPR. We propose a framework which uses pattern matching to identify tampered data, provenance models for data assurance and audit trails to recover original data.**

***Keywords*—Cloud Security; Audit; Provenance; Tamper Detection; Data Recovery; GDPR**

I. Introduction

CLOUD computing research has experienced a surge of interest in recent years. Unfortunately, cloud security has advanced at a slower pace than other aspects of cloud systems and given the changes soon to affect the cloud security community, namely the EU GDPR [1], more of our attention must be directed to improve cloud security.

Recent reports have noticed a significant reduction in the global average time between a security breach and reporting a detection to just under 3 weeks in 2016 [2] compared to 1 month in 2015 [3] or 6 months in 2012 [4]. Although this is a significant improvement, in order to comply with the forthcoming GDPR, reporting of a breach must take place within 72 hours of discovery [1]. From this, it is clear that many enterprises will be unable to comply with the requirement to report any and all breaches within 72 hours of detection. It also suggests that system monitoring is not being done properly [5].

The new GDPR, applying in the UK from 25 May 2018 [1], aims to encourage firms to protect their client's personal data by penalising those with inefficient security measures. One of the stronger encouragement measures is fining companies that do not report breaches within 72 hours of discovery. The report should include what the intruder was looking at, what was tampered with, what was deleted and what was stolen. Many companies will struggle to meet this requirement because of the loss of vital forensic records. The longer an intruder remains inside a system undetected, the more damage they can do. The short reporting time (compared to the current average of 3 weeks) could hopefully reduce the amount of damage an intruder is able to do; the sooner a breach is detected and reported.

Detecting and reporting breaches as soon as they happen needs to be at the forefront of security for compliance with the new regulations. There is also a clear, outstanding need for a specified policy to control and track data as it flows throughout cloud infrastructure. This is to ensure that data custodians are meeting their obligations [6].

Corporate governance rules are also constantly changing. The emphasis of these changes are to place more on responsibility and accountability [7], social conscience [8], sustainability [9], [10], resilience [11] and ethics [12] on companies and data custodians. These changes alongside new legislations will force traditional principles of corporate governance towards towards stricter and more robust cyber security measures. Ever evolving technologies (with increasing complexity) heighten exposure to risk, particularly if the technologies (and their potential problems) are not fully understood [13]. Thus, there is a need for a more effective approach to address these security issues.

This paper focuses on tamper detection and data recovery and is structured as follows: Section II describes the motivations, implications and background related to this research. Section III describes the proposed framework and Section IV gives a breakdown of the benefits the framework could have on a system. The remaining sections consist of a discussion, which includes limitations, in Section V, and finally, a conclusion and future work in Section VI.

II. Motivation & Background

The new GDPR will displace some pressure from the customers to the data custodians. Although this is positive for customers, the companies who own the data need to radically improve breach detection and reporting as they will be held accountable for any and all unreported security breaches.

A. Implications

Intruders within a system have the potential to illegally access, modify, delete and/or extract data. Any of which could financially harm a company as well as damage their reputation [14]. Regulations, such as GDPR, encourage firms to protect

 ISBN: 978-1-61208-607-1

personal data by penalising those not seen to be doing so properly.

Post-attack business continuity measures are paramount in minimizing intruder damage [15]. Unfortunately, pre-attack measures (intruder deterrents) have received a lot more academic attention than post-attack measures (data recovery, forensic trails, etc.). Simple post-attack methods such as data recovery through regular back-ups could cause more damage if not done with care. For example, if a safe back-up is updated with unsafe (tampered) data. This can occur for a number of reasons, for instance if tampering is not detected prior to a back-up being taken, it may then be impossible to recover the original data.

The vast majority of financial institutions in the UK are woefully under-prepared to comply with the forthcoming GDPR. Current estimates suggest that UK banks could potentially suffer fines in the first year alone of over €5 Billion [16]. Campbell et al. [17] and Farrow et al. [18] have investigated the impact cyber breaches will have on the stock market value of firms with varying results. They found that the significance and the effect breaches will have is likely to be evolving over time.

Chow et al [19] also consider some implications and discuss difficulties with cloud auditing. They found that cloud doubts largely stem from the perceived loss of control of sensitive data and that current control measures do not address cloud's third party data storage and processing requirements adequately. They also express the likelihood of problems arising from over relying on cloud computing.

Having back-up data could help recover tampered data, unfortunately, it might not be efficient for companies with large amounts of data to store them. Even in cases where back-up data is kept, the intruder may still be able to access it also. Assuming the intruder tampers with a small part of the data, how can we: 1. locate the tampered data and 2. recover the original data?

Duncan and Whittington [20] explore checklists within various fields (medicine and accounting) and examine problems that are inherent with checklists in order to identify strategies that might be adopted by cloud computing to improve efficiency. One benefit found is that checklists enable systems to conform with standards, but note that this does not guarantee improved security. One drawback from checklists is that it may "deny an experienced practitioner the opportunity to develop a rounded understanding of the situation by being forced to focus on the individual trees rather than the wood as a whole".

B. Audit Trail

Audit trails are a fundamental part of accounting and finance, they provide assurance that company managers have presented a "true and fair" view of a companys financial performance and position, underpinning the trust and obligation of stewardship between company management and the owners of the company [21]. Accounting audit can be extended to IT audit, and further to cloud audit, where rather than treat the IT systems as black box components of the company systems, the IT systems themselves are audited to provide assurance that they are capable of delivering what is needed by the company [22].

An area of weakness arises when taking audit professionals from the accounting world out of their comfort zone, and placing them in a more technical field. Whilst the use of people with a computing background can overcome some of these issues, their lack of audit background presents another weakness [23]. Clearly further research is needed in this area [24].

Cloud adoption has not been straight forward either. This may be due to difficulties within cloud audit [25] as well as the possible belief that trust and privacy issues [26]–[29] also need further work before cloud auditing is achieved. A common theme is the recognition that cloud audit is far harder to perform than audit of non-cloud systems.

Forensic audit is used when fraud is discovered, to find and collect suitable evidence for presentation in a court case, whether criminal or civil. This can be extended to IT audit forensic trails which could be used to trace the acts of an intruder or backtrack the steps a system has taken when an error has occurred. This way, we may be able to identify errors and/or see what an intruder may have been interested in [30].

Greater accountability, and particularly a broadening of the scope of Service Level Agreements (SLAs) have been considered as a way to enhance cloud security and privacy. Achieving cloud accountability is a complex challenge; as we now have to consider large-scale virtual and physical distributed server environments to achieve (1) real-time tracing of source and duplicate file locations, (2) logging of a files life cycle, and (3) logging of content modification and access history [31].

C. Data Provenance

Data provenance (also referred as data lineage or pedigree) was introduced to better understand the origins of data within databases [32], [33]. Whole system provenance goes further, it gives the complete picture of a system, from initialisation to shutdown, by tracking metadata and transient system objects [34]. Provenance alone is not enough to detect an intruder in a system, so further components will need to be in play in order to identify a breach [35].

Like auditing, provenance is not well researched within the computing community. One of the reasons why it may not receive so much attention could be that provenance information cannot be trusted unless its integrity is assured. Moreover, provenance must be protected differently than regular data [36]. The fact that provenance models are fairly novel added to the large initial effort required to implement such a system to work within a current firm may be deterring companies from using such models.

Another issue which plagues both auditing and provenance is that the benefits do not become apparent during profitable and calm periods of business. They only appear when something (a security breach, an accounting error, etc.) disrupted normal working procedures or an unwanted result has been reported. Of course, by the time a company realises it might need provenance or auditing it will be too late.

 ISBN: 978-1-61208-607-1

Provenance is currently being used to create models which may be able to distinguish between legitimate and illegitimate behaviour in applications on a large cluster of machines [37]. The model could be extended for data recovery by restoring the data to a pre-breach safe state, the system could then carry out any operations it carried out during the breach to update the system to the desired state. One drawback from this method could be that the data recovery might not justify the potentially large computational expense; if only a small amount of data is tampered with, the whole system will have to be restored, not just the tampered data.

D. Tampered Data Detection

Current tampered data detection focuses on either time stamping [38] or system calls and activity logging [39]. Although both could be used to detect breaches, hackers may be able to access the time stamp files or activity files and remove or alter the activities during a breach. Through forensic analysis, it may be possible to determine when the tampering occurred, what data was tampered with, and perhaps who did the tampering [40].

A complementary soft security solution relying on detecting behavioural anomalies by evidence theory is proposed [41] and although this approach could identify anomalous behaviour as it is happening, it may not identify tampered data. Another possible limitation in such a system could be a lack of backtracking. If a behavioural anomaly is not detected, the damage caused may go unnoticed and may cause future problems without the ability to revert them.

By maintaining an audit log in the background of a system and using cryptographic techniques to ensure that any alterations to entries in the log are stored, we may be able detect unwanted tampering [42]. Unfortunately, such a system could easily become computationally expensive and may not be necessarily viable for large systems, especially if the larger system requires quick processing as this might be affected by the constant monitoring of the audit logs.

A method was proposed for secure logging which relied on secure keys between the logging machines [43]. Two major flaws were detected: (1) truncation attack (a special kind of deletion attack whereby the attacker deletes a contiguous subset of tail-end log entries) and (2) delayed detection attack (Where the system uses an old log file to verify current actions, an intruder could delete a file the system doesn't know exists yet) [44]. Both of which could seriously damage a system unless other mechanisms are in place to reinforce the system's weaknesses.

E. Data Recovery

Forensic trails are at the forefront of data recovery [45], from an efficient auditing system we may be able to find what was looked at, what was modified, what was deleted and possibly how it all happened. With auditing, it is possible to demonstrate compliance with data management policy and/or provide forensic data to determine the cause of any unintended data disclosure [6]. An intruder's objective, once they are embedded within a cloud system, will be to edit or delete forensic data in order to conceal their behaviour and avoid being detected.

Immutable data logging [5] is one part of an audit system which could greatly benefit a system. The advantage of being able to store every movement within a system has to be balanced against the large amount of memory required to store such movements. Another issue arises if an intruder finds a way to tamper with the logging files. If the system relies on the logging files and they are tampered with, then an intruder might be able to conceal their actions long enough to carry out whatever actions they want on the system without being detected.

System calls, storing keystroke and deletion requests are other aspects of an audit system that could be used to recover tampered data. Again, similar to immutable data logs, they can be computationally expensive and may cause greater harm if not used with care. These methods should be used within auditing systems but should be used with other methods in order to provide a complete system. Over relying on parts of auditing could cause more harm than good as aforementioned.

III. Proposed Framework

The proposed framework can be split into two stages. The first stage is tamper detection, this stage will determine where an intruder has breached the system and what data has been tampered with. The first stage will help firms comply with the GDPR breach reporting policy and help avoid potentially large fines which could cripple a company.

The second stage may undo the intruder's damage by recovering the original data through provenance and audit trails. This stage works alongside the first stage by identifying which data has been tampered with and focusing on that data, thus not having to waste computing power recovering non-tampered data.

```
ranges ← initialise pattern model ;
for every day do
    compare model range with data;
    if data within range then
        no breach ;
        update pattern model ;
    else if data outside range then
        investigate ;
        if tamper detected then
            report breach ;
            recover tampered data ;
        else if data untampered then
            update pattern model ;
end
```

Algorithm 1: Pseudo code for tamper detection. This pseudo code is designed to run on a single piece of data, for multiple datasets we can run the system on them individually creating custom ranges. The time between runs can vary according to different needs, we have used "every day" as an example.

 ISBN: 978-1-61208-607-1

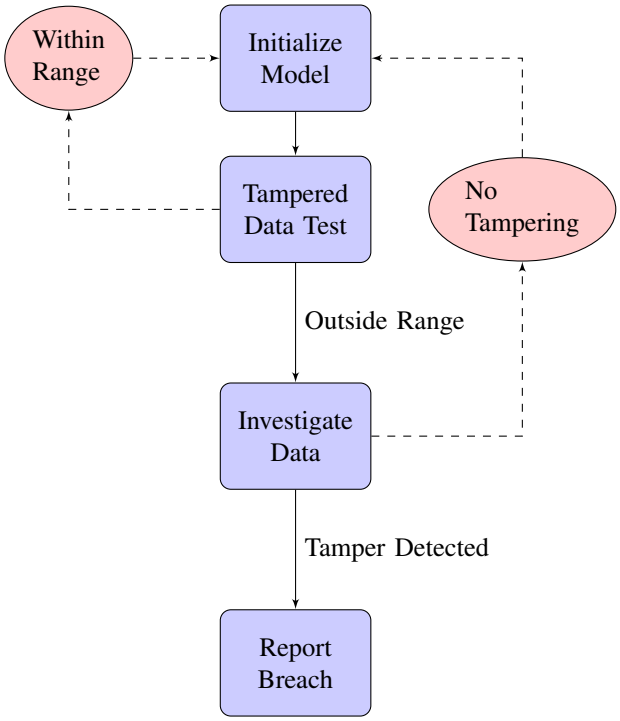

Fig. 1. Tamper detection flowchart.

A. Tamper Detection

Missing data prediction (imputation) uses statistical models to predict missing data based on the observed data [46]. These models create a range of possible values for every missing entry. Using this range they impute the missing values and account for uncertainty by running multiple times with a specified probability distribution [47].

We can adapt these methods to create the relationship models and ranges as it would normally do and then use these ranges to compare against the data. We could then identify whether any data falls outside their respective range and investigate whether it has been tampered with or not.

Firstly, we will create an initial model and ranges from the data by creating regression models for every data type, this will create the initial ranges and they will change as the system progresses over time. The initial ranges will be created using data already in the system. Using the initial model and ranges we can then compare, after a chosen time period (i.e. every 12 hours), the ranges from the initial data to the data currently in the system, as shown in Figure 1.

Notice that within the chosen time period, the data will have changed, if the data is within the range, then no tampering has been detected and we can recalculate the model (to include the the new data) and create new ranges. If any data falls outside the range, we should investigate the data and decide whether it has been tampered with or not. If it has, go to stage two (the data recovery stage) and then recalculate a new model and ranges after any tampered data has been recovered.

B. Data Recovery

Once tampered data has been detected we must recover the original data. Data provenance models and audit trails could work together to recover tampered data. The provenance models could provide assured data by stating a safe stage that can be trusted by the system. Once the assured data is ready, we can use auditing trails to recreate the original data by applying procedures that occurred between the assured data stage and the tampered data detection.

There are different ways to tackle this problem; one way would use the tamper detection model (the first stage of the framework) to identify which data has been tampered. Doing so will mean we can recover only the tampered data and not have to use unnecessary computing power on the rest of the dataset. Alternatively, we could use all of the assured data and all of the audit trails to recreate the complete data from the time of assured data to present, tamper detection, time. This method would make sure all tampered data during that period is recovered and it is less likely that any tampered data will be missed.

Pasquier et al. [48] proposed an approach based on Information Flow Control (IFC) that allows: (1) the continuous, end-to-end enforcement of data flow policy, and (2) the generation of provenance-like audit logs to demonstrate policy adherence and contractual/regulatory compliance. We can also extend this work to provide data-centric audit logs akin to provenance metadata in a format in which analyses can easily be automated.

IV. Framework Benefits

The proposed framework may be able to not only detect whether data has been tampered with but also locate the tampered data. By locating the tampered data, the system will be able to minimize computing powered required to recover tampered data. This may enable us to only work on the tampered data and not have to use more computing power than necessary dealing with the whole dataset.

Reporting time may be greatly reduced by discovering that your system has been breached. The time between breach and discovery may be reduced by running the tamper checking software on a regular basis. This may enable companies to comply with GDPR's 72 hour breach discovery reporting. Reducing the time from breach to discovery may also reduce the risk of companies having to pay large fines for non-compliance with increasingly strict regulations and also better protect personal and confidential data.

Once tampered data has been detected, the original data can be recreated through provenance and auditing. Being able to re-create data may remove or minimize the need for large data backups, thus reducing memory and hardware required.

Potentially, depending on the size of the data and computing power available, this system could be run daily or even a few times a day. Doing so will enable intruder detection at a daily or even hourly rate, hugely reducing the time it takes to detect an intruder after the system has been breached.

The audit models within the system maybe provide essential forensic data which may improve a company's security and potentially our understanding of the intruder's intentions. Through audit trails, we may be able to see how the intruder infiltrated the system and whether any data was stolen. This, of course, will help firms comply with the new GDPR.

 ISBN: 978-1-61208-607-1

V. Discussion & Limitations

Our proposed framework focuses on detecting (as opposed to preventing) breaches and recovering tampered data. Research regarding breach prevention is crucial to cloud security and having methods to cope with breaches, when preventions mechanisms fail, will only strengthen systems. Tamper prevention mechanisms include user monitoring and anomalous behavior detection [21]. System calls can also be used to ensure the credibility of logged data [49], they could also be used to detect intruders within a system as they may be analysed in order to identify the intruder's malicious intent.

Using pattern matching to detect tampered data may not detect intruders if they do not modify data. In such cases, system calls could be used to capture the intruder's path. Intruders will want to modify these calls, again this may be recognized by the models and alert the company of a security breach. By working together, pattern matching models and system calls may strengthen a system's security.

The new GDPR poses a moral dilemma for firms. The regulations state that a firm has to report a breach within 72 hours of discovery. Notice, it states within 72 hours of discovery, not 72 hours after the system is breached. It could be possible for firms to not apply intruder detection software until it affects the running of the company, thus not having to use time and money complying with GDPR unless the breach affects them. This could potentially expose or compromise personal and confidential data.

When considering the proposed framework a number of limitations were identified, the first one is the computational expense vs benefits from the framework. Businesses will have to consider whether implementing such a system will benefit them enough to justify running it. This will be especially challenging for smaller firms, which are usually (although wrongly so) less likely to be concerned about cyber attacks than larger firms. Additionally, implementing such systems may have bigger initial financial impact on smaller businesses, implying they will be less likely to want to use such systems.

A more technical limitation lies at the heart of modeling theory. Although regressions have great modeling power they also come with a pinch of uncertainty. If not done with care, the data "ranges" proposed in this paper might either overestimate or underestimate data tampering. Overestimating may produce threat warning for data when no breach has occurred, this may waste computational power, labour and ultimately, money. Underestimating may cause the converse problem and may not detect tampered data, this may lead to breaches going unnoticed and possible large fines from governing bodies such as GDPR, not to mention damage to customers whose data has been stolen.

VI. Conclusion & Future Work

This paper has identified some potential weakness which if not corrected before the new GDPR is enforced (25th May 2018 for the UK) could lead to firms being financially penalised. One problem identified is the current average time for breach detection is circa 3 weeks. This time needs to be reduced to minimize the amount of damage caused by breaches. Another problem identified is the lack of post attack coping mechanisms, specifically for rectifying tampered data. Finally we noticed a gap in research for locating tampered data and separating it from the rest of the dataset.

We proposed a framework which may be used to identify tampered data at intervals during the day to minimise the time an intruder spends within a system after they have breached it. The framework proposed includes a method for recovering tampered data in an efficient way by working only on the tampered data by minimising the computing power required to recover data. The method could also solve issues with back-up data recovery by not relying on large digital storage or potentially compromised back-ups.

The framework can be applied at different intervals according to company needs. The companies will have to decide the optimal interval for the framework which both minimizes the time between breach and discovery as well as optimising the computational power allocated for the system. Checking for tampered data too often may take up too much computing power but a large interval may delay detection. The proposed framework will enable firms to not only comply with the new GDPR but also further protect personal and/or confidential data. This is especially important since we live in a world where regulations and corporate governance rules are constantly evolving.

Finally, future investigations could be carried out to address the already discussed limitations. To minimise over and underestimation, testing scenarios should be created which will create empirical data that could be used to better understand the effects of over and underestimation. By better understanding the effects of over and under estimation, we may be able to optimise the system to efficiently detect breaches. When it comes to justifying the software to businesses of all sizes, it might be beneficial to simulate cyber attack behaviour and have user studies demonstrating how the software might work. Only by educating the users (not just financial institutions are at risk) of cyber security risks and potential financial damage posed by ever changing regulations, will they be able to make a fully informed decision on whether this type of framework is suitable for them

References

[1] ICO. (2017) Overview of the general data protection regulation (gdpr). [Online]. Available: https://ico.org.uk/for-organisations/data-protection-reform/overview-of-the-gdpr/ [Last Accessed: 22 Dec 2017]

[2] Verizon, "2016 data breach investigations report," 2016.

[3] Verizon, "2015 data breach investigations report," 2015.

[4] Verizon, "2012 data breach investigations report," 2012.

[5] R. A. K. Duncan and M. Whittington, "Creating an immutable database for secure cloud audit trail and system logging," in *Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, 19 February 2017-23 February 2017, Athens, Greece*, 2017, pp. 54–59.

[6] T. F.-M. Pasquier, J. Singh, J. Bacon, and D. Eyers, "Information flow audit for paas clouds," in *Cloud Engineering (IC2E), 2016 IEEE International Conference on*. IEEE, 2016, pp. 42–51.

[7] M. Huse, "Accountability and creating accountability: a framework for exploring behavioural perspectives of corporate governance," *Brit J. Mgt*, 2005, pp. 65–79.

[8] A. Gil, "Corporate governance as social responsibility : A research agenda," *Berkeley J. Intl L.*, 2008, pp. 452–478.

[9] C. Ioannidis, "Sustainability in information stewardship: Time preferences, externalities and social co-ordination," *WEIS 2013*, 2013.

 ISBN: 978-1-61208-607-1

[10] A. Kolk, "Sustainability, accountability and corporate governance: Exploring multinationals reporting practices," *Business Strategy and the Environment*, 2008, pp. 1–15.

[11] F. S. Chapin, "Principles of ecosystem stewardship: Resilience-based natural resource management in a changing world," *Springer*, 2009.

[12] S. Arjoon, "Corporate governance: An ethical perspective," *J. Bus Ethics*, 2005, pp. 343–352.

[13] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering & System Safety*, 2009, pp. 125–141.

[14] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and L. Zhou, "Increasing cybersecurity investments in private sector firms," *Journal of Cybersecurity*, vol. 1, no. 1, pp. 3–17, 2015.

[15] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[16] P. Tobin, M. McKeever, J. Blackledge, M. Whittington, and B. Duncan, "UK Financial Institutions Stand to Lose Billions in GDPR Fines: How can They Mitigate This?" in *Br. Account. Financ. Assoc. Scottish Area Gr. Annu. Conf.*, BAFA, Ed., 2017, p. 6.

[17] K. Campbell, L. A. Gordon, M. P. Loeb, and L. Zhou, "The economic cost of publicly announced information security breaches: empirical evidence from the stock market," *Journal of Computer Security*, vol. 11, no. 3, pp. 431–448, 2003.

[18] S. Farrow and J. Szanton, "Cybersecurity investment guidance: Extensions of the gordon and leob model."*Journal of Information Security*,vol. 7, no. 2, pp. 15, 2016.

[19] R. Chow et al., "Controlling data in the cloud: outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 85–90.

[20] B. Duncan and M. Whittington, "Reflecting on whether checklists can tick the box for cloud security," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*. IEEE, 2014, pp. 805–810.

[21] M. Neovius and B. Duncan, "Anomaly Detection for Soft Security in Cloud based Auditing of Accounting Systems," in *Closer 2017 - 7th Int. Conf. Cloud Comput. Serv. Sci.*, 2017, pp. 1–8.

[22] B. Duncan and M. Whittington, "Compliance with standards, assurance and audit: Does this equal security?" in *Proceedings of the 7th International Conference on Security of Information and Networks*. ACM, 2014, p. 77.

[23] B. Duncan and M. Whittington, "Enhancing cloud security and privacy: broadening the service level agreement," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, 2015, pp. 1088–1093.

[24] R. A. K. Duncan and M. Whittington, "Enhancing cloud security and privacy: the power and the weakness of the audit trail," *Cloud Computing 2016*, 2016.

[25] H. S. Herath and T. C. Herath, "It security auditing: A performance evaluation decision model," *Decision Support Systems*, vol. 57, pp. 54–63, 2014.

[26] R. K. Ko, P. Jagadpramana, and B. S. Lee, "Flogger: A file-centric logger for monitoring file access and transfers within cloud computing environments," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*. IEEE, 2011, pp. 765–771.

[27] R. K. Ko, B. S. Lee, and S. Pearson, "Towards achieving accountability, auditability and trust in cloud computing," *Advances in Computing and Communications*, pp. 432–444, 2011.

[28] S. Pearson, "Taking account of privacy when designing cloud computing services," in *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*. IEEE Computer Society, 2009, pp. 44–52.

[29] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 693–702.

[30] B. Duncan, "Enhancing cloud security and privacy: The cloud audit problem," *The Seventh International Conference on Cloud Computing, GRIDs, and Virtualization*, 2016.

[31] R. K. L. Ko, B. S. Lee, and S. Pearson, "Towards achieving accountability, auditability and trust in cloud computing," *Commun. Comput. Inf. Sci.*, vol. 193 CCIS, pp. 432–444, 2011.

[32] P. Buneman, S. Khanna, and W. C. Tan, "Why and where: A characterization of data provenance." Springer.

[33] A. Woodruff and M. Stonebraker, "Supporting fine-grained data lineage in a database visualization environment," in *Data Engineering, 1997. Proceedings. 13th International Conference on*. IEEE, 1997, pp. 91–102.

[34] T. Pasquier et al., "Practical whole-system provenance capture," in *Proceedings of the 2017 Symposium on Cloud Computing*. 2017, pp. 405–418.

[35] T. Pasquier et al., "Data provenance to audit compliance with privacy policy in the internet of things," *Personal and Ubiquitous Computing*, pp. 1–12, 2017.

[36] A. Bates, B. Mood, M. Valafar, and K. Butler, "Towards secure provenance-based access control in cloud environments," in *Proceedings of the third ACM conference on Data and application security and privacy*. ACM, 2013, pp. 277–284.

[37] X. Han et al., "Frappuccino: Fault-detection through runtime analysis of provenance,". 2017.

[38] A. Imran, N. Nahar, and K. Sakib, "Watchword-oriented and time-stamped algorithms for tamper-proof cloud provenance cognition," in *Informatics, Electronics & Vision (ICIEV), 2014 International Conference on*. IEEE, 2014, pp. 1–6.

[39] H. Nguyen et al., "Cloud-based secure logger for medical devices," in *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*. IEEE, 2016, pp. 89–94.

[40] K. E. Pavlou and R. T. Snodgrass, "Forensic Analysis of Database Tampering," *ACM Trans. Database Syst.*, vol. 33, no. 4, pp. 30:1–30:47, nov 2008.

[41] M. Neovius and B. Duncan, "Anomaly detection for soft security in cloud based auditing of accounting systems," in *Proceedings of the 7th International Conference on Cloud Computing and Services Science*. SciTePress, 2017.

[42] R. T. Snodgrass, S. S. Yao, and C. Collberg, "Tamper detection in audit logs," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 504–515.

[43] B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 2, pp. 159–176, 1999.

[44] D. Ma and G. Tsudik, "A new approach to secure logging," *ACM Transactions on Storage (TOS)*, vol. 5, no. 1, p. 2, 2009.

[45] S. Thorpe et al., "Towards a forensic-based service oriented architecture framework for auditing of cloud logs," in *Services (SERVICES), 203 IEEE Ninth World Congress on*. IEEE, 2013, pp. 75–83.

[46] D. McNeish, "Missing data methods for arbitrary missingness with small samples," *Journal of Applied Statistics*, vol. 44, no. 1, pp. 24–39, 2017.

[47] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–67, 2011

[48] T. F.-M. Pasquier, J. Singh, and J. Bacon, "Clouds of things need information flow control with hardware roots of trust," in *Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 467–470.

[49] G. R. Weir and A. Aßmuth, "Strategies for intrusion monitoring in cloud services," in *The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*, 2017, pp. 1–5.

 ISBN: 978-1-61208-607-1