Security and Data Accountability in Distributed Systems: A Provenance Survey

Yu Shyang Tan, Ryan K L Ko, Geoff Holmes
Cyber Security Lab, Dept of Computer Science
University of Waikato, Hamilton, New Zealand
{yst1, ryan, geoff}@waikato.ac.nz

***Abstract*—While provenance research is common in distributed systems, many proposed solutions do not address the security of systems and accountability of data stored in those systems. In this paper, we survey provenance solutions which were proposed to address the problems of *system security* and *data accountability* in distributed systems. From our survey, we derive a set of minimum requirements that are necessary for a provenance system to be effective in addressing the two problems. Finally, we identify several gaps in the surveyed solutions and present them as challenges that future provenance researchers should tackle. We argue that these gaps have to be addressed before a complete and fool-proof provenance solution can be arrived at in the future.**

Keywords - Provenance, Security, Data Accountability, Distributed Systems, Cloud Computing

I. Introduction

In 2011, statistics showed that the US suffered an average of 117 cybersecurity attacks a day that involve data loss or theft, computer intrusion and/or privacy breaches [1]. This fiscal report also showed that spending $13 billion dollars did not stop security breaches from happening. This fact showed clearly that traditional security measures such as setting up of perimeter defence [2], are both ineffective in preventing data loss and costly.

The openness of the Internet has led to calls for new ways to look at security and privacy of data [3, 4]. Yet the abstraction layer in distributed systems such as cloud systems [5], contrasts with the principles of accountability of data. Accountability of data requires transparency on how data is handled. However, the abstraction layer that is used to hide the complexity of virtualisation from end users, abstracts away how data is managed and accessed.

On the other hand, distributed platforms such as Hadoop [6] and Grids [7] focus on providing scalability and computing power to users. Such platforms overlook the issue of providing accountability for data. Hence, there is an urgent need for tools that enable accountability for distributed systems. This is even more so as distributed systems become increasingly popular as the choice for data sharing [8] and project collaborations [7].

We believe that protecting privacy rights of data can be achieved by enabling accountability in systems [9, 10]. Such accountable systems are able to report back to users on how their data is being managed, who has accessed their data, when and what modifications have been performed on their data. Only by knowing what is going on with their data, can users then be sure that their data is not being misused unknowingly by others [11]. Data provenance research addresses the issue of tracking such information for data.

Data provenance is defined differently based on the context where it is applied. In data-centric areas such as databases, data provenance is defined as the description of the origins of a piece of data and the process by which it arrives at the database [12]. In workflow-centric areas such as e-Science, data provenance is largely regarded as the (semi-) or automatically and systematically captured and recorded information that helps users or computing systems to determine the derivation history of a data product, starting from its original sources and ending at a given repository [13].

In the context of this paper, we view provenance of data as the information that depict the actions performed on data and the entities responsible for those actions, throughout its life cycle. A data's life cycle can consist of several stages. This includes from creation, using the data and till the data is destroyed. For more details, we refer readers to [14].

In this paper, we look at how provenance can be used to improve security and protect the privacy of data in distributed systems. We first survey past work on provenance in distributed systems in this paper's context in Section III. We derive from the survey, a set of minimum requirements necessary for provenance systems to drive accountability and security in distributed systems in Section IV. Finally, we identify and discuss research gaps in provenance for distributed systems in the context of security and accountability in Section V.

II. Related Works

The use of provenance techniques in computer science was first discussed by Becker et al. in the seminal paper in [15]. Since then, the usage of provenance in different fields of computer science has been widely appreciated.

One of the main driving factors for the use of provenance in distributed systems was e-Science. However, the intention was not for security or protecting the privacy of data in the systems that those proposed solutions were meant for. Rather, provenance was applied to tackle issues like reproducibility and re-usability of experimental results and workflows [16] and troubleshooting of scientific workflows [17]. This was illustrated through comprehensive survey studies, such as [13, 18, 19], conducted throughout the years.

Provenance is applied to databases for providing a platform for users to understand the data being stored. Questions that are addressed by provenance research in databases includes; how

the data is derived [20], where the source is and why the result of the data is what it is [12]. Studies from different perspectives were conducted to understand how provenance was applied in databases. For example, Glavic et al. [21] studied and categorized how different provenance models were used in databases, while Tan et al. gave an overview of proposed provenance solutions in [22] from a technical point of view and in [23], from a framework point of view. Freire et al. [24] gave a general overview of applications of provenance in distributed systems. The surveyed provenance solutions were categorized according to the type of systems which they were designed for; workflow, OS and process based systems. There were many other surveys of applications of provenance such as [25, 26] in other fields. Having said that, the focus of the surveys was on capturing provenance information for analytical and exploration purposes.

Zhang et al. provided a more relevant survey of provenance in [27]. They looked at mechanisms that support the collection of provenance in the cloud and discussed the challenges of provenance collection in the cloud. Their proposed DataPROVE approach provided the basis for provenance as an enabler for accountability of data in the cloud. In this survey paper, we look at not only using provenance for accountability of data, but also for securing systems. We also seek to extend our scope from cloud systems to general distributed systems.

III. Provenance in Distributed Systems

Distributed systems lack provenance solutions that addresses the security of systems and data accountability. System level solutions such as Chimera [28], myGrid [30], TAP [31], Quill++ [32] and Taverna [33] and from a model perspective [34, 35], are some of the proposed provenance solutions in distributed systems that focus on addressing other issues.

The main motivation of providing provenance capabilities for the solutions mentioned thus far, was to support application needs (E.g. e-Science applications). Provenance provided the means to verify datasets and for troubleshooting workflows.

In this section, we divert our focus to studying the usage of provenance in distributed systems with the goal of securing systems and for enabling accountability and privacy of data stored on those systems. The securing of distributed systems and ensuring the accountability and privacy of data in distributed systems may seem to be two disparate goals. Nevertheless, we show through the approaches we surveyed in this paper, that both goals can be achieved through provenance capabilities within the said system. A quick overview of the granularity of the approaches surveyed in this paper is illustrated in Figure 1. We do emphasise that the figure does not attempt to describe an order of layering, but instead is a list of layers on which provenance solutions have been proposed.

A. System Layer

Approaches for securing systems can be divided into two categories; *prevention* and *detection*. Researchers have found provenance better suited to support *detection-oriented* approaches (E.g. fault detection, forensic analysis). To achieve application independent provenance collection, the approaches discussed in this subsection looks at collecting provenance

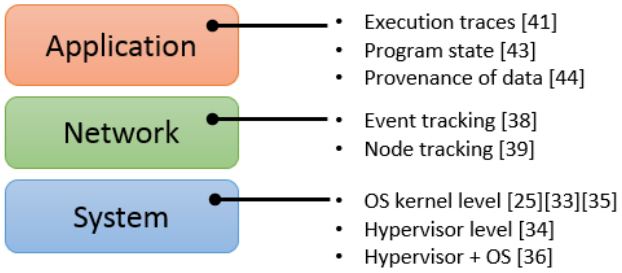

Fig. 1: List of granularity of surveyed approaches

information at the layer where file and system operations take place; the system layer.

Muniswamy-Reddy et al. proposed PASSv2 [36], an extension from their prior work PASS [29]. In PASS, provenance of objects was collected on a single host by intercepting system commands (E.g. moving and deletion of files) at the kernel level. Provenance collection was extended in PASSv2 to other layers within a system; system, application and middleware layers. They argued that through collecting the provenance of objects on multiple granularities, users would be able to detect intrusions or perform forensic analysis on distributed systems. Having said that, the PASS prototypes were kernel version dependent, as the proposed implementations required tight integration with the kernel of the underlying operating system. Provenance collection was also absent at the network layers in the PASS prototypes. PASSv2's cross host tracking relied on coordination of data transfer on a higher layer (E.g. copying of files from local to a network file server through a workflow engine). This resulted in the PASS prototypes not being able to track propagation of data from host to host that originated from the host level (E.g. spreading of viruses).

Recognizing PASS's dependency on the host kernel, Macko et al. [37] took a different approach; capturing of provenance from the hypervisor level. They modified the Xen hypervisor to collect provenance information from guest kernels (E.g. kernels of virtual machines running on the hypervisor). The collection of provenance was achieved by placing an interceptor on the DomU to intercept system calls from Xen's syscall_enter_mechanism. While the proposed solution was kernel independent, it was still not able to track provenance across physical machines in a cloud system due to the absence of network provenance. Zhang et al. [27] also pointed out the lack and need for security mechanisms to protect and maintain the integrity and confidentially of provenance records.

Differing from the previous two approaches, the S2Logger [38] and Flogger [39] took a data-centric approach to provenance tracking. Flogger was designed to track data in cloud architectures. File operations were logged both on the virtual and physical machine level. This allowed Flogger to map file operations logged on the virtual level to actual file operations captured on the physical host. Hence, manipulation of files from different virtual machines which sits on the same physical host can be correlated.

Coming from a data-centric approach, S2Logger [38] the first working cloud provenance solution to track data from end-to-end, was built on Flogger. Activities such as create, reads and writes of data on files were tracked at the file and block

level in the cloud. This was achieved by leveraging Flogger's ability to track file operations on both physical and virtual machines. S2Logger also extended provenance collection to include network send and receive events. By correlating the send and receive events with file reads and writes between different hosts, S2Logger was able to track data transfers across different hosts (both virtual and physical) in the cloud.

This was in contrast with the PASS system, where only file operations were tracked. From the PASS perspective, transferring data read from a file from one host to another would appear as two separate sets of read/write operations in the provenance records of two different hosts. There would be no assured way to indicate that the two sets of operations originated from a data transfer operation (This was not achievable if the data was transferred to a file with different name. (E.g. transferring from file A in host X to file B in host Y)). In contrast, because S2Logger was able to correlate network and system events, the two sets of operations would be linked clearly by send and receive events on both hosts (send on host X and receive on host Y).

To better aid analysis, S2Logger provided a visualisation interface to help users clearly establish such relationships between seemingly disparate events in the provenance records. Through such a visualisation interface, S2Logger was able to help identify the flow of data in the cloud. Suen et al. demonstrated how the visualisation interface, coupled with S2Logger's provenance capturing capabilities in the cloud, was able to detect malicious insiders and cloud data leakages in near real-time. Having said that, S2Logger suffered from the same inadequacy as the PASS system; being kernel version dependent due to tight integration with the kernel.

The approaches described thus far mainly concern provenance collection at the system layer for forensic analysis of system security. However, by taking a data-centric approach such as in [38], how data changes and who, where and how data is accessed in the cloud can be tracked. With such information, accountability of data can be established as highlighted by some of the use cases in [36] and [38].

B. Network Layer

In this subsection, we look at provenance tools that operate at the network level (E.g. tracking of messages sent and received). These tools usually track and capture the provenance of network events (E.g. monitoring network ports for send and receive messages) or the state of nodes within a network.

BackTracker [40], a tool that monitors for intrusion at the network protocol level, was extended in [41] to enable tracing the source of intrusion across multiple hosts within a network. The Bi-directional Distributed Backtracker (BDB), tracks intrusion from a detected host to the source by building causality graphs using events collected from each individual machine deployed with the BDB tool. Such a causality graph formed a provenance-like graph that gave the analyst a view of the sequence of events that depicts the path of the intrusion as it travels through the system.

To counteract the tendency of graph explosion, heuristics such as *highest process and most recent packet* were used to prioritise packets to be added to the graph. The downside was that BDB required each host within the network to be deployed with the BDB tool. Failure to do so potentially opens up an avenue for intrusion to go undetected as traffic coming from an unmonitored host cannot be tracked. Also the BDB tool did not take measures to secure the collection and management of provenance (events) records against tampering.

Recognizing the need for provenance recording in an adversarial setting, Zhou et al. proposed secure network provenance (SNP) [42]. They assumed a threat model of a Byzantine fault [43] and used authenticators to maintain tamper-evident logs. Through a set of acknowledge procedures that involved the use of the authenticators, send and receive events are guaranteed to be executed by the respective nodes. Consistency of logs (E.g. whether the logs have been tampered with) were determined by comparing the entries between the logs kept by different nodes. The SNP technique was tested on various applications through their prototype, SNooPy. Tests have shown that the prototype generated a substantial amount of overhead in terms of processing load and network traffic, due to the sending, receiving of authenticators and verifying and signing of signatures. Also, a modification of code is required on both SNooPy and the applications before interaction can happen, making the prototype impractical for deployment into real systems.

In the context of system security analysis, provenance collected by the BackTracker tool is more useful as compared to the SNooPy prototype. BackTracker allows analysts to accurately pin-point the source of the intrusions and how the intrusions propagate through the network. In contrast, SNooPy only allows the identification of faulty nodes within the network, without further details. However, the capability to provide tamper-evident provenance information in an adversarial setting is an important feature which is lacking in many provenance solutions.

C. Application Layer

In this subsection, we look at collecting provenance for assessing whether an application is working correctly and accountability of data. Provenance solutions discussed in this subsection operate on the application layer or middleware layer as opposed to the network or system layer.

Singh et al. [45] presents a platform where execution traces of applications were monitored, logged and analysed. The proposed platform was built on top of P2 [47], a system for distributed algorithm development. The state of the algorithms were monitored and logged using a *tracer*, through the buffer of the P2 system. The collected logs can be queried using a query language, OverLog. The platform which Singh et al. proposed allows programmer to detect and analyse for bugs, security compromises or identifying fault-tolerance problems within their algorithms.

Pip [46] provided developers with an annotation library which allows programs to generate events and resource measurements for monitoring purposes. Through a declarative language, developers describe their expectations of how their application is to work. With the expectations and the application's execution traces logged, the Pip middleware then checks and reports unexpected behaviours to the developers.

E-notebook [44] is designed for supporting trust and accountability in data sharing. The middleware interfaces directly

with instruments' on-board software and records the context in which the raw data is generated. It also records all transformations applied to the dataset. These provenance records are modelled as a directed acyclic graph (DAG) and digitally signed using the user's key, so as to certify the authorship of data. A role-based trust management language is then used to setup a social trust model where users can recommend levels of trust for other users, based on their own experience in using data from those users.

These proposed provenance solutions are useful for detecting unexpected behaviours and for tracking and recording transformations applied to data, on the application layer. However modifications or states which are produced external of the application will not be detected as the monitoring tools are not designed to capture those actions (E.g. rootkits, trojans). As such, there is a need to track provenance at multiple levels to allow analysis a more complete overview of what is happening.

IV. Requirements for a provenance framework

A. Requirements for Provenance Systems

To summarise our survey thus far, we compiled a list of minimum requirements necessary for provenance frameworks to be effective when used in the context of system security and data accountability in distributed systems. We describe the requirements in the list below:

- **Cross host tracking**: In a distributed system, it is essential that provenance tracking of an object is possible across different hosts within a network. This is especially so if the provenance information is to be used for security of system or data accountability purposes as interaction with the object can come from other hosts or spread to other hosts. (E.g. virus infecting other hosts through a network, accessing data and modifying it from another remote host).
- **Decoupling**: In this context, we interpret decoupling on two levels, application and platform.
  - **Application Decoupling** refers to enabling solutions to monitor generic applications running on the layers above without having the need to perform modification to those applications. Such a property will allow the provenance solutions to monitor new applications or programs as they are being run on the system. This is important as it enables detecting of malware or to support ad-hoc data monitoring.
  - **Platform Decoupling** looks at solutions which do not require tight integration with the underlying system components such as an operating system's kernel module. This will allow solutions to be impervious to sudden changes made to the system.
- **Multi granularity**: As illustrated in [36] and [38], provenance from multiple layers of a system helps the analyst to construct a complete picture of what is going on in the system. This is especially useful when attempting to analyse how an intrusion began or what or who is modifying the data.
- **Security mechanisms**: In the context of system security and accountability of data, it is critical that provenance records and their integrity remain secured. This requirement is further broken down into sub requirements and discussed in-depth in Section IV-B.
- **Interface for analysis**: Equally important to capturing and managing provenance records, is an interface for users to retrieve and analyse the recorded provenance information. Without such an interface, even the most detailed provenance will be just a white elephant. Intuitive interfaces such as interactive visual interfaces are necessary in helping analysts identify problems, detect trends and understand what is going on, at a glance.

We compare and show the surveyed solutions thus far with the list of requirements in Table I. We observe that most of the surveyed solutions do not address provenance collection on multiple levels of a system, possibly due to the complexity involved in coordinating analysis and collection of provenance information. However, new generation malwares and exploitation techniques such as rootkits [48] and alternate data streaming [49], often inflicts damage to a system at a different granularity as shown in [49]. We also observe few solutions that address the requirement of being platform decoupled. We attribute the lack of solutions addressing this requirement to the technical difficult and complexity involved in constructing such a solution. Having said that, we recognize that the table is **not representative** of the provenance research landscape in distributed systems.

B. Securing Provenance

Securing of provenance information is equally important as the tracking of it, especially in the context of system security and data accountability. Having reliable provenance is crucial as accurate and truthful analysis of the state of distributed

TABLE I: Minimum requirements for provenance frameworks for security and accountability

Provenance approaches	Cross host tracking	Decoupling		Multi granularity	Security mechanisms	Interface for analysis
		Platform Decoupled	Application Decoupled			
PASS[29]	✗	✗	✓	✗	✓	✗
PASSv2[36]	✓	✗	✓	✓	✓	✗
PASS with Xen[37]	✗	✓	✓	✗	✓	✗
Flogger[39]	✗	✗	✓	✗	✓	✓
S2Logger[38]	✓	✗	✓	✓	✓	✓
BDB[41]	✓	✓	✓	✗	✗	✓
SNooPy[42]	✓	✗	✗	✗	✓	✗
E-notebook[44]	✗	-	✓	✗	✓	✓
P2 extended[45]	✓	-	✓	✗	✗	✗
Pip[46]	✓	-	✗	✗	✗	✓

TABLE II: Comparing provenance approaches to list of security properties

Solutions focusing on provenance security	Confidentiality	Tamper-evident	Authenticity	Reliable collection
Rosenthal et al.[50]	✓	✗	✗	✗
SecProv[51]	✓	✗	✗	✗
Lu et al.[52]	✓	✓	✓	✗
Kairos[53]	✗	✓	✓	✗
Bonsai [54]	✗	✗	✓	✗
Provenance frameworks that satisfies the security requirement in Table I				
SNooPy[42]	✗	✓	✓	✗
E-notebook[44]	✗	✗	✓	✗
Flogger[39]	✗	✗	✗	✓
S2Logger[38]	✗	✗	✗	✓
PASS[29]	✗	✗	✗	✓
PASSv2[36]	✗	✗	✗	✓
PASS with Xen[37]	✗	✗	✗	✓

system's security and whether a data is trustworthy depends on the underlying provenance information. However, reliable provenance information involves several properties which we list below:

- **Confidentiality**: Confidentially of provenance records is important due to the sensitivity of the information within the records. Unauthorised users should not have access to sensitive information about the object which the provenance is describing. Good confidentiality properties also means unauthorised users should not be able to infer sensitive information from their view of the provenance information.
- **Tamper-evident/Integrity**: Tamper-evident measures are measures that maintain the integrity of the content of provenance records. With such measures, if any entity modifies the provenance records, be it intentional or unintentional, it will be obvious to the checker when the provenance records are validated.
- **Authenticity**: Authenticity of provenance records refers to allowing analysts to identify accurately who generated this provenance record. Authenticity also covers the identity and ownership of the data.
- **Reliable collection**: Reliable collection of provenance records looks at having trustworthy and accurate provenance collection mechanisms. This will help establish that the provenance records are accurate, right from the beginning.

Next, we survey some existing works that address the security requirements mentioned in the list above.

Securing of provenance records need to start from the very moment it is being collected. Without reliable collection mechanisms or a trustworthy environment for provenance to be collected, there is no way to determine whether provenance information collected can be trusted.

One possible solution to this issue is to collect provenance at the lowest level; the system kernel level. Since normal users usually do not have access to the system level, one can weakly infer that provenance collected at the system level is reliable. This solution is adopted in many system level provenance collection frameworks such as [29, 38, 39]. However, this still does not give the guarantee that provenance collected is accurate.

Lyle et al. suggested the use of trusted computing techniques to tackle the issue of reliability [55]. Through the use of a Trusted Platform Module (TPM), the hardware is made tamper-resistant. Hence a trusted and reliable environment where provenance collection can take place is created. However, the major disadvantage with using TPM-based techniques is the slow processing speed. This is mainly due to the amount of hashing and encryption operations involved and the low-speed processors that are being used by TPM hardware.

Rosenthal et al. [50] looked at scalable methods to manage access policies on components of a provenance graph. They adopted an attribute-based access control methodology, where factors were considered as attributes (E.g.project assignments, threat severity). Confidentiality was achieved by omitting provenance information from the resulting provenance graph when unauthorised users queried the database. On the other hand, Chebotko et al. proposed SecProv [51], a visual interface that used role-based access control to control a user's view of provenance information. Access control was implemented at the *task*, *port* and *data channel* level of the provenance graph. Similar to Rosenthal et al.'s approach, provenance information were omitted from unauthorised users.

Lu et al. addressed the issue of making provenance records tamper-evident, another important property for securing provenance in [52]. They used the bilinear paring technique [56] to uphold the properties of tamper-evident and confidentiality in provenance records. In this manner, they argue that provenance information can then be trusted.

Kairos [53] looked at the authenticity of provenance records in a grid computing environment. The authorship of the provenance record was first marked through the use of a user digital signature. A time-stamp authority was then used to generate a tamper-evident proof of authorship for the provenance record. Bonsai [54] attempts to establish the authenticity of provenance records by appending digital signatures of users or operators to the provenance records. To make the approach scalable, verification in Bonsai is done on-demand (E.g. verification is executed only when requested by users).

There are other works such as [57–61] that looked at securing provenance, however, we do not discuss those works here as they are done out of the context of distributed systems.

To summarise, we compare the surveyed solutions in this subsection, along with the solutions which are marked as satisfying the security requirement in Table I, with the list of security requirements. We present the comparison in Table II. We observe that confidentiality of provenance information

is not addressed by many of the surveyed solutions. This might be due to the contrast in hiding provenance information. We discuss this in more detail in Section V. Another observation we made was the lack of solutions for making provenance tamper-evident, especially the frameworks that deal with provenance collection. Ensuring the integrity of provenance records is crucial, especially in the context of security and data accountability.

Having said that, we recognize that the tables presented are not representative of the provenance research landscape in distributed systems. However, we do seek to provide insights to readers on the important factors to consider when designing solutions for the security of systems and data accountability using provenance.

V. Challenges Ahead

The provenance solutions surveyed in this paper addressed analysing security state of distributed systems and data accountability in different aspects. This includes provenance collection on different granularities and the securing of provenance information. However, there are still gaps that need to be addressed before a complete and fool-proof provenance solution can be arrived at.

- **Tracking of provenance outside of the system**: One of the weaknesses of solutions proposed thus far; tracking of provenance of data or state of the system can only be achieved within the boundary of the proposed solution. For example, BDB requires the tool to be installed on all hosts within the network in-order for it to be able to track intrusions. Provenance of data can only be monitored and logged if the data is within the system. While this works, in reality, users move data in and out of the system where it is stored (E.g. cloud storage services). Hence, there is a need to develop mechanisms to allow tracking provenance of objects even when they are taken out of the boundary of the system. This is especially crucial for accountability of data. Such mechanisms will fill in the black-out periods in provenance information, where the data is being modified outside of the system boundaries. Tan et al. took a preliminary approach at addressing this issue and proposed a prototype, CloudDT [62], to track data outside of the cloud. The approach used a self-executing container to encapsulate the data before it exits the cloud. Users can only access the data through executing the self-executing container. The container will in-turn invokes a viewer program which then is responsible for logging user actions and sends it back to the cloud for provenance collection. Having said that, the approach still contains various flaws which need to be addressed.
- **Confidentiality and analysis**: The omission of information due to access control or lack of authorisation works for situations where the entire set of information has to be withheld from users. However, this is different in provenance. Users rely on provenance information to determine whether a piece of data can be trusted or if there are abnormalities with the security state of a system. In such a situation, users rely on a complete and reliable provenance to make the right analysis. By omitting certain parts of the information due to security reasons, it gives users the wrong impression of the provenance information presented. (E.g. if a provenance of the state is A-B-C, and B is hidden from the user, they might think that A transits to C directly and not factor B into the cause.) There is a need for new techniques and mechanisms that uphold confidentiality and still give users an estimated view of the provenance. Zhou et al.'s [42] proposed use of maybe state in upholding confidentiality while maintaining the structure of the provenance graph maybe a good basis for such a mechanism.
- **Partial or absence of provenance records**: In an ideal world, the provenance of all objects and systems are collected and made available for analysis. However, in reality, this is not the case due to reasons such as uptake of technology in commercial environments. Therefore, there is a need to address the issue of how provenance information can be inferred or extracted from the data itself or from partial provenance information. In doing so, the gap between legacy systems and provenance-aware systems can be bridged. This will allow the smooth integration of new generation provenance-aware systems to integrate into existing infrastructures and frameworks.

VI. Conclusions

As cloud computing matures, cloud users are becoming more aware of the importance of security in cloud systems and the need to secure their data in the cloud. Hence, an efficient system for forensic analysis, be it for security of systems or for determining the trustworthiness of data is required. Provenance is the key to building such systems.

In this survey paper, we surveyed past provenance solutions that was designed to support security analysis and data accountability, in a distributed systems context. We categorised these approaches into the granularity on which they function and discussed them in Section III.

A list of minimum requirements for a provenance system, designed to support security and data accountability is derived and presented in Section IV. This list covers five points; cross host tracking, decoupling, multi granularity, security mechanisms and interface for analysis. We argue that provenance systems will be able to provide complete and trustworthy provenance information for intuitive analysis and remain robust against changes in the system through those five points.

We also pointed out three areas which need to be addressed in next generation provenance solutions in Section V. Currently, these areas either do not have solutions that address them directly or only addresses part of the problems. It is our belief, that through closing the gaps identified in this survey paper, a complete provenance solution that is capable of supporting security and data accountability in the modern world can be achieved eventually.

REFERENCES

[1] "Where the battle lines are lines of code," http://www.facethefactsusa.org/facts/where-the-battle-lines-are-lines-of-code/ (Accessed: 1/07/2013), May 30 2013.

[2] R. K. L. Ko, M. Kirchberg, and B. S. Lee, "From System-centric to Data-centric logging - Accountability, Trust and Security in Cloud Computing," in *Defense Science Research Conference and Expo (DSR)*, 2011.

[3] C. Runnegar, "Security, Openness and Privacy," http://www.internetsociety.org/security-openness-and-privacy (Accessed: 1/07/2013), 29 Sep 2011.

[4] "The way forward: Preserving the Openness of the Internet," http://openmedia.ca/plan/economic-growth/way-forward (Accessed: 1/07/2013).

[5] "Amazon EC2," http://aws.amazon.com/ec2/ (Accessed: 1/07/2013). [Online]. Available: http://aws.amazon.com/ec2/

[6] "Apache Hadoop," http://hadoop.apache.org/ (Accessed: 1/07/2013).

[7] "Worldwide LHC Computing Grid," http://lcg.web.cern.ch/LCG/ (Accessed: 1/07/2013). [Online]. Available: http://lcg.web.cern.ch/LCG/

[8] "National Center for Biotechnology Information," http://www.ncbi.nlm.nih.gov/ (Accessed: 1/07/2013).

[9] R. K. L. Ko, B. S. Lee, and S. Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," in *Advances in Computing and Communication*, 2011.

[10] R. K. L. Ko, "Data Accountability in Cloud Systems," in *Security, Privacy and Trust in Cloud Systems*. Springer-Verlag Berlin Heidelberg, 2013.

[11] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," in *Proceedings of IEEE World Congress on Services (SERVICES'11)*, 2011.

[12] P. Buneman, S. Khanna, and W. C. Tan, "Why and Where: A Characterization of Data Provenance," in *Proceedings of 8th International Conference on Database Theory (ICDT'01)*, 2001.

[13] S. M. S. D. Cruz, M. L. M. Campos, and M. Mattoso, "Towards a Taxonomy of Provenance in Scientific Workflow Management Systems," in *IEEE Congress on Services*, 2009.

[14] R. K. L. Ko, G. Goh, T. Mather, S. Jaini, and R. Lim, "Cloud Consumer Advocacy Questionnaire and Information Survey," Cloud Security Alliance Cloud Data Governance Working Group, Cloud Security Alliance, Tech. Rep., 2011.

[15] R. A. Becker and J. M.Chambers, "Auditing of Data Analyses," in *Proceedings of 3rd International Workshop on Statistical and Scientific Database Management*, 1986.

[16] O. Biton, S. Cohen-Boulakia, S. B.Davidson, and C. S. Hara, "Querying and managing provenance through user views in scientific workflows," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE'08)*, 2008.

[17] S. Miles, E. Deelman, P. Groth, K. Vahi, G. Mehta, and L. Moreau, "Connecting Scientific data to scientific experiements with provenance," in *Proceedings of the third IEEE International Conference on e-Science and Grid Computing (e-Science'07)*, 2007, pp. 179–186.

[18] Y. L. Simmhan, B. Plale, and D. Gannon, "A Survey of data provenance in e-science," *ACM SIGMOD Record*, vol. 34, pp. 31–36, 2005.

[19] S. Miles, P. Groth, M. Branco, and L. Moreau, "The requirements of using provenance in e-science experiments," *Journal of Grid Computing*, vol. 5, pp. 1–25, 2007.

[20] T. J.Green, G. Karvounarakis, and V. Tannen, "Provenance Semirings," in *Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART symposium on Principes of database systems (PODS'07)*, 2007.

[21] B. Glavic and K. R. Dittrich, "Data Provenance: A Categorization of Exisitng Approaches," in *Proceedings of the 12th GI Conference on Datenbanksysteme in Business, Technologie und Web (BTW)*, 2007.

[22] W. C. Tan, "Research Problems in Data Provenance," *IEEE Data Engineering Bulletin*, vol. 27, pp. 45–52, 2004.

[23] W. C.Tan, "Provenance in Database:Past, Current and Future," *IEEE Data Engineering Bulletin*, vol. 30, pp. 3–13, 2007.

[24] J. Freire, D. Koop, E. Santos, and C. T. Silva, "Provenance for Computational Tasks: A Survey," *Computing in Science and Engineering, Journal*, vol. 10, pp. 11–21, 2008.

[25] R. Bose and J. Frew, "Lineage retrieval for scientific data processing: a survey," *ACM Computing Survey, Journal*, vol. 37, pp. 1–28, 2005.

[26] J. Cheney, S. Chong, N. Foster, M. Seltzer, and S. Vansummeren, "Provenance: A Future History," in *Proceedings of 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, 2009.

[27] O. Q. Zhang, M. Kirchberg, R. K. L. Ko, and B. S. Lee, "How to Track Your Data: The Case for Cloud Computing Provenance," in *Proceedings of IEEE 3rd International Conference on Cloud Computing Technology and Science (CloudCom'11)*, 2011.

[28] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying and Automating Data Derivation," in *Proceedings of 14th International Conference on Scientific and Statistical Database Management (SSDM'02)*, 2002.

[29] K.-K. Muniswamy-Reddy, D. A.Holland, U. Braun, and M. Seltzer, "Provenance-aware Storage Systems," in *Proceedings of the Conference on USENIX'06 Annual Technical Conference (ATEC'06)*, 2006.

[30] R. D.Stevens, A. J. Robinson, and C. A. Goble, "myGrid: Personalised BioInformatics on teh Information Grid," in *BioInformatics, Journal*, 2003.

[31] W. Zhou, L. Ding, A. Haeberlen, Z. Ives, and B. T. Loo, "TAP: Time-aware Provenance for Distributed Systems," in *Proceedings of USENIX Workshop on Theory and Practice of Provenance (TaPP'11)*, 2011.

[32] C. F. Reilly and J. F. Naughton, "Exploring Provenance in a Distributed Job Execution System," *Provenance and Annotation of Data, Journal*, pp. pp 237–245, 2006.

[33] M. N. Alpdemir, A. Mukherjee, N. W. Paton, A. A. A. Fernandes, P. Watson, K. Glover, C. Greenhalgh, T. Oinn, and H. Tipney, "Contextualised Workflow Execution in

myGrid," in *Proceedings of European Grid Conference, Lecture Notes in Computer Science*, 2005.

[34] P. Groth, M. Luck, and L. Moreau, "Formalising a protocol for recording provenance in grids," in *Proceedings of the UK OST e-Science second all hands meeting*, 2004.

[35] I. Souilah, A. Francalanza, and V. Sassone, "A Formal Model of Provenance in Distributed Systems," in *Proceedings of First Workshop on Theory and Practice of Provenance (TaPP'09)*, 2009.

[36] K.-K. Muniswamy-Reddy, U. Braun, D. A. Holland, P. Macko, D. Maclean, D. Margo, M. Seltzer, and R. Smogor, "Layering in Provenance Systems," in *Proceedings of the Conference of USENIX Annual Technical Conference (USENIX'09)*, 2009.

[37] P. Macko, M. Chiarini, and M. Seltzer, "Collecting Provenance via the Xen Hypervisor," in *Proceedings of USENIX Workshop on Theory and Practice of Provenance (TaPP'11)*, 2011.

[38] C. H. Suen, R. K. L. Ko, Y. S. Tan, P. Jagadpramana, and B. S. Lee, "S2Logger: End-to-End Data Tracking Mechanism for Cloud Data Provenance," in *Proceedings of 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'13)*, 2013.

[39] R. K. L. Ko, P. Jagadpramana, and B. S. Lee, "Flogger: A File-centric Logger for Monitoring File Access and Transfers with Cloud Computing Environments," in *3rd IEEE International Workshop on Security in e-Science and e-Research (ISSR'11), in conjunction with IEEE TrustCom'11*, 2011.

[40] S. T. King and P. M. Chen, "Backtracking Intrusions," in *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, 2003.

[41] S. T. King, Z. M. Mao, D. G. Lucchetti, and P. M. Chen, "Enriching Intrusion Alerts through Multi-host Causality," in *Proceedings of Network and Distributed System Security Symposium (NDSS'05)*, 2005.

[42] W. Zhou, Q. Fei, A. Narayan, A. Haeberlen, B. T. Loo, and M. Sherr, "Secure Network Provenance," in *Proceedings of 23rd ACM Symposium on Operating Systems Principles (SOSP'11)*, 2011.

[43] L. Lamport, R. Shostak, and M. Pease, "The Byzantine General Problem," *ACM Transaction on Programming Language and Systems (TOPLAS)*, vol. 4, pp. 382–401, 1982.

[44] P. Ruth, D. Xu, B. Bhargava, and F. Regnier, "E-notebook Middleware for Accountability and Reputation Based Trust in Distributed Data Sharing Communities," in *Proceedings of 2nd Intenational Conference on Trust Management. iTrust'04*, 2004.

[45] A. Singh, P. Maniatis, T. Roscoe, and P. Druschel, "Using Queries for Distributed Monitoring and Forensics," in *Proceedings of 1st ACM EuroSys European Conference on Computer Systems (SIGOPS'06)*, 2006.

[46] P. Reynolds, C. Killian, J. L. Wiener, J. C.Mogul, M. A.Shah, and A. Vahdat, "Pip: Detecting the Unexpected in Distributed Systems," in *Proceedings of 3rd Symposium on Networked Systems Design and Implementation (NSDI'06)*, 2006.

[47] B. T. Loo, T. Condie, J. M.Hellerstein, P. Maniatis, T. Roscoe, and I. Stoica, "Implementing declarative overlays," in *Proceedings of 20th ACM Symposium on Operating Systems principles (SOSP'05)*, 2005.

[48] "About viruses: PC Safety (Anti-rootkit utility)," http://support.kaspersky.com/5350?el=88446 (Accessed: 09/07/2013).

[49] D. Parker, "Windows NTFS Alternate Data Streams," http://www.symantec.com/connect/articles/windows-ntfs-alternate-data-streams (Accessed: 09/07/2013).

[50] A. Rosenthal, L. Seligman, A. Chapman, and B. Blaustein, "Scalable Access Controls for Lineage," in *Proceeding of First Workshop on Theory and Practice of Provenance (TAPP'09)*, 2009.

[51] A. Chebotko, S. Chang, S. Lu, F. Fotouhi, and P. Yang, "Secure Scientific Workflow Provenance Querying with Security Views," in *Proceedings of 9th IEEE International Conference on Web-Age Information Management (WAIM'08)*, 2008.

[52] R. Lu, X. Lin, X. Liang, and X. S. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," in *Proceedings of 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, 2010.

[53] L. M. R. Gadelha.Jr and M. Mattoso, "Kairos: An Architecture for Securing Authorship and Temporal Information of Provenance Data in Grid-Enabled Workflow Management Systems," in *Proceedings of IEEE 4th International Conference on e-Science (e-Science'08)*, 2008.

[54] A. Gehani and U. Lindqvist, "Bonsai: Balanced Lineage Authentication," in *23rd Conference of Computer Security Applications (ACSAC'07)*, 2007.

[55] J. Lyle and A. Martin, "Trust Computing and Provenance: Better Together," in *Proceedings of the 2nd Conference on Theory and Practice of Provenance (TaPP'10)*, 2010.

[56] X. Boyen and B. Waters, "Full-domain Subgroup Hiding and Constant-size Group Signatures," in *Proceedings of 10th International Conference on Practice and Theory in Public-key Cryptography (PKC'07)*, 2007.

[57] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," in *Proceedings of the 7th conference on Files and Storage Technologies (FAST'09)*, 2009.

[58] Q. Ni, S. Xu, E. Bertino, R. Sandhu, and W. Han, "An Access Control Language for a General Provenance Model," in *PRoceedings of 6th Very Large DataBase Workshop on Secure Data Management (VLDB'09)*, 2009.

[59] M. Nagappan and M. A.Vouk, "A Model for Sharing of Confidential Provenance Information in a Query Based System," *Provenance and Annotation of Data and Processes*, pp. 62–69, 2008.

[60] S. B.Davidson, S. Khanna, S. Roy, J. Stoyanovich, V. Tannen, and Y. Chen, "On Provenance and Privacy," in *Proceedings of 14th International Conference on Database Theory (ICDT'11)*, 2011.

[61] U. Braun and A. Shinnar, "A Security Model for Provenance," Harvard University Computer Science, Tech. Rep. TR-04-06, 2006 (Accessed: 1/07/2013).

[62] Y. S. Tan, R. K. L. Ko, P. Jagadpramana, C. H. Suen, M. Kirchberg, T. H. Lim, B. S. Lee, A. Singla, K. Mermoud, D. Keller, and H. Duc, "Tracking of Data Leaving the Cloud," in *Proceedings of IEEE 11th International Conference on Trust , Security and Privacy in Computing and Communications (TrustCom'12)*, 2012.