IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 12, 2020, accepted February 16, 2020, date of publication February 24, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975820

A Big Data Provenance Model for Data Security Supervision Based on PROV-DM Model

YUANZHAO GAO[1,2], **XINGYUAN CHEN**[1,2], **AND XUEHUI DU**[1]

[1]Zhengzhou Science and Technology Institute, Zhengzhou 450000, China
[2]State Key Laboratory of Cryptology, Beijing 100878, China

Corresponding author: Xingyuan Chen (chxy302@vip.sina.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803603.

ABSTRACT Nowadays, big data has become a hot research topic. It gives fresh impetus to the economic and social development. However, the huge value of big data also makes it the focus of attacks. Big data security incidents occur frequently in recent years. The security supervision capacities for big data do not match its important role. Data provenance which describes the origins of data and the process by which it arrived the current state, is an effective approach for data supervision. For the full use of provenance in big data supervision, a provenance model which defines the concepts used to represent the provenance types and relations is required to be built in advance, but current provenance models do not adapt to big data scenarios well. In this paper, we comprehensively consider the characteristics of big data and the requirements of data security supervision, extend the widely used provenance model PROV-DM by subtyping and new relation definition, and propose a big data provenance model (BDPM) for data supervision. BDPM model supports the provenance representation of various data types and diverse data processing modes to represent the entire data transformation process through different components in the big data system, and defines new relations to enrich provenance analysis functions. Based on BDPM model, we introduce the constraints that should be satisfied in the construction of valid provenance graph and present the data security supervision methods via provenance graph analysis. Finally, we evaluated the satisfiability of BDPM model through a case study.

INDEX TERMS Provenance model, big data, provenance representation, data security supervision.

I. INTRODUCTION

With the advent of big data era, data has become a kind of basic production factors as important as physical assets and human capital. Due to its huge value, big data faces serious security risks and challenges such as data leakage, malicious use, etc. Data security supervision is an important safeguard for data security. It analyzes and detects data security threats from the perspective of data rather than users or applications, which is significantly different in method and technology from traditional security supervision. Data provenance describes the origins of data and the process by which it arrived the current state [1]. It is an effective approach for data security supervision [2]. In recent years, with the development of big data and the rapid growth of data value utilization demand, big data provenance has gradually attracted the attention of researchers [3].

For the full use of provenance in big data supervision, it is necessary to build a provenance model in the context of big data at first. Provenance model formally defines the elements used in the provenance representation, the dependencies between them, and the rules established on the elements and dependencies to effectively express the evolution process of data. It provides guidance for provenance information collection, organization and analysis, and provides support for the sharing of provenance data [4]. Since provenance was proposed, the research on provenance model has been continuously promoted [5]. Current representative practices for provenance representation are Open Provenance Model (OPM) [4] and its successor PROV-DM model [6], which have been adopted by many provenance studies [7].

PROV-DM defines the provenance elements (entities, activities and agents) and their dependencies in a domain-agnostic way. It is able to represent the provenance of a broad diversity of things with some fixed aspects, such as digital objects, physical things, abstract concepts, etc. To encourage

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Agostino Ardagna.

widespread adoption, PROV-DM's design is intentionally minimal and lightweight. To build a big data provenance model for data security supervision, it is necessary to refine the element types defined by PROV-DM model according to the big data characteristics and data security supervision requirements. Moreover, PROV-DM model only defines the derivation relation between entities, but in the big data system, there are other relationships between entities, such as the coexistence relation between a file in the Hadoop Distributed File System (HDFS) and its corresponding local block files. The joint provenance analysis of coexisting entities plays an important role in detecting some types of data leakage [8]. Therefore, the provenance relationships defined by PROV-DM model need to be expanded for better data supervision.

To address the issues aforementioned, we extend PROV-DM model and propose a big data provenance model (BDPM) for data security supervision. The contributions of this paper are as follows:

(1) We analyze the big data characteristics [9], [10] and typical big data system technology frameworks [11], [12], review the research related to OPM and PROV-DM, and then present the requirements for building a big data provenance model.

(2) We propose a big data provenance model BDPM for data security supervision based on PROV-DM model. Preserving the core structure of PROV-DM, BDPM model refines the types of entities, activities and agents according to the data types of big data and components of big data system, and expands the provenance relationships to enrich provenance analysis functions. BDPM model supports the provenance representation of various types of data in multiple data organization layers and the provenance representation of the entire data transformation process through diverse storage, processing and communication components in the big data system.

(3) Based on BDPM model, we introduce the constrains that a valid provenance graph should satisfy and present the data security supervision methods based on provenance graph analysis, such as inference rule based analysis, vertical provenance analysis and horizontal provenance analysis.

The rest of the paper is organized as follows. Section II consists of related work. In Section III, we present the requirements for big data provenance model building. In Section IV, we describe BDPM model in detail, and introduce provenance graph definition and analysis. In Section V, we discuss the satisfiability of BDPM model. Finally, we conclude the paper and present future work in Section VI.

II. RELATED WORK

A. BIG DATA CHARACTERISTICS

It is necessary to understand the big data characteristics for building a practical big data provenance model. In fact, big data has been defined as early as 2001 [13]. Doug Laney, an analyst of META (presently Gartner) presented a "3Vs" model to describe the data increase in volume, velocity and variety [9]. In the "3Vs" model, great volume is the most significant feature of big data different from traditional data. High velocity means that data need to be collected and analyzed rapidly and timely. Variety indicates the various types of big data, which include structured, semi-structured and unstructured data. Although "3Vs" model was not originally used to define big data, Gartner and many other enterprises like IBM [14] and Microsoft [15] still used the "3Vs" model to describe the big data characteristics. In 2011, International Data Corporation (IDC) put forward a "4Vs" model [10], which added a feature value besides volume, velocity and variety. Value refers to the huge value but low value density of big data. In 2015, National Institute of Standards and Technology (NIST) presented another "4Vs" model [16], which added the feature variability besides the "3Vs" features [9]. Variability refers to the changes on the three other features that impact the data processing.

To deal with the characteristics of big data introduced above, many enterprises and research institutions put forward specific big data system technology framework to implement data processing [11]. Among them, NIST summarized existing frameworks and presented a technology- and infrastructure-agnostic big data reference architecture (NBDRA) [12], which has great influence internationally [17]. NBDRA is organized around five major roles: system orchestrator, data provider, big data application provider, big data framework provider and data consumer. The application provider is responsible for data collection, preparation, analysis, visualization, and access. The framework provider provides the infrastructure and data storage and processing platform. The types of data processing include batch processing, interactive processing and stream processing. The data storage approaches include indexed storage and file system style storage. Intended to enable big data professionals (such as data scientists, data architects, etc.) to develop solutions to issues that require diverse approaches due to the convergence of big data characteristics, NBDRA supports a variety of business environments and is helpful for us to build a big data provenance model.

B. OPM AND PROV-DM

Since the interest in provenance in data management, e-science and other fields is growing, the first International Provenance and Annotation Workshop (IPAW'06) was held in 2006, involving participants interested in the challenging issues of data provenance. During a session for provenance standardization, the first Provenance Challenge (PC1) was born. Via PC1, PC2 and PC3, the research community analyzed the capacities of various provenance systems and the expressiveness of their provenance representations, reached an agreement on the core representation of provenance, released OPM v1.00 [18], and then discussed and improved it to form OPM v1.1 [4], the latest version of OPM. OPM is a provenance model built in a precise and technology-agnostic manner. It defines a core set of rules that identify the valid inferences that can be made on provenance graph,

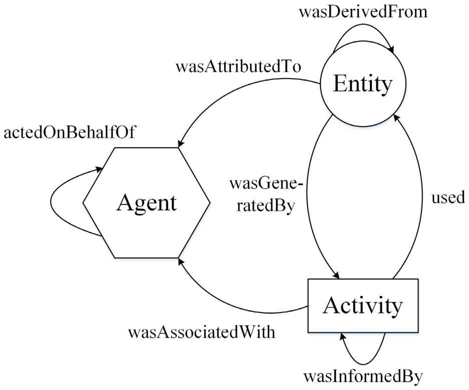

FIGURE 1. Core structure of PROV-DM.

allows the provenance information to be exchanged between different systems, and allows developers to build provenance tools based on the model.

After the fourth and last Provenance Challenge, World Wide Web Consortium (W3C) Provenance Working Group took over the research of the Provenance Challenge Workshop, and released PROV-DM model based on OPM. PROV-DM distinguishes core structures, forming the essence of provenance information, from extended structures catering for specific application scenarios. The core structure consists of three types of nodes: entity, activity and agent, and seven types of edges representing provenance relationships: generation, usage, communication, derivation, attribution, association and delegation, as shown in Fig. 1. In the provenance graphs of this paper, entities are represented by circles, activities are represented by squares and agents are represented by hexagons. The extended structure extends the core structure to support broader use of provenance by the way of subtyping, optional identification, new relation definition, etc. However, the extended structure is still highly abstract and needs to be further extended according to the terminologies in big data. For example, Crawl *et al.* [19] proposed a provenance model to describe the data processing in MapReduce based on OPM and MapReduce terminologies, but the model is not generic-purpose.

In this paper, preserving the PROV-DM core structure and partial extended structure, we further extend PROV-DM model via subtyping and new relation definition according to the big data characteristics and security supervision requirements.

III. REQUIREMENTS FOR BIG DATA PROVENANCE MODEL BUILDING

Considering the big data characteristics and data supervision requirements, we present the requirements for big data provenance model building.

A. GENERALITY

The Big data ecosystem widely collects heterogeneous data from different data sources, and support diverse data storage and processing modes for different application scenarios. The big data provenance model should support effective representation of the characteristics of heterogeneous data, and diverse data storage and processing modes.

B. COMPLETENESS

A data object and the data objects derived from it may go through different data storage and processing components. Therefore, besides generality, the provenance model should support effective representation of the complete process of data transformation through diverse data storage, processing and communication components.

C. MULTI-GRANULARITY

At present, most approaches on the use of provenance for data supervision are coarse-grained, that is, the whole file or dataset is taken as the supervision target [20], [21]. However, these approaches lack fine-grained provenance representation that is difficult for them to detect sensitive records leakage [22], [23], de-anonymization [24], and other malicious operations at record level in the data analysis process. Therefore, in addition to supporting coarse-grained provenance representation to monitor the abnormal flow of data, a provenance model for data supervision also needs to support fine-grained provenance representation to monitor whether a data analysis process complies with the data security and privacy policies.

D. MULTI-LAYER

First, the big data system presents multi-layer feature on data organization. For example, a NoSQL database is located on a distributed file system (DFS), and a DFS file is distributed across multiple local file systems (LFS). The malicious behavior against big data can be implemented at all layers, and the behavior implemented in the lower-layer data organization system cannot be detected in the upper-layer system. For example, a malicious user can steal HDFS data directly from underlying LFS without leaving traces in the audit or monitoring information of HDFS [25]. Therefore, it is necessary for the provenance model to express the relationship of related data between different data organization systems so as to realize the joint analysis of operations on them to detect possible abnormal operations.

Second, the data organization also presents multi-layer feature inside a data organization system, such as the tree structure of file system, the table, partition and row of database. The provenance model should enable the representation of relationships between these data objects.

Third, the smallest management unit of a data organization system may still have internal structure, such as a file consisting of records in some form. The provenance model should support the representation of the structure inside the smallest unit for fine-grained data supervision.

E. CONCISENESS

The provenance data of big data system is also of great volume, and even larger than the original data in some cases [26].

While providing sufficient semantic information, the model should keep concise to reduce the overhead of provenance data collection and storage, and reduce the complexity of provenance data analysis.

IV. BDPM MODEL

Comprehensively considering PROV-DM model, big data characteristics, NBDRA and data security supervision, we extend PROV-DM and build BDPM model. BDPM involves the data application, processing, and storage procedures in NBDRA, and focuses on the provenance information about the data operations inside the big data cluster and the data interaction between the cluster and outside. Meanwhile, BDPM covers the provenance representation of the data on cluster hosts associated with the data in the cluster.

A. NODES

PROV-DM defines three types of top-level nodes: entity, activity and agent. We refine the three types of nodes via subtyping according to the big data characteristics while ensuring the generality of the model to allow developers to apply it to specific big data systems. Due to space limitation, we do not introduce all the formal definitions of nodes and dependencies. But we build a big data provenance ontology based on BDPM model. The ontology represents the attributes of each node and dependency, and it can be found at https://github.com/gewuzhao/big-data-provenance-ontology.

1) ENTITY

In BDPM, entity refers to the digital object, which is formally represented as: *Entity*: <*ID*, [$attr_1 = value_1, \cdots$]>, where *ID* is the identifier of *Entity*, the attributes in the square bracket is optional. In digital objects, data which is the core of provenance tracing includes file data, indexed data, streaming data and message in the big data scenario. To completely express the origin and transformation information of data, the entity also include job, task and network object. In addition, developers can add environment information of the cluster if needed, such as defining a subtype "cluster" under "entity".

a: DATA

(1) File data

File data is a common data type. The structure of entity types related to file data is shown as Fig. 2.

According to the layer between file systems, the file can be classified as DFS (such as HDFS) file and LFS (such as Ext4) file. Inside a file system, the data is organized as a hierarchy of directories and files. It is known that DFS is the basic data organization method of big data system. The reason we pay attention to the provenance information of local files is that they are the sources of some data in the cluster on the one hand. Defining the provenance representation of LFS file ensures the provenance completeness of the data inside the cluster. On the other hand, DFS is established upon LFSs, a DFS file is stored in LFS in some form. For example, a HDFS file is stored in DFS in the form of block files which are named by the block numbers the HDFS file uses. The construction and analysis of provenance of local block files is an important means of HDFS data supervision [25].

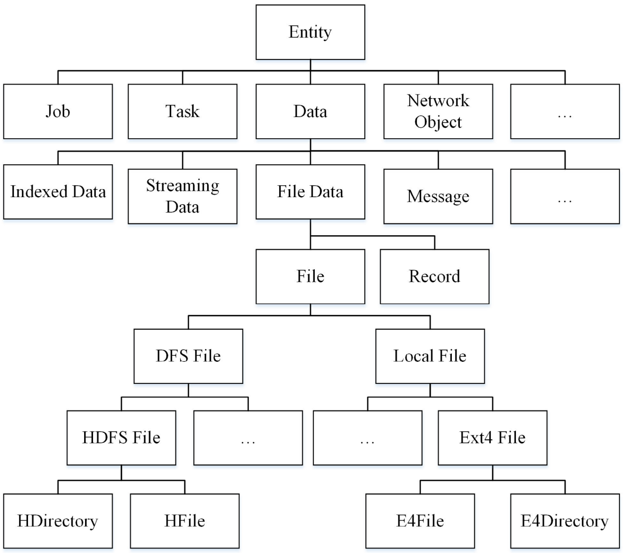

FIGURE 2. Structure of entity types related to file data.

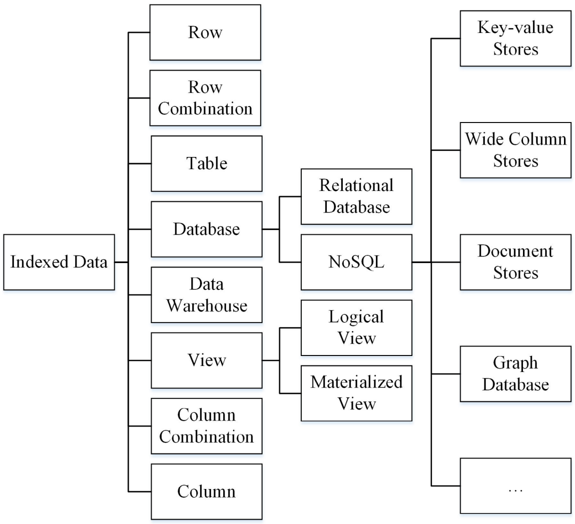

FIGURE 3. Structure of indexed data types.

According to internal data structure, the file can be classified as semi-structured file (such as XML) and unstructured file (such as PDF) [27]. For a semi-structured file, it can be further divided into a set of records which can be used for fine-grained supervision.

(2) Indexed data

Indexed data refers to the data stored in database or data warehouse. According to the hierarchy of database/data warehouse, the indexed data can be divided into database/data warehouse, table/view, row/column combination and row/column. The structure of indexed data types is shown as Fig. 3.

Database is usually used for online transaction processing to enable efficient access to transactions and maintain data integrity in multi-user scenarios, such as Oracle, MySQL, Microsoft SQL Sever, etc. Data warehouse is usually used for online analytical processing to enable complex queries across large-scale data with longer query delay compared with database [28], such as Apache Hive [29]. The database can be divided into relational database and NoSQL according to its data model. NoSQL can be further classified as key-value stores, wide column stores, document stores, graph database, etc.

The next layer of database/data warehouse is table/view. A view is created from one or several basic tables by a set of query instructions for the sake of security or query simplification. The view can be further divided into logical view and materialized view. A logical view is a virtual structure that does not store the data the view represents, while a materialized view stores actual data.

Row and column are respectively the basic data and attribute unit in database or data warehouse. The row/column combination is created for the reason of performance, purpose, etc., such as the partition (row combination) of Hive and the column family (column combination) of the wide column store Apache HBase [30].

(3) Streaming data

Streaming data is a real-time and large-scale data sequence that is continuously generated and sequentially transferred from source to destination [31]. It can be formalized as: $<\cdots, a_{t-1}, a_t, a_{t+1}, \cdots>$, where t represents timestamp and a represents the tuple in the stream.

According to organization mode, the streaming data can be divided into stream, stream window and stream tuple, where the stream window is a group of stream tuples within a given time window [32]. The representation of stream and stream tuple is the same as entity. The stream window is formalized as: *StreamWindow*: <*ID*, t_{start}, t_{end}, [$attr_1 = value_1, \cdots$]>. The size of stream window can be set according to the supervision granularity, or the stream tuple can be taken as the supervision unit. The source and destination of streaming data are also entities and are expressed via provenance relationship. Thus, they are not included in the attributes of streaming data. In addition, according to the source of streaming data, it can be divided into file stream, network stream, message stream, etc.

(4) Message

Big data system consists of diverse data storage and processing components. These components are not independent of each other, but are interconnected in some way to achieve richer functions. As an intermediary for data interaction, the messaging system (such as Apache Kafka [33]) provides approaches to promote the communication between these components. It can receive various types of data from different data sources and distribute them to different destinations.

Messages are organized differently in different messaging systems. Taking Kafka as an example, the message is organized via a three-layer structure of topic, partition and record. To ensure the generality of BDPM, the representation of general messages is the same as entity. For Kafka message, we define the attributes of the three data types in the provenance ontology.

b: JOB AND TASK

Job and task are used to express the provenance information of big data processing whose typical feature is distributed and parallel processing. Popular big data processing frameworks (such as MapReduce, Spark, Storm, Flink, etc.) conduct data processing by the way of job and task. Job is the basic unit submitted by user and task is the basic data processing unit of the frameworks above. A job is divided into a set of tasks which execute in parallel. The job is formalized as: *Job*: <*ID*, *name*, [$parameter_1 = arg_1, \cdots$], t_{start}, t_{end}, [$attr_1 = value_1, \cdots$]>. The task is formalized as: *Task*: <*ID*, t_{start}, t_{end}, [$attr_1 = value_1, \cdots$]>. Analyzing the provenance information of jobs and tasks promotes the detection of compromised computing nodes and the verification of computing results [34]. For example, *input* in the job *parameters* is set by the agent who submits the job, while the actual input can be determined by analyzing the provenance related to the job. If the actual input is inconsistent with the set input, an anomaly is detected.

c: NETWORK OBJECT

Network object includes web page, network resource, bookmark, cookie, website, etc. [35]. According to the application scenario of BDPM model, network object is mainly used to represent the source and destination of some data objects in the cluster. Thus, we only define a subtype "network address" under the "network object". It is formalized as: *NetworkAddress*: <*ID*, *protocol*, *IP*, [*port*], [$attr_1 = value_1, \cdots$]>, where *protocol* represents data transfer protocol, such as HTTP, FTP, Remote Procedure Call (RPC), etc.

2) ACTIVITIES

An activity is something that occurs over a period of time and acts upon or with entities [36], which is formalized as: *Activity*: <*ID*, *IP*, [$parameter_1 = arg_1, \cdots$], *status*, [$attr_1 = value_1, \cdots$]>. Users generally interact with the big data system through network. *IP* represents the IP address from where the user launched the activity and can be used for event backtrack. *parameters* can be used to record the attribute changes of the entity upon which the activity acted, such as permission, owner, etc. *status* represents the current state of an activity, including success, failure and in process. Failed activities provide important clues for anomaly detection. It should be noted that we do not define the time information in the formal representation of activity, because some activities will last for some time, and thus have start time and end time, while some activities start and end in an instant so that we just need to record their occurrence time. The time information will be described in the specific definition of each activity.

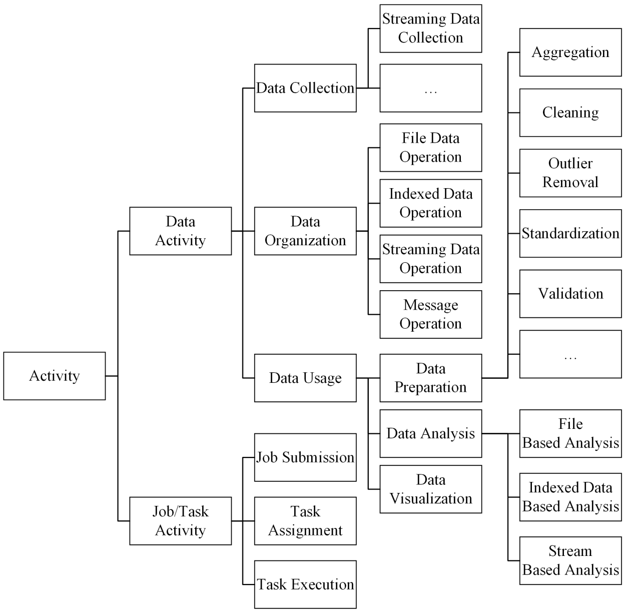

FIGURE 4. **Structure of activity types.**

According to entity types, we define the provenance representation of activities related to data, jobs and tasks. The structure of activity types is shown as Fig. 4. Job and task related activities include job submission, task assignment and task execution. These activities which express the main workflows of the distributed data processing framework are easy to understand. We do not describe them in detail and focus on the data related activities.

We classify data related activities into three categories: data collection, data organization and data usage. Data organization includes data storage, access, distribution, and deletion related activities. Data usage includes data preparation, analysis and visualization. The entire life cycle of data can be covered by these activities.

a: DATA COLLECTION

From the perspective of cluster, data collection represents the activities which extract or accept data from outside the big data cluster and will trigger data creation activities inside the cluster. The provenance information of data collection can be used for data quality or credibility assessment, and can be used as the medium for further tracking the transformation process of data before entering the cluster.

b: DATA ORGANIZATION

Data organization refers to the operations act upon data directly, such as creation, reading, writing, deletion, etc. Data collection and usage related activities will be eventually transformed into operations in the data organization activity. According to data types, data operation includes file data operation, indexed data operation, streaming data operation and message operation. The indexed data operation can be further divided into database/data warehouse, table/view, row/column combination and row/column operation.

c: DATA USAGE

Data usage refers to the activities aimed at extracting information and mining knowledge from the original data, includes data preparation, analysis and visualization.

Data preparation includes data validation, cleaning, outlier removal, standardization, aggregation, etc. [12]. According to the data organization mode, data analysis includes file-based, indexed-data-based and streaming-data-based analysis, that is, batch processing, interactive processing and stream processing. Data visualization represents the activity that displays the data or corresponding analysis results in the form of chart, report, etc.

3) AGENTS

An agent is something that bears some form of responsibility for an activity occurring, for the existence of an entity, or for the activities of other agents [36]. It is formalized as: *Agent*: <*ID*, *name*, [$attr_1 = value_1, \cdots$]>.

The agent includes single user, group (such as company, community) and software agent. According to the types of entities and activities, the three types of agents have three common subtypes: data producer, data consumer and job submitter. In addition, the software agent also includes data collector, job master and task worker.

Data producer, also known as data provider, can be a user, a group, or a software agent. For example, a data producer may be a server continuously generating data. Similarly, a data consumer may be a software agent, such as a streaming data processing engine consuming the messages stored in Kafka.

Data collector mostly refers to the data collection software in the big data scenario, such as the streaming data collector Apache Flume [37], Kafka, etc.

In the big data processing frameworks such as MapReduce, a job is divided into a set of tasks which execute in parallel. The job master is responsible for task creation and assignment. The task worker is responsible for task execution. They are formalized as: *JobMaster/TaskWorker*: <*ID*, *name*, *IP*, t_{start}, t_{end}, [$attr_1 = value_1, \cdots$]>. The provenance information of job master and task worker can be used to detect compromised nodes [34].

Job creation is triggered by job submitter, who can be a user, a group, or an application providing data service to users.

B. DEPENDENCIES

As the edges of provenance graph, dependencies represent the provenance relationships among entities, activities and agents. The provenance structure of BDPM is shown as Fig. 5. A dependency can be regarded as an influence taken by the influencer on the influenced. Thus, the dependency is formalized as: *Dependency*: <[*ID*], *ifd*, *ifr*, [$attr_1 = value_1$, $\cdots$]>, where *ID* is an optional identifier, *ifd* and *ifr* represent

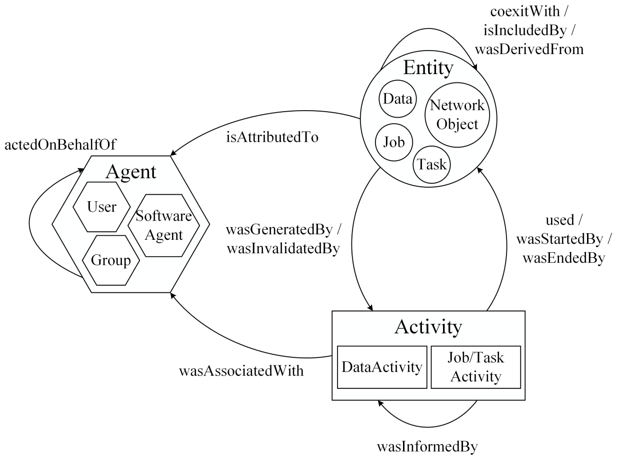

FIGURE 5. **Dependencies structure of BDPM model.**

the identifier of the influenced and the influencer respectively. The edges of provenance graph point from the influenced to the influencer. Taking the three types of nodes as influencer, the dependency can be divided into three types.

1) DEPENDENCIES WITH ENTITIES AS INFLUENCERS

In the dependencies with entities as influencers, we adopt four kinds of dependencies in PROV-DM: derivation, usage, end and start, and add two kinds of dependencies according to the characteristics of big data organization: coexistence and inclusion.

Derivation, coexistence and inclusion represent the dependencies between entities, and are formalized as: *Derivation/Coexistence/Inclusion*: <[ID], e_1, e_2, [$attr_1 = value_1, \cdots$]>.

Derivation represents that a new entity (the influenced) is generated by an activity via using an existing entity (the influencer).

Coexistence represents the dependency between an entity in the upper-layer data organization system and its corresponding entities in the lower-layer system. In a coexistence relation, the influenced is the lower-layer entity and the influencer is the upper-layer entity. For example, a HDFS file f_1 coexists with a LFS file f_2, if f_1 is deleted, f_2 will be deleted, but the reverse is not true. The joint analysis of operations on coexisting data promotes the detection of possible abnormal operations in the lower-layer data organization system.

Inclusion on the one hand represents the dependencies between entities at different layers in a data organization system, such as the dependencies between a directory and the files it contains. This relation facilitates provenance analysis based on how data is organized, such as detecting whether the lower-layer files of a directory have suffered some abnormal operation uniformly [38]. On the other hand, if the smallest unit of a data organization system still has internal structure, inclusion represents the dependencies between the unit entity with the smaller entities it contains, such as the dependencies between a HDFS file and the records it contains. The relation can be used to express the origin of the small-grained entities which are not directly managed by the data organization system. Moreau [4] represent this relation by subtyping of derivation, but we define a specific dependency type here. In an inclusion relation, the lower/smaller is the influenced and the upper/larger is the influencer.

Usage, start and end represent the influence of entities on activities, and are formalized as: *Usage/Start/End*: <[ID], a_1, e_1, [$attr_1 = value_1, \cdots$]>. Usage indicates the beginning of using an entity by an activity, that is, the activity begins to be affected by the entity. "Usage" is different from "data usage" defined earlier in the article. Data usage represents a type of activity acting upon data, while usage represents a provenance relation between activities and entities. Start and end indicate that the start and end of an activity are triggered by an entity.

2) DEPENDENCIES WITH ACTIVITIES AS INFLUENCERS

Dependencies with activities as influencers include generation, invalidation and communication.

Generation and invalidation represent the influence of activities on entities, where invalidation includes destruction, cessation, etc. Communication represents the dependencies between activities. It indicates that the execution of an activity is dependent on some unspecified entity generated by another activity, that is, the latter generates an entity and the entity triggers the start of the former.

3) DEPENDENCIES WITH AGENTS AS INFLUENCERS

Dependencies with agents as influencers include attribution, association and delegation. Attribution indicates the ascribing of an entity to an agent. Association represents that an agent is responsible for the execution of an activity. Delegation is the assignment of authority and responsibility by an agent (such as ag_1) to another agent (such as ag_2) to carry out a specific activity (such as a_1) as a delegate. It can be formalized as: *Delegation*: <[ID], ag_2, ag_1, a_1, [$attr_1 = value_1, \cdots$]>.

C. PROVENANCE GRAPH DEFINITION AND ANALYSIS

1) PROVENANCE GRAPH DEFINITION

A BDPM provenance graph consists of entities, activities, agents and the dependencies between them. A valid provenance graph should satisfy the uniqueness constraints, event ordering constraints, type constraints and impossibility constraints proposed for PROV-DM model [39].

Uniqueness constraints require that all node and edge instances in a provenance graph can be uniquely identified. The nodes of BDPM provenance graph are identified by unique identifiers. The identifiers of edges are optional, if not set, an edge is uniquely identified by the influenced and influencer it connects.

Event ordering constraints refer to the chronological order that correlative entities, activities, or agents must follow. For example, an entity can only be used after it has been generated and before it is invalidated.

Type constraints require that for a node instance (such as entity e_1) or a dependency instance (such as $Communication_1$: <a_1, a_2>), the set of types associated with e_1 must include type "Entity", and the set of types associated with a_1 and a_2 must include type "Activity".

Impossibility constraints require that certain patterns of statements never appear in valid provenance instances, such as required field missing, causality violation, the above three constraints violation, etc.

In addition, Moreau [4] proposed the concept of overlapping account to support the different explanations of the same process at different observation granularities or from different viewpoints in a single graph. BDPM graph supports overlapping account.

2) PROVENANCE GRAPH ANALYSIS

The data security supervision based on BDPM graph analysis can detect abnormal data operations via the semantic/attribute information or structural information of the graph. Semantic information based supervision analyzes whether the attributes of nodes and edges are consistent with expectation. Structural information based supervision includes two methods. One is to adopt inference rules which can be directly obtained from specific provenance relationships to detect abnormal operations. The other is to establish more complex anomaly detection algorithms by the statistical analysis or mining of provenance graph to determine whether the characteristics or trend of the graph is abnormal.

The provenance relationship that can be used to establish inference rules for anomaly detection is coexistence. Considering the data backup mechanism used for disaster recovery by the big data system, we suppose that lower-layer entities E_2, E_3 and E_4 coexist with an upper-layer entity E_1, as Fig. 6 (a) shows. Under normal circumstances, if an entity reading operation P_1 used E_1 at T_1, we can obtain that a corresponding entity reading operation P_2 used E_2, E_3 or E_4 at the same time, as Fig. 6 (b) shows. In reverse, if P_2 used one of E_2, E_3 and E_4 (such as E_2) at T_1, we can obtain that P_1 used E_1 at the same time, as Fig. 6 (c) shows. If there is only an edge representing that P_2 used E_2 without a corresponding edge representing that P_1 used E_1, an anomaly is detected, like Fig. 6 (d). The rule is not limited to reading activities, but applicable to all activities acting upon the contents of the entity. If the activity only affects the attributes of the entity, the rule does not work.

The establishment of more complex anomaly detection rules is beyond the scope of this paper, but we introduce two strategies for provenance analysis: vertical analysis and horizontal analysis. Vertical provenance which represents the transformation process of a specific data object can be used for abnormal data flow detection and access control policies building [20], [40], [41]. In some cases, the vertical provenance of several data objects which are included by the same upper data object in a data organization system can be analyzed together to detect abnormal data operations [38]. Horizontal provenance which represents the data objects used by the same user or software agent can be used to determine whether the behavior of the user or software agent is abnormal [42], [43].

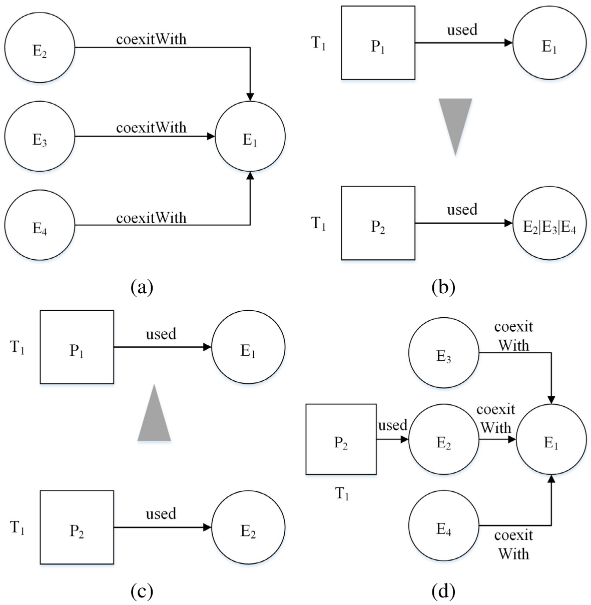

FIGURE 6. Inference rule for activities of coexisting entities.

V. DISCUSSION

In this section, we analyze the satisfiability of BDPM model based on the big data provenance model building requirements proposed.

For generality and completeness, we analyze existing research on real-time log collection and processing [44]– [46] and present a case study for test:

A data provider *Admin* adopted the distributed streaming data collection system Apache Flume to collect the system log $E4File_1$ stored in the cluster's local file system Ext4 in real time. The collected streaming data $FileStream_1$ was then temporarily stored in the $Partition_1$ of Kafka $Topic_1$. A Spark Streaming job read the data in Kafka $Topic_1$ via a receiver task and processed $KafkaStream_1$ received via a computing task. The computing result was stored into $HBaseTable_1$ of HBase for interactive query and stored into $HiveTable_1$ of Hive for batch processing respectively. A data consumer $User_1$ queried the data in $HiveTable_1$. The query was then converted into a MapReduce job. The job took HDFS file $HFile_2$ which coexists with $HiveTable_1$ as input, and output the result into $HFile_3$. The provenance graph of the entire procedure is shown as Fig. 7.

The scenario above represents a real-time log data collection and processing procedure, involving heterogeneous data types (file data, streaming data, indexed data and message), diverse data processing modes (stream processing, data warehouse query and batch processing), and data collecting and communication components. The effective representation of the scenario by the graph illustrates the generality

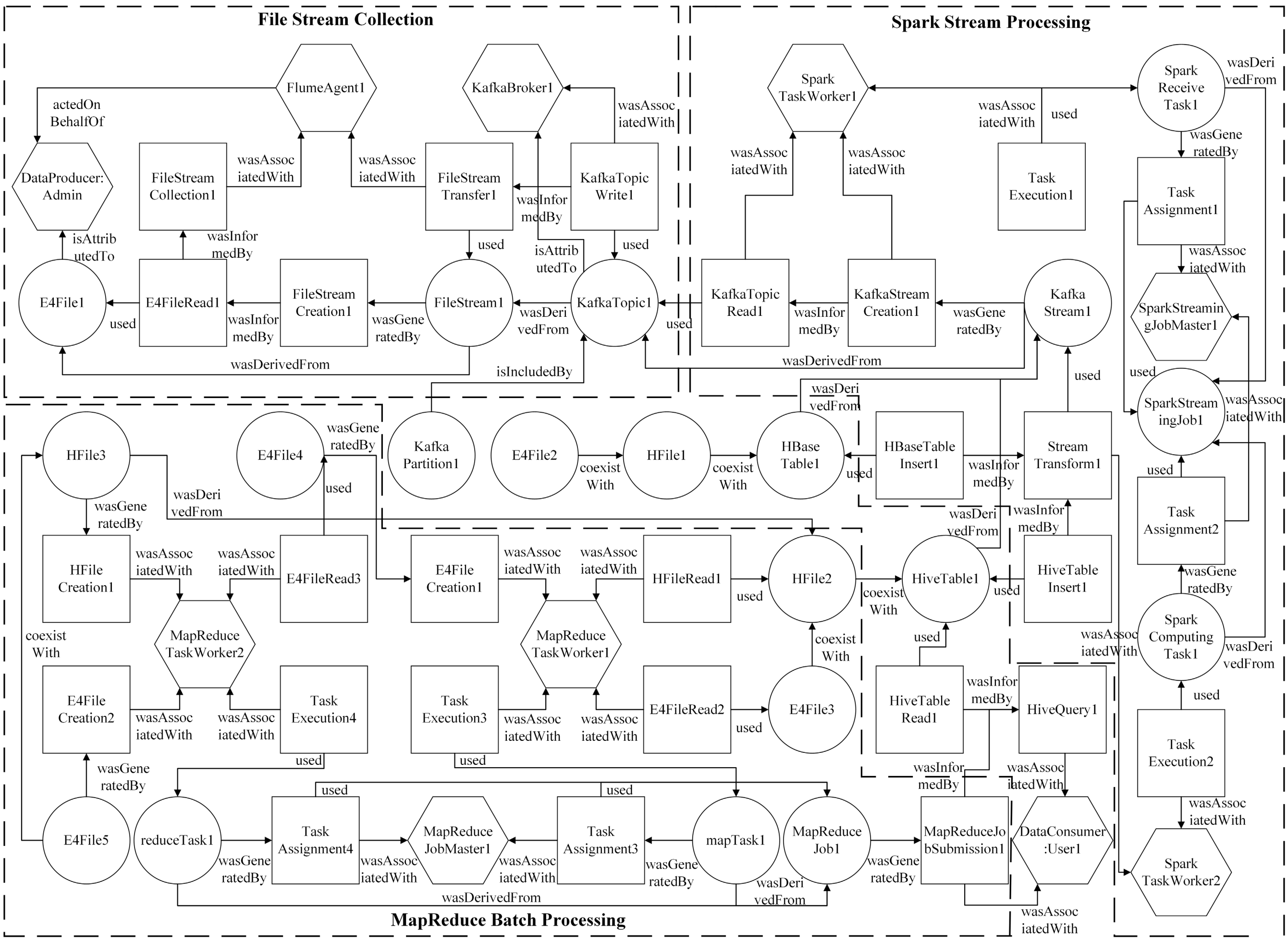

FIGURE 7. Provenance graph of real-time log data collection and processing.

of BDPM model. In the procedure, $E4File_1$ was transformed into $FileStream_1$, $KafkaTopic_1$, $KafkaStream_1$, $HBaseTable_1$, $HiveTable_1$ and $HFile_3$ in turn. It illustrates that BDPM model can effectively represent the entire procedure of data transformation through different data storage, processing and communication components. Based on the semantic and structural information of the provenance graph, we can detect whether a data object has suffered some abnormal operation in the lifecycle, such as whether it was stored in an unexpected location, whether it was used in an unallowed way, etc.

In addition, it should be noted that for expression simplification, the scenario does not contain all possible dependencies. For example, a topic may have several partitions and be maintained by several Kafka Brokers (a kind of software agent in Kafka). A Spark or MapReduce job will be divided into several tasks if the input is large. A HBase or Hive table will be divided into several partitions and then stored in several HDFS files for performance improvement. But these occasions do not affect the proof of generality and completeness of BDPM model.

For multi-granularity and multi-layer, BDPM model refines the entity types of PROV-DM, supports the provenance representation of data at different layers of diverse data organization systems and the provenance representation of data types inside the smallest management unit of the data organization system. Meanwhile, BDPM model defines the relation "inclusion" to represent the dependencies between entities at different layers in the data organization system. The refined entity types and "inclusion" relation allows data supervision at different granularities. BDPM model defines the relation "coexistence" to represent the dependencies between coexisting entities belonging to different data organization systems. The relation can be used for joint analysis of operations on coexisting data to detect possible abnormal operations.

The conciseness is reflected in that BDPM model only defines provenance and safety supervision related information for the representation of nodes and edges, and does not include all attribute information to achieve very accurate definition of concepts.

VI. CONCLUSION AND FUTURE WORK

Building a provenance model to realize effective representation of provenance information is the foundation for the full use of provenance in data security supervision. However, current provenance models do not adapt to big data scenarios well. In this paper, considering the big data characteristics and data security supervision requirements, we propose a big data provenance model BDPM for data security supervision based on PROV-DM model. Via node type refinement and dependency type expanding, BDPM model supports the provenance representation of various types of data in multiple data organization layers and the representation of the entire data transformation process through diverse storage, processing and communication components in the big data system. Based on BDPM model, we introduce the constraints that should be satisfied in the construction of valid provenance graph and present the data security supervision strategies and methods based on the provenance graph analysis.

In this paper, BDPM model is used for data security supervision, but it can also be used in other scenarios, such as data quality assessment, scientific experiment procedure analysis, etc. Researchers can select partial elements of BDPM model or further extend it to apply to specific scenarios. In future, we will further improve BDPM model and the corresponding provenance ontology with deeper research on big data and carry out works on big data provenance tracking, query and analysis based on the model.

REFERENCES

[1] P. Buneman, S. Khanna, and W.-C. Tan, "Why and where: A characterization of data provenance," in *Proc. Int. Conf. Database Theory (ICDT)*, London, U.K., 2001, pp. 316–330.

[2] B. Glavic. (2012). *Big Data Provenance: Challenges Implications for Benchmarking*. [Online]. Available: http://cs.iit.edu/~dbgroup/assets/pdfpubls/G13.pdf.

[3] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, Aug. 2012.

[4] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. Van den Bussche, "The open provenance model core specification (v1.1)," *Future Gener. Comput. Syst.*, vol. 27, no. 6, pp. 743–756, 2011.

[5] S. S. Sahoo, "Provenance algebra and materialized view-based provenance management," in *Proc. 2nd Int. Provenance Annotation Workshop*, Salt Lake City, UT, USA, 2008, pp. 531–540.

[6] P. Missier, K. Belhajjame, and J. Cheney, "The W3C PROV family of specifications for modelling provenance metadata," in *Proc. 16th Int. Conf. Extending Database Technol. (EDBT)*, Genoa, Italy, 2013, pp. 773–776.

[7] B. Pérez, J. Rubio, and C. Sáenz-Adán, "A systematic review of provenance systems," *Knowl. Inf. Syst.*, vol. 57, no. 3, pp. 495–543, Feb. 2018.

[8] O. Alabi, J. Beckman, M. Dark, and J. Springer, "Toward a data spillage prevention process in Hadoop using data provenance," in *Proc. Workshop Changing Landscapes HPC Secur. (CLHS)*, Portland, OR, USA, 2015, pp. 9–13.

[9] D. Laney, "3-D Data Management: Controlling Data Volume, Velocity and Variety," META Group Inc., Stamford, CT, USA, Tech. Rep. 949, 2001.

[10] J. Gantz and D. Reinsel, "Extracting value from chaos," Int. Data Corp., Framingham, Ma, USA, Tech. Rep. IDC iView 1142, 2011.

[11] S. Mishra and W. Chang, "NIST big data interoperability framework: Volume 5, architectures white paper survey," Nat. Inst. Std. Tech., Gaithersburg, MD, USA, Tech. Rep. NIST.SP.1500-5, 2015.

[12] O. Levin, D. Boyd, and W. Chang, "NIST big data interoperability Framework: Framework: Volume 6, reference architecture," Nat. Inst. Std. Tech., Gaithersburg, MD, USA, Tech. Rep. NIST.SP.1500-6, 2015.

[13] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[14] P. Zikopoulos, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. New York, NY, USA: McGraw-Hill, 2011, pp. 3–13.

[15] E. Meijer, "The world according to LINQ," *Commun. ACM*, vol. 54, no. 10, pp. 45–51, 2011.

[16] N. Grady and W. Chang, "DRAFT NIST big data interoperability framework: Volume 4, security and privacy," Nat. Inst. Std. Tech., Gaithersburg, MD, USA, Tech. Rep. NIST.SP.1500-4, 2015.

[17] X. Chen, Y. Gao, H. Tang, and X. Du, "Research progress on big data security technology," *SCIENTIA SINICA Informationis*, vol. 50, no. 1, pp. 25–66, Jan. 2020, doi: 10.1360/N112019-00077.

[18] L. Moreau. (2007). *The Open Provenance Model (v1.00)*. [Online]. Available: http://eprints.ecs.soton. ac.uk/14979/1/opm.pdf

[19] D. Crawl, J. Wang, and I. Altintas, "Provenance for MapReduce-based data-intensive workflows," in *Proc. 6th Workshop Workflows Support Large-Scale Sci. (WORKS)*, Seattle, WA, USA, 2011, pp. 21–30.

[20] A. Bates, D. J. Tian, K. R. B. Butler, and T. Moyer, "Trustworthy whole-system provenance for the Linux kernel," in *Proc. 24th USENIX Secur. Symp. (USENIX Security)*, Washington, DC, USA, 2015, pp. 319–334.

[21] O. Q. Zhang, R. K. L. Ko, M. Kirchberg, C. H. Suen, P. Jagadpramana, and B. S. Lee, "How to track your data: Rule-based data provenance tracing algorithms," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Liverpool, U.K., Jun. 2012, pp. 1429–1437.

[22] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: Privacy-aware data intensive computing on hybrid clouds," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, 2011, pp. 515–526.

[23] K. Y. Oktay, "SEMROD: Secure and efficient MapReduce over hybrid clouds," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, Melbourne, VIC, Australia, 2015, pp. 153–166.

[24] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. IEEE Symp. Secur. Privacy*, May 2008, pp. 111–125.

[25] Y. Gao, X. Fu, B. Luo, X. Du, and M. Guizani, "Haddle: A framework for investigating data leakage attacks in Hadoop," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[26] J. Wang, D. Crawl, S. Purawat, M. Nguyen, and I. Altintas, "Big data provenance: Challenges, state of the art and opportunities," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2015, pp. 2509–2516.

[27] R. Sint, "Combining unstructured, fully structured and semi-structured information in semantic wikis," in *Proc. 6th Eur. Semantic Web Conf. (ESWC)*, Heraklion, Greece, 2009, pp. 73–87.

[28] N. Jukić, A. Sharma, S. Nestorov, and B. Jukić, "Augmenting data warehouses with big data," *Inf. Syst. Manage.*, vol. 32, no. 3, pp. 200–209, Apr. 2015.

[29] *Apache Hive*. Accessed: Dec. 12, 2019. [Online]. Available: http://hive.apache.org/

[30] *Apache Hbase Project*. Accessed: Dec. 12, 2019. [Online]. Available: https://hbase.apache.org/

[31] C. Q. Jin, W. N. Qian, and A. Y. Zhou, "Analysis and management of streaming data: A survey," *J. Softw.*, vol. 15, no. 8, pp. 1172–1181, 2004.

[32] B. Glavic, K. S. Esmaili, P. M. Fischer, and N. Tatbul, "Efficient stream provenance via operator instrumentation," *ACM Trans. Internet Technol.*, vol. 14, no. 1, pp. 1–26, Aug. 2014.

[33] *Apache Kafka*. Accessed: Dec. 12, 2019. [Online]. Available: http://kafka.apache.org/

[34] C. Liao and A. Squicciarini, "Towards provenance-based anomaly detection in MapReduce," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 647–656.

[35] Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi, "An ontology-based approach for the reconstruction and analysis of digital incidents timelines," *Digit. Invest.*, vol. 15, pp. 83–100, Dec. 2015.

[36] *PROV-DM: The PROV Data Model*. Accessed: Dec. 12, 2019. [Online]. Available: https://www.w3.org/TR/2013/REC-prov-dm-20130430/

[37] *Apache Flume Project*. Accessed: Dec. 12, 2019. [Online]. Available: http://flume.apache.org/

[38] J. Grier, "Detecting data theft using stochastic forensics," *Digit. Invest.*, vol. 8, pp. S71–S77, Aug. 2011.

[39] *Constraints PROV Data Model*. Accessed: Jan. 4, 2020. [Online]. Available: https://www.w3.org/TR/2013/REC-prov-constraints-20130430/

[40] J. Park, D. Nguyen, and R. Sandhu, "A provenance-based access control model," in *Proc. 10th Annu. Int. Conf. Privacy, Secur. Trust*, Jul. 2012, pp. 137–144.

[41] 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 *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jul. 2013, pp. 594–602.

[42] M. B. Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, Menlo Park, CA, USA, vol. 2011, pp. 181–200.

[43] M. Barre, A. Gehani, and V. Yegneswaran, "Mining data provenance to detect advanced persistent threats," in *Proc. 11th Int. Workshop Theory Pract. Provenance (TAPP)*, Philadelphia, PA, USA, vol. 2019, pp. 1–11.

[44] W. Peng, Y. Li, B. Li, and X. Zhu, "An analysis platform of road traffic management system log data based on distributed storage and parallel computing techniques," in *Proc. IEEE Int. Conf. Big Data Cloud Comput. (BDCloud), Social Comput. Netw. (SocialCom), Sustain. Comput. Commun.(SustainCom) (BDCloud-SocialCom-SustainCom)*, Atlanta, GA, USA, Oct. 2016, pp. 585–589.

[45] A. Ichinose, A. Takefusa, H. Nakada, and M. Oguchi, "A study of a video analysis framework using kafka and spark streaming," in *Proc. IEEE Int. Conf. Big Data*, Boston, MA, USA, Dec. 2017, pp. 2396–2401.

[46] B. Debnath, M. Solaimani, M. A. G. Gulzar, N. Arora, C. Lumezanu, J. Xu, B. Zong, H. Zhang, G. Jiang, and L. Khan, "LogLens: A real-time log analysis system," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 1052–1062.

YUANZHAO GAO received the B.S. and M.S. degrees from the Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interest includes big data security.

XINGYUAN CHEN received the Ph.D. degree from the Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2003, where he is currently a Professor. His research interests include networks and information security.

XUEHUI DU received the Ph.D. degree from the Zhengzhou Institute of Science and Technology, Zhengzhou, China, in 2012. She is currently a Professor with the Zhengzhou Institute of Science and Technology. Her research interests include cloud computing and big data security.