Colombo and Ferrari *Cybersecurity* (2019) 2:3
https://doi.org/10.1186/s42400-018-0020-9

Cybersecurity

SURVEY Open Access

Access control technologies for Big Data management systems: literature review and future trends

Pietro Colombo and Elena Ferrari*

Abstract

Data security and privacy issues are magnified by the volume, the variety, and the velocity of Big Data and by the lack, up to now, of a reference data model and related data manipulation languages. In this paper, we focus on one of the key data security services, that is, access control, by highlighting the differences with traditional data management systems and describing a set of requirements that any access control solution for Big Data platforms may fulfill. We then describe the state of the art and discuss open research issues.

Keywords: Big Data, Access control, Privacy, NoSQL data management systems

Introduction

The term Big Data refers to a phenomenon characterized by "5 V". By analysing huge Volumes of data with a high Variety of formats, Big Data analytic platforms allow making predictions with high Velocity, thus, in a timely manner, low Veracity, therefore with low uncertainties, and with a high Value, namely, with an expected significant gain (Jin et al. 2015). As a matter of fact, business strategies are more and more driven by the integrated analysis of huge volumes of heterogeneous data, coming from different sources (e.g., social media, IoT devices).

This phenomenon has been pushed by numerous technological advancements. The most significant include the birth of NoSQL datastores (Cattell 2011), and distributed computational paradigms, like MapReduce (Dean and Ghemawat 2004), which have jointly opened the way to the management and systematic analysis of huge volumes of semi-structured data (e.g., transactions, electronic documents and emails).

Overall, the support provided by Big Data platforms for the storage and analysis of huge and heterogeneous datasets cannot find a counterpart within traditional data management systems. In addition, the advantages of these new systems are not only related to the outstanding flexibility and efficacy of the analysis services, as Big Data platforms outperform traditional systems even with respect to performance and scalability.

However, BigData systems do not show the same level of excellence with data protection features (Colombo and Ferrari 2015b). For instance, while a variety of data protection frameworks have been proposed for traditional systems (see e.g., Agrawal et al. (2002); Byun and Li (2008); Colombo and Ferrari (2014a; 2014b; 2015a); Ferrari (2010)), the majority of Big Data platforms integrate quite basic access control enforcement mechanisms (Colombo and Ferrari 2015b). As a result, the unconstrained access to high volume of data from multiple data sources, the sensitive and private contents of some data resources, and the advanced analysis and prediction capabilities of Big Data analytic platforms, might represent a serious threat. For instance, the analysis capabilities can be exploited to derive correlations between sensitive and personal data. As an example, let us consider the domain of fitness apps which nowadays are more and more deployed on mobile and wearable devices and gym equipment. The joint analysis of movement data, hearth beats, and weight might allow profiling users life style and inferring users inclination to pathologies. As a consequence, although the potential benefits of Big Data analytics are indisputable, the lack of standard data protection tools open these services to potential attackers.

*Correspondence: elena.ferrari@uninsubria.it
DiSTA, University of Insubria, Via Mazzini 5, 21100 Varese, Italy

The definition of proper data protection tools tailored for Big Data platforms is as a very ambitious research challenge. State of the art enforcement techniques proposed for traditional systems cannot be used as they are, or straightforwardly adapted to the Big Data context. This is mainly due to the required support for semi structured and unstructured data (Variety), the quantity of data to be protected (Volume), and the very strict performance requirements (Velocity) affecting these systems. Therefore, the challenge is protecting privacy and confidentiality while not hindering data analytics and information sharing. Additional aspects contribute to raise the complexity of this goal, such as the variety of data models and data analysis and manipulation languages which are used by Big Data platforms. Indeed, different from RDBMSs, Big Data platforms are characterized by various data models (Cattell 2011), the most notable being the key-value, wide column, and document oriented ones.

In this paper, we focus on access control, by first identifying a set of requirements that any access control solution for Big Data platforms should address (cfr. "Requirements" section). Then, we classify and analyze the related literature ("State of the art", "Platform specific approaches", "Platform independent approaches" and "Domain specific Big Data approaches" sections), and discuss key research challenges ("Research issues" section). Finally, we conclude the paper in "Conclusions" section.

This paper is an invited extended version of a paper published in the proceedings of the 23rd ACM Symposium on Access Control Models and Technologies (SACMAT'18)[1]. Current version differs from the original conference paper for a wider and updated analysis of state of the art access control solutions for Big Data systems, which also takes into consideration domain specific platforms, and the related open research challenges.

Requirements

In this section, we provide an overview of the key requirements behind the definition of an access control mechanism for Big Data platforms.

- **Fine-grained access control**. In terms of features the access control mechanism should support, fine grained access control (FGAC) has been widely recognized as one of the fundamental component for an effective protection of personal and sensitive data (e.g., see Agrawal et al. (2002); Rizvi et al. (2004)). Since data processed by Big Data analytics platforms often refer user personal characteristics, it is important that access control rules can be bound to data at the finest granularity levels. However, the related enforcement mechanisms need to be invented from scratch, as those proposed for traditional systems rely on data referring to known schema, while in the context of Big Data, data are heterogeneous and schemaless.
- **Context management**. Another key aspect that should be considered is the support for context based access constraints, as these allow highly customized access control forms. For instance, they can be used to constrain access to specific time periods or geographical locations. In case contexts are used to derive access control decisions, access authorizations are granted when conditions referring to properties of the environment within which an access request has been issued are satisfied.
- **Efficiency of access control**. The characteristics of the Big Data scenario, such as the distributed nature of the considered platforms, the complexity of the queries, and the focus on performance, require access control enforcement strategies that do not compromise the usability of the hosting analytic frameworks. Indeed, based on the considered queries, the number of checks to be executed during access control enforcement can match or be even greater than the number of data records, and, in the Big Data scenario, data sets can include up to hundreds of millions of such records. This requires efficient policy compliance mechanisms. FGAC has been enforced in traditional relational DBMSs according to two main approaches. The first is the view-based one, where users are only allowed to access a view of the target dataset that satisfies the specified access control restrictions, whereas the second one is based on query rewriting. Under such an approach instead of pre-computing the authorized views, the query is modified at run-time by injecting restrictions imposed by the specified access control rules. It is therefore important to determine to what extent these approaches are suitable for the Big Data scenario and how they can be possibly customized or extended.

As it should be clear from the previous discussion, one of the main difficulties in developing an access control solution for Big Data platforms is the lack of a standard model and related manipulation languages to which access control rules and the related enforcement monitor can be bound.

State of the art

In the literature various proposals exist which address the issue of access control for Big Data platforms and satisfy some of the requirements illustrated in "Requirements" section. These proposals can be classified into three main categories:

- **Platform specific approaches**. Access control solutions under this category are designed for one

system only (e.g., MongoDB, Hadoop), and possibly leverage on native access control features of the protected platform. The main advantage of this approach is that the devised access control solution can be optimized for the target system, however, its usability and interoperability are greatly limited.

- **Platform independent approaches**. The approaches falling under this category propose access control solutions which do not target a specific platform only. Platform independent approaches have the advantage of being more general than platform specific solutions, however they cannot compete with them in terms of efficiency. Existing proposals in this category mainly leverage on recent research efforts that aim at defining a unifying query language for NoSQL datastores (e.g., JSONiq (Florescu and Fourny 2013) and SQL++ (Ong et al. 2014)).
- **Domain specific Big Data approaches.** This complementary category includes platform specific and platform independent approaches that target domain specific Big Data systems, designed to fulfill specific requirements related to data management needs of a target scenario. As a matter of fact, a variety of Big Data systems have been designed to handle specific application scenarios, and the literature has shown that in these cases the integration of access control mechanisms has mainly been driven by intrinsic features of these systems. In particular, among the various scenarios that can benefit from Big Data systems, we focus on two of the most relevant ones, namely, data stream analysis and Internet of Things applications, by analyzing related access control enforcement techniques.

In what follows, we analyze the related literature in view of this classification, then we discuss related research challenges.

Platform specific approaches

The great majority of access control frameworks targeting Big Data platforms propose enforcement approaches designed on the basis of platform specific features and which can only be used with the platform for which they have been defined.

In the remainder of this section, we analyze platform specific approaches defined for MapReduce-based analytics platforms[2], and NoSQL datastores, which together cover the majority of existing Big Data systems.

MapReduce systems

MapReduce is a distributed computational paradigm that allows analyzing very large data sets (Dean and Ghemawat 2004). Within MapReduce systems, data resources are partitioned into multiple chunks of data and distributed in a cluster of commodity hardware nodes. Data are analyzed in parallel by means of MapReduce tasks, characterized by users defined Map and Reduce functions. These tasks operate by first extracting and then manipulating flows of key-value pairs, each modeling a portion of the target data resource. The considered computation paradigm allows processing unstructured and semi-structured data resources.

In Ulusoy et al. (2015), a framework denoted GuardMR has been proposed, to enforce fine grained Role-based Access Control (RBAC) (Ferraiolo et al. 2001) within Hadoop[3], a very popular Big Data analytics platform built on top of MapReduce. GuardMR enforces data protection by filtering, and possibly altering, the key-value pairs derived from a target data resource by a MapReduce task, which are then provided as input to the Map function.

Filters are used to generate views of the analyzed resources which are authorized for the subject who requires the execution of the MapReduce task. The views are generated in such a way that any unauthorized content included in the analyzed resource is removed or obfuscated. More precisely, filters specify: i) preconditions to the processing of any key-value pair p extracted from a target resource under analysis, as well as ii) the rationale for deriving from p a new pair p', which models the authorized content of p. The use of filters had previously been considered in Vigiles (Ulusoy et al. 2014), a fine grained access control framework for Hadoop. In Ulusoy et al. (2014), authorization filters are handled by means of per-user assignment lists, and filters are coded in Java by security administrators. In contrast, in GuardMR filters are assigned to subjects on the basis of the covered roles, and a formal specification approach to the definition of filters is proposed, which allows specifying selection and modification criteria at a very high level of abstraction using the Object Constraint Language (OCL)[4] (Warmer and Kleppe 1998; Clark and Warmer 2002). GuardMR relies on automatic tools[5] to generate Java bytecode from OCL-based filter specifications, as well as to integrate the generated bytecode into the bytecode of the MapReduce task to be executed. GuardMR has been used with MapReduce tasks targeting both textual and binary resources (Ulusoy et al. 2015), showing the flexibility of the approach. GuardMR and Vigiles do not require Hadoop source code customization, however, they rely on platform specific features, such as the Hadoop APIs and the Hadoop control flow for regulating the execution of a MapReduce task. A reasonably low enforcement overhead has been observed with both Vigiles and GuardMR. Neither Vigiles nor GuardMR provide support for context aware access control policies.

A recent work targeting access control enforcement within MapReduce systems is described in Gupta et al.

(2017). More precisely, Gupta et al. (2017) introduces the foundations of an access control model, called HeAC, which formalizes the authorization model of Apache Ranger[6] and Apache Sentry[7], as well as the native access control features of Hadoop. Apache Ranger and Apache Sentry represent state of the art technologies for the enforcement of fine grained access control in Hadoop ecosystems. Authorization assignments are specified for operations and objects, possibly on the basis of object tags, namely attributes specifying properties, like sensitivity, content, or expiration date. Moreover, Gupta et al. (2017) introduces the foundation of Object Tagged RBAC, an RBAC model which, while preserving RBAC role based permission assignments, introduces support for object attributes. A prototypical implementation of the model has been defined by introducing role support into Apache Ranger. The proposed enforcement approach is again platform specific as it has been designed on top of Hadoop specific features. No support is given to context related properties, and no performance evaluation is presented.

NoSQL datastores

NoSQL datastores represent highly flexible, scalable, and efficient data management systems for Big Data, based on different data models. Cattell 2011 classifies NoSQL systems into three classes, on the basis of the adopted data model, namely key value, wide column, and document-oriented datastores, each suited to specific application scenarios. Key-value datastores (e.g., Redis[8]) can be seen as big hash tables with persistent storage services. Data are modeled by means of key-value pairs, where values of primitive or complex type are directly addressed by means of a key. Key value datastores are suited to application scenarios where efficient look-up operations are required. For instance, they are used to manage web session information and users profile data. Wide column stores (e.g., Cassandra[9]) model data as records with variable structures, which are then grouped into tables with flexible schema. Wide column stores are a good fit for the data management requirements of blogging platforms and content management systems. Document-oriented datastores (e.g., MongoDB [10]) model data as hierarchical records, denoted documents, whose fields either specify a primitive value, or are in turn records composed of multiple fields. Documents are partitioned into collections, which in turn are grouped in a database. Typical applications of document oriented datastores include event logging systems and content management systems.

Fine grained access control within NoSQL datastore management systems is still in the very early stage, and only few access control frameworks have been proposed so far for wide column and document oriented datastores.

K-VAC (Kulkarni 2013) is among the earliest fine grained access control frameworks targeting wide-column NoSQL datastores which have been proposed in the literature. K-VAC supports the enforcement of content-based, and context-based access control policies possibly specified at different levels of the data model hierarchy (e.g., for a column or for a row). Two prototypical versions of K-VAC have been released. One has been specifically designed as an internal module of Cassandra, a popular wide-column datastore whose source code has been modified to host K-VAC's enforcement monitor. In contrast, the latter version has been released as an external library, with the aim to enforce access control on multiple datastores. However, the use of the proposed library still requires ad-hoc implementation of binding criteria, which so far have been only defined for Cassandra and HBase[11]. Overall the integration of K-VAC requires deep customizations of the hosting platform. Empirical performance evaluations show the efficiency of both the proposed prototypes, with a lower overhead measured with the customized version of Cassandra.

Another work targeting Cassandra has been proposed in Shalabi and Gudes (2017), where an approach to the cryptographic enforcement of RBAC policies has been defined. Predicate (Katz et al. 2013) and second level encryption (Nabeel and Bertino 2014) are used for the definition of an efficient scheme for RBAC enforcement which operates within Cassandra distributed architecture. The proposed approach is an example of platform specific solution designed on top of specific features, such as the distributed architecture of Cassandra. Also in this case no support is given for context-aware policies, and, unfortunately, the enforcement overhead is not discussed.

As far as document-oriented datastores, efficient solutions to the integration of fine-grained purpose-based access control into MongoDB have been proposed in Colombo and Ferrari (2016) and (2017a). In Colombo and Ferrari (2017a) the RBAC model natively integrated in MongoDB has been enhanced with the support for the specification and enforcement of purpose-based policies (Byun and Li 2008) regulating the access up to document level. The proposed approach refines the granularity level at which the native MongoDB RBAC model operates. An enforcement monitor, called Mem (MongoDB enforcement monitor), has been designed, which monitors and possibly manipulates the flow of messages exchanged by MongoDB clients and the MongoDB server, thus acting like a proxy. Once Mem intercepts a message m issued by a MongoDB client on behalf of a subject s, it forwards m to the server, or it temporary blocks m, and issues additional messages finalized at profiling s. If m models a query q, Mem rewrites m as m' in such a way that m' encodes a query q' that only accesses those documents accessed by q which result authorized by the applicable access control policies. Mem's proxy based architecture allows the straightforward integration of the enforcement

monitor into existing MongoDB deployments with basic configuration tasks. Experimental evaluations show the efficiency of the proposed approach, however also in this case no support is given for context-aware policies.

In Colombo and Ferrari (2016), the framework presented in Colombo and Ferrari (2017a) has been significantly extended, introducing the support for access control policies regulating the access up to field level, and providing support to specification and enforcement of content and context based policies. The proposed enforcement monitor, denoted ConfinedMem, applies the same logic as Mem, but it operates according to a two-step process, which consists of: 1) the derivation of the authorized views of all documents to be accessed by a submitted query *q* included in a message *m* requiring the access to data resources, 2) the rewriting of *m* as *m'* in such a way that *m'* specifies a query *q'* which can only access the authorized views of the documents to be accessed by *q*. Different implementation techniques have been considered for queries specifying different operations (e.g., selection and aggregations) with the aim to minimize the overhead. Experimental evaluations show that, overall, the enforcement overhead which has been observed with access control policies specified at field level is significantly higher than the one measured for document level policies.

Platform independent approaches

The great majority of the research contributions in the field of access control for Big Data analytics platforms propose a platform specific solution.

The lack of a reference standard query language and data model has caused the birth of a variety of proprietary solutions. As a matter of fact, numerous NoSQL datastores exist, most of which operate with a platform specific query language (e.g., the query language of MongoDB can only be used with that platform), and adopt a different data model. Even different datastores that nominally refer to the same data model can use different data organization and terminology. For instance, both MongoDB and CouchDB[12] use the document oriented data model, however the concept of collection is not supported by CouchDB, whereas collections are basic data organization features of MongoDB. The great heterogeneity of the scenario has significantly raised the complexity of devising enforcement solutions that can work with multiple platforms. Overall, the definition of a general access control enforcement approach represents a very ambitious task.

In the recent years, academia and industry started collaborating to the definition of unifying query languages for NoSQL datastores. To the best of our knowledge, JSONiq (Florescu and Fourny 2013) and SQL++ (Ong et al. 2014) represent the most relevant results that have been so far achieved towards the fulfillment of this goal. JSONiq is an Xquery (Chamberlin 2003) based language that has been defined with the aim to analyze data handled by NoSQL datastores adopting a JSON-based data model. Unfortunately, at present JSONiq is only supported by Zorba[13], and Sparksoniq[14], which allow processing data serialized in JSON format, and by a platform denoted 28msec[15], which supports the execution of JSONiq queries on MongoDB databases.

SQL++ (Ong et al. 2014) is a recent proposal of unifying query language that allows analysing semi-structured data handled by NoSQL datastores as well as structured data of traditional DBMSs. SQL++ has been recently adopted by Couchbase[16] and AsterixDB[17](Alsubaiee et al. 2014), whereas Apache Drill[18], is in the process of aligning with SQL++. The diffusion of this language is thus growing, and the adopted SQL based syntax and the backward compatibility with relational DBMSs promise to further increase its popularity and diffusion.

In Colombo and Ferrari (2017b) an SQL++ based Attribute-based Access Control (ABAC) (Hu et al. 2013; 2015) framework for NoSQL datastores has been proposed. The choice to base the framework on SQL++ allows protecting any NoSQL datastore which provides support to this language. Therefore, the proposal distinguishes from all other work introduced in "Platform specific approaches" section for higher generality and applicability, which may even grow with a future potential wider diffusion of SQL++. The framework operates at a very fine grained level, in that it allows regulating the access up to single data fields. The supported granularity is thus equivalent to cell level within relational DBMSs. Enforcement is based on query rewriting and operates with heterogeneous data with no assumption on data schema, thus overcoming state of the art query rewriting techniques proposed for RDBMSs (Rizvi et al. 2004; LeFevre et al. 2004).

Query rewriting techniques finalized at enforcing cell-level access control within traditional DBMSs operate by projecting or nullifying the value of each cell to be accessed by a query *q* on the basis of the compliance of the access performed by *q* with the applicable access control policies (LeFevre et al. 2004). More precisely, a query *q* submitted for execution is rewritten in such a way to: i) include a subquery *s* for each table *t* accessed by *q*, which, cell by cell, generates an authorized view of *t*, and ii) perform the same analysis tasks as *q* on the result set of *s*. The subquery *s* specifies projection criteria conditioned by the compliance of the accesses operated by *q* with the cell level access control policies that have been specified for *t*'s cells. A similar approach can only be used if the scheme of any accessed table is a priori known, as the projection criteria of the subqueries need to refer to table columns. The schemaless and highly heterogeneous nature of the data

within Big Data platforms does not allow to use similar techniques.

In Colombo and Ferrari (2017a) this issue has been handled by means of SQL++ operators that allow achieving the projection without knowing in advance the accessed fields. The approach operates by visiting, field by field, the data unit[19] *du* of an analyzed resource, and adding a visited field *f* to the authorized view *du'* of *du* only if the access to *f* complies with the ABAC policies specified for *f*. The proposed approach allows deriving in-memory authorized views of the data resources to be analyzed, and executing the analysis tasks of the original queries on such derived views. The ABAC framework proposed in Colombo and Ferrari (2017a) supports the specification and enforcement of context-aware access control policies. Empirical performance assessments show an enforcement overhead that varies with the characteristics of the specified policies and the number of fields of the analyzed documents. The overhead is high when field level policies cover high percentage of data units fields.

Another language-based ABAC approach has been proposed in Longstaff and Noble (2016), with the goal to be usable with traditional data management systems, Mapreduce systems, as well as NoSQL datastores. The work proposes a query rewriting approach that targets user transactions specified with an SQL-like language. Unfortunately, a detailed description of the adopted query language and data model is missing, which makes unclear how the approach could be used with different platforms, and how the heterogeneity of schemaless data can be handled by means of an SQL-like language.

A summary of the access control frameworks discussed so far along with the supported access control requirements (cfr. "Requirements" section) is shown in Table 1.

Domain specific Big Data approaches

In this section, we focus on the state of the art approaches to the integration of access control into Big Data systems designed for specific application domains. In particular, we first analyze approaches that target Big Data platforms supporting data stream analytics, and then we focus on those for Internet of Things ecosystems.

Big Data streaming analytics

In recent years, the number of Big Data platforms that provide support to data stream management is growing. Apache Spark[20] is probably the most popular open source framework which supports the analysis of continuous streams of data. Apache Storm[21] is another open source distributed real-time computation system which can also be used for real-time analytics and continuous computation. In addition, several commercial solutions exist, such as, for instance, Amazon Kinesis[22], which is a service for real-time processing of streaming data on the cloud, and IBM Streaming analytics[23], a platform supporting risk analysis and decision making in real-time. Due to the growing emphasis to real-time analysis of data flows, access control enforcement mechanisms targeting continuous flows of data are strongly required. A few results have been presented in the past years in the field of Data Stream Management Systems (DSMSs) (e.g., Nehme et al. (2010), Carminati et al. (2010), and Puthal et al. (2015)).

In Nehme et al. (2010), a framework, called FENCE, has been proposed, which supports continuous access control enforcement. Data and query security restrictions are modeled as meta-data, denoted security punctuations, which are embedded into the data streams. Different enforcement mechanisms have been proposed, which operate by analyzing security punctuations, such as special physical operators which are integrated within query execution plans with the aim to filter the tuples which can be analyzed, and rewriting mechanisms targeting continuous queries.

The framework in Carminati et al. (2010) assumes that data analysis within DSMSs is achieved by continuous queries, and enforces access control by means of query

Table 1 Summary of the surveyed platform specific and platform independent access control frameworks

AC framework	Target platform	AC model	Max granularity	Context support	Efficiency
GuardMR (Ulusoy et al. 2015)	Hadoop	RBAC	K,V pair	No	Medium/High
Vigiles (Ulusoy et al. 2014)	Hadoop	DAC	K,V pair	No	Medium/High
HeAC (Gupta et al. 2017)	Hadoop	RBAC	K,V pair	No	Not available
K-VAC (Kulkarni 2013)	Cassandra/ HBase	DAC	Cell	Yes	High
Shalabi and Gudes (2017)	Cassandra	RBAC	Cell	No	Not available
Mem (Colombo and Ferrari 2017a)	MongoDB	RBAC	Document	No	High
ConfinedMem (Colombo and Ferrari 2016)	MongoDB	DAC	Field	Yes	Medium/Low
Colombo and Ferrari (2017b)	All those supporting SQL++	ABAC	Cell/Field	Yes	Medium
Longstaff and Noble (2016)	Not clear	ABAC	Cell/Field	Yes	Not available

rewriting, where rewritten queries are defined by composition of secure query operators. In contrast, Puthal et al. (2015) presents a crypto-based solution to verify authenticity and integrity of data streams.

Complex event processing (CEP) systems (Cugola and Margara 2015) represent the evolution of DSMSs (Cugola and Margara 2012), and are nowadays used for many different applications, such as Internet of Things applications and Smart Cyber-physical systems (Dayarathna and Perera 2018).

CEPs support the processing of heterogeneous streams from multiple sources, as well as advanced forms of reasoning over such data streams. On the basis of the experience with DSMSs in Carminati et al. (2010), a novel access control model for CEP platforms has been proposed in Carminati et al. (2016). The model assumes an application scenario where users generating continuous flows of data, specify how their data can be processed and what cannot be inferred from the data. The compliance of the access performed by a query with the specified user preferences is checked by verifying that each operator in the submitted query complies with the user preferences specified for the accessed attributes of the analyzed data streams. In Migliavacca et al. (2010), a system, called DEFCON, has been presented to enforce decentralised event flow control. The system, which has been designed targeting the financial trading scenario, applies information flow control principles and leverages on security labels assigned to event messages. Event flow control is achieved through a lightweight approach that makes use of application-level virtualisation to separate processing units.

Internet of Things

Internet of Things (IoT) ecosystems are representative cases of Big Data applications. IoT applications are rapidly getting popularity in a variety of domains for the indisputable improvements of people life style they bring. Nowadays a growing number of users cannot do without wereable devices that track their movements, sport activities and health conditions, and a variety of devices and apps exist for this purpose. IoT applications are used to control the safety of the environments where people live, as well as to improve their life style. As a matter of fact, the diffusion of home automation services and smart devices like smart locks, smart meters, and smart lights is growing.

Due to the personal and sensitive nature of the handled information, security and privacy of these systems have become a major concern. Therefore, in the recent years, several research efforts have been devoted to security and privacy of IoT applications, and a variety of access control models have been proposed (see, for instance, Ouaddah et al. (2017) for a compendium).

For instance, Gusmeroli et al. (2013) and Hernández-Ramos et al. (2013) propose the use of the Capability based access control model (CapBAC) within IoT ecosystems.

CapBAC distinguishes from other models in the literature as it allows externalizing and distributing the management of access authorizations. However, it does not take context awareness into account, and for this reason it has been criticized (Ouaddah et al. 2015).

RBAC (Ferraiolo et al. 2001) and ABAC (Hu et al. 2013; 2015) have also been proposed to regulate the access within IoT ecosystems.

For instance, in Zhang and Tian (2010), with the aim to fit IoT dinamicity, an extended version of RBAC supporting contextual constraints has been introduced. However, the resulting enhanced model has been criticized (e.g., see Rajpoot et al. (2015)) as it is affected by shortcomings, like role explosion, which also characterize RBAC. A few approaches have been based on the ABAC model. For instance, Kaiwen and Lihua (2014) propose an ABAC model that extends RBAC with the dynamic assignment of roles to users. However, the proposed model only partially exploits ABAC features, as it only supports subject attributes. Another ABAC model operating with a predefined set of attributes has been proposed in Hemdi and Deters (2016). The proposed enforcement monitor has been designed for IoT ecosystems that use CoAP[24] as communication protocol. Unfortunately, the focus of Hemdi and Deters (2016) is on implementation aspects, and neither the enforcement mechanism nor the supported access control policies are formally specified.

In Marra et al. (2017), La Marra et al. (2018) and 2017 a framework is proposed, supporting the enforcement of Usage Control (UCON) (Zhang et al. 2005) within IoT ecosystems. The approach is illustrated discussing the policy enforcement mechanism within a Smart Home environment. However, the generality of the proposed mechanism is limited by constraining assumptions, such as the use of ad-hoc defined brokers.

A general enforcement mechanism has been proposed in Colombo and Ferrari (2018), which allows enforcing policies of different access control models within MQTT-based IoT ecosystems. The proposed framework provides a monitor that enforces access control by regulating the flow of the exchanged MQTT control packets. The framework is illustrated using ABAC, but other models are also supported.

A recent research line targets the study of access control enforcement for cloud-enabled IoT (see e.g., Alshehri and Sandhu (2016; 2017); Bhatt et al. (2017; 2018); Ahmad et al. (2018)). Alshehri and Sandhu propose an access control oriented (ACO) architecture (Alshehri and Sandhu 2016; 2017), which supports the definition of access control models for cloud-based IoT services. ACO has been

used to define enforcement mechanisms tailored for specific IoT platforms (Bhatt et al. 2017), and applications (Bhatt et al. 2018).

Access control enforcement for cloud-enabled IoT systems has also been investigated by Ahmad et al. (2018), who, starting from a case study related to a smart home environment, have identified a set of key requirements for the enforcement of access control within IoT ecosystems. The authors have also proposed an approach to handle access control as a service, outsourcing policy management to a trusted third party, while relying on the native mechanisms of state of the art IoT platforms for policy enforcement. The feasibility of the approach has been assessed wrt the satisfiability of the identified requirements.

Finally, some proposals target of intelligent transportation systems. Recent research efforts in this field have been devoted to enable advanced communication forms among vehicles, road infrastructures, drivers, as well as intra-vehicle devices. The envisaged services rely on a variety of technologies, which range from dedicated hardware and software components, to the enabling communication infrastructures, possibly cloud or fog based. In this complex scenario vehicular security represents a major concern, and the US Department of Transportation has already outlined the strategic goals of an Intelligent Transportation System Program (Barbaresso and et al. 2014). Initial research results in this field have been described in Gupta and Sandhu (2018), where an extended version of the ACO architecture presented in Alshehri and Sandhu (2016) is discussed, called E-ACO. The paper also discusses enforcement mechanisms tailored for various E-ACO layers, however the topic remains an open research field, with room for investigations in manifold directions (see Gupta and Sandhu (2018)).

Research issues

In what follows, we discuss some open research issues in the field of Big Data access control.

Unifying access control models and mechanisms

State of the art review done in "Platform specific approaches", and "Platform independent approaches" sections has highlighted that, although research in the area of access control for Big Data platforms is progressing, no solution has been proposed so far for a unifying access control framework which can combine generality and efficiency of access control. The heterogeneous schemaless nature of the managed data significantly complicates the definition of this framework, and so far this has lead mainly to ad-hoc platform specific solutions (see "Platform specific approaches" section). In contrast, language centric approaches still suffer of limited applicability (see "Platform independent approaches" section). For instance, although the popularity of the SQL++ (Ong et al. 2014) initiative is growing, the support provided to this language is still limited to a small number of platforms.

One key element that may be instrumental to fill this void is the definition of a unifying data model capable of representing data resources of the different data models currently adopted by Big Data platforms. The ability to represent data resources is a fundamental requirement for binding access control policies to the protected data, as well as for the specification of policies regulating the access on the basis of the protected objects' attributes. Indeed, in the literature on access control, multiple models allow enforcing content-based access constraints (e.g., Kulkarni (2013); Colombo and Ferrari (2016)), as well as access control rules that refer to various security meta-data related to the protected data resources (e.g., Colombo and Ferrari (2015a)).

The key-value, wide column, and document-oriented models adopt different data modeling criteria, however, in all these models data are hierarchically organized as tree structures, where nodes at different height of the tree represent resources at different granularity levels of the related data model (e.g., database, table, row, and cell). Data models differ among them for the height of the tree with which data resources can be represented. This may range from 2, within key-value datastores (since all key-value pairs – leaf nodes, belong to a key-space – root node.) to a height of variable length n ($n > 2$) for document-oriented datastores, where a database (root node), groups a variable number of collections (level 2 nodes), which in turn include a variable number of documents (level 3 nodes), each composed of a variable number of fields, which in turn are possibly hierarchically organized into a tree structure (level 4 to n). A data resource of a data model corresponds to a node n of the tree representing all the resources handled by a platform, and it can be accessed traversing the path from the root of the tree to n. Therefore, we believe that a unifying representation of data resources of multiple data models should take into account the identification of proper modeling strategies for the nodes of the above mentioned resource tree. In particular, nodes should be specified in such a way to keep track of: i) any structural property related to the modeled resource, ii) hierarchical relations with other nodes (e.g., a parent of relationship), iii) possible meta-data, and iv) access control policies specified for the modeled resource. The considered policies may refer to different access control models, specifying context aware access control rules as well as content-based constraints.

Going one step further, the specified unifying model could also be used for enforcement purposes. For instance, enforcement mechanisms can be achieved by means of bidirectional mappings between resources

represented with a platform specific native data model and the unifying data model. Overall, the analysis of related work has revealed that fine grained access control with schemaless data is usually enforced executing the submitted analysis tasks on authorized views of the accessed resources (e.g., see Colombo and Ferrari (2017b)). Therefore, a platform independent strategy to handle fine grained access control enforcement may consist of a pipeline of operations supported by any platform, which, by means of the unifying data model, handle the generation of authorized views. The generated view can then be analyzed by the originally submitted query without additional platform specific rewriting activities. The above mentioned pipeline is illustrated in Fig. 1. For each accessed resource *rs*, represented as a tree characterized by different nodes n_i, the process: i) derives a unifying model-based representation *urs* of *rs*, ii) derives the authorized view *urs'* of *urs*, where the unauthorized contents have been removed, and, finally, iii) maps back the authorized view *urs'* to the native data model, so that the generated view *rs'* can be analyzed by the originally submitted analysis task. In order to support such approach within multiple platforms, the above mentioned mapping and view generation mechanisms should be defined in such a way that any platform, independently from the supported query language and data model, could handle the execution of this process. To the best of our knowledge, the majority of today Big Data platforms provide support for MapReduce computational paradigm, independently from the adopted data model and query language. Therefore, a promising approach could be that of specifying mapping and view generation mechanisms by means of MapReduce operations.

The enforcement overhead of the above discussed technique is expected to depend on the platform hosting the data to be protected, as different behaviors are expected to be observed. For instance, Apache Spark[25], integrates a highly efficient computation engines, which promises to be significantly faster than Hadoop[3] (up to 100 times faster[26].) The overhead is expected to be reasonably contained in all those platforms supporting in-memory MapReduce computations, as well as data streams.

Policy analysis tools

The availability of a unifying data model on which access control policies can be specified would also allow to support policy analysis and reasoning at an abstract layer independent from any specific platform. As a matter of fact, the variety of data models, access control models, and related configuration options, such as policy propagation and conflict resolution criteria, adopted by Big Data platforms, can make really hard for security administrators to understand the effect of a set of access control policies on the data resources which are managed by their systems, as well as assessing the quality of the specified policies. Most of the research efforts in this field have been devoted to correctness verification, detection of inconsistencies and redundancies, as well as reasoning on policy sets completeness. A variety of approaches have been adopted to achieve the analysis, which range from the use of formal methods, to machine learning and data mining techniques. For instance, Datalog-based approaches have been proposed in Pasarella and Lobo (2017) and Tsankov et al. (2014), which respectively target Relationship-based Access Control (ReBAC) policies, and decentralized composite access control systems. Approaches based on Answer Set Programming (ASP), such as the ones proposed in Ahn et al. (2010) and Kencana Ramli et al. (2013), allow the derivation of ASP programs from XACML[27] policies, and the analysis on the specified policies by means of ASP solvers. Model checking approaches have been proposed in Guelev et al. (2004) and Zhang et al. (2005), whereas SAT solvers and Multi-Terminal Binary Decision Diagrams based techniques in

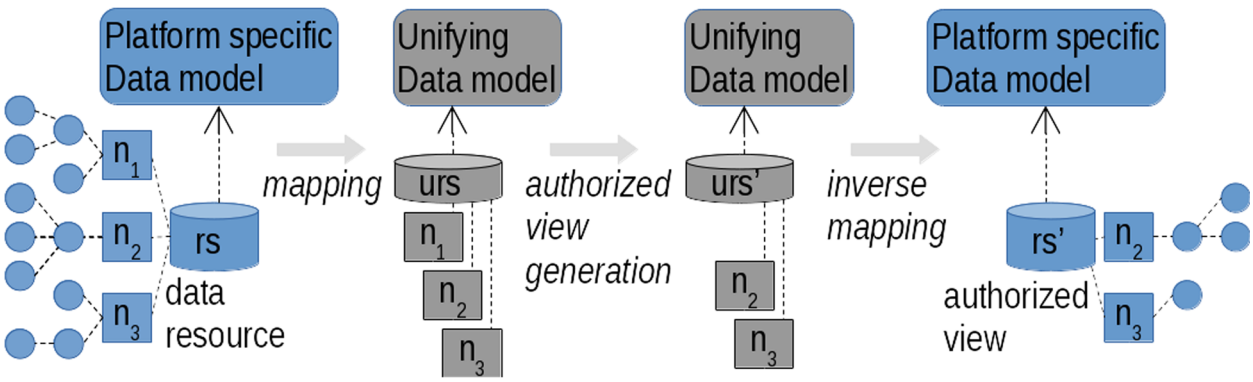

Fig. 1 The pipeline of operations at the basis of an enforcement mechanism leveraging on the unifying data model

Lin et al. (2010) as a basis for reasoning on the permissions granted by access control policies. Graph-based analysis approaches for category based access control policies have been proposed in Alves and Fernández (2015), with the aim to ease verification tasks of security administrators. Finally, data mining techniques have been primarily used for the detecting policy anomalies (e.g., Hu et al. (2013)).

In Bertino et al. (2017) provenance techniques have been proposed to check the quality of the specified access control policies for a scenario where collaborations are carried out by autonomous cognitive devices. However, to the best of our knowledge, so far no proposal has yet targeted Big Data platforms. The model centric approach previously discussed may be exploited as a basis for the definition of such policy analysis framework. For instance, it may be used to generate views of the protected resources that show the authorized and unauthorized contents when different policies and configuration options are used, as well as to quantify policy coverage for a requesting subject with respect to an execution context.

The definition of a policy reasoning tool is also instrumental to fulfill the new EU General Data Protection Regulation, (GDPR)[28] which is intended to strengthen data protection for all individuals within the European Union. GDPR applies regardless of where a company is located, provided that the company manages data of EU residents. GDPR introduces a set of very important principles for Big Data management, such as privacy by-design and by-default. The new regulation also emphasizes accountability for data controllers to demonstrate compliance to GDPR, whereas article 35 requires controllers to carry out Data Protection Impact Assessments in case of potentially high-risk processing activities. All such principles require tools to clearly assess the effect of access control policies on the managed data.

Finally, a policy analysis framework is also required for community centered collaborative systems, such as online social networks and collaborative editing platforms, which may be seen as federated applications that handle Big Data. Recent surveys pointed out that these systems typically provide rudimentary forms of access control (Paci et al. 2018). A key requirement for access control models tailored for collaborative systems is to allow users to understand collaborative decisions, as well as to inspect users access preferences, and to evaluate their effects (Paci et al. 2018). Paci et al. (2018) claim that, although a few work exist which explain the effect of access decisions (Hu et al. 2013), and the reasons for which certain decisions have been taken (den Hartog and Zannone 2016), the above mentioned requirements are still largely understudied. Therefore, the definition of a reasoning framework capable of operating within such federated environments with multiparty access control models appears as a research challenge of paramount importance.

Overall, so far research on policy analysis has primarily focused on different properties of policy sets abstracting from the effects of policy enforcement on the protected resources. In contrast, we believe that frameworks capable of evaluating the effect of policy sets on resource accessibility within different Big Data platforms are required, which may provide support to multiple access control models and configuration options.

Issues related to domain specific Big Data systems

Let us now consider open challenges related to access control enforcement within domain specific Big Data systems. A selection of approaches targeting the enforcement of access control policies within traditional DSMSs and CEP platforms have been shortly presented in "Big Data streaming analytics" section. A possible strategy to integrate similar enforcement approaches into Big Data analytics platforms may consist in designing the mechanism on top of one of the existing streaming framework. However, similar to the platform specific approaches presented in "Platform specific approaches" section, such a solution would suffer from a limited applicability. Moreover, existing solutions (e.g., Nehme et al. (2010)) operate at tuple level and scheme level (e.g., Carminati et al. (2016)), whereas cell/field level granularity may be necessary in the Big Data scenario (see "Platform specific approaches" and "Platform independent approaches" sections), requiring a data filtering approach that operates at a finer granularity level. The development of an enforcement mechanisms based on language centric approaches seems still impracticable, as no standard continuous query language exists. In contrast, since some of these platforms can implement MapReduce tasks (e.g., Apache Spark, Apache Storm), a model centric approach may be a possible strategy, however, thorough investigations are required to support this intuition.

For what IoT ecosystems are concerned, the initial efforts shortly summarized in "Internet of Things" section have mainly produced models adopting centralized enforcement mechanisms (e.g., see Colombo and Ferrari (2018)). However, multiple IoT ecosystems may be connected to each other exchanging data, and federated systems where multiple IoT applications cooperate cannot be handled with centralized enforcement mechanisms. Multiparty access control solutions for IoT ecosystems are thus needed, and they must be suited to operate at Big Data scale. To the best of our knowledge, the definition of such access control frameworks still represent a big open research challenge.

Conclusions

Security services for Big Data represent a key feature instrumental to foster trust on how data are managed and analyzed by Big Data platforms. This paper has focused

on one of the key security service, that is, access control, by discussing the requirements that an access control solution for Big Data platforms should address, also with reference to specific key application scenarios (i.e., IoT and data streams). Moreover, the paper has provided a review of the state of the art in view of the devised requirements, and it has also discussed future research challenges in the area.

Endnotes

[1] Details are omitted due to the blind submission requirements.

[2] MapReduce-based analytics platforms are hereafter denoted MapReduce systems for the sake of brevity.

[3] http://hadoop.apache.org/

[4] https://www.omg.org/spec/OCL

[5] Dresden OCL Toolkit, http://st.inf.tu-dresden.de/oclportal

[6] https://ranger.apache.org/

[7] https://sentry.apache.org/

[8] https://redis.io/

[9] http://cassandra.apache.org/

[10] https://www.mongodb.com

[11] HBase is a popular wide-column store, https://hbase.apache.org/

[12] http://couchdb.apache.org/

[13] http://www.zorba.io

[14] http://sparksoniq.org/

[15] https://www.28msec.com/

[16] https://www.couchbase.com/

[17] https://asterixdb.apache.org/

[18] https://drill.apache.org/

[19] SQL++ can be used with datastores adopting different data models, thus, the term data unit is used to denote a table row, or a document.

[20] https://spark.apache.org/

[21] http://storm.apache.org/

[22] https://aws.amazon.com/kinesis/

[23] https://www.ibm.com/cloud/streaming-analytics

[24] http://coap.technology/

[25] https://spark.apache.org/

[26] https://www.datamation.com/data-center/hadoop-vs.-spark-the-new-age-of-big-data.html

[27] eXtensible Access Control Markup Language (XACML) Version 3.0 http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[28] https://www.eugdpr.org/

Acknowledgements
Not applicable.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Authors' contributions
The authors declare that they have equally contributed to the preparation of the article, all authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 27 August 2018 Accepted: 19 December 2018
Published online: 24 January 2019

References

Agrawal R, Kiernan J, Srikant R, Xu Y (2002) Hippocratic Databases. In: Proceedings of the 28th International Conference on Very Large Data Bases, VLDB '02. pp 143–154

Ahmad T, Morelli U, Ranise S, Zannone N (2018) A Lazy Approach to Access Control As a Service (ACaaS) for IoT: An AWS Case Study. In: Proceedings of the 23Nd ACM on Symposium on Access Control Models and Technologies. SACMAT '18. ACM, New York. pp 235–246

Ahn G, Hu H, Lee J, Meng Y (2010) Representing and Reasoning about Web Access Control Policies. In: 34th Annual Computer Software and Applications Conference. IEEE, Seoul. pp 137–146. https://doi.org/10.1109/COMPSAC.2010.20

Alshehri A, Sandhu R (2016) Access Control Models for Cloud-Enabled Internet of Things: A Proposed Architecture and Research Agenda. In: 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC). pp 530–538

Alshehri A., Sandhu R. (2017) Access Control Models for Virtual Object Communication in Cloud-Enabled IoT. In: 2017 IEEE International Conference on Information Reuse and Integration. pp 16–25

Alsubaiee S, Altowim Y, Altwaijry H, Behm A, Borkar V, Bu Y, Carey M, Cetindil I, Cheelangi M, Faraaz K, et al. (2014) AsterixDB: A scalable, open source BDMS. Proc VLDB Endowment 7(14):1905–1916

Alves S, Fernández M (2015) A Framework for the Analysis of Access Control Policies with Emergency Management. Electron Notes Theor Comput Sci 312:89–105. Ninth Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2014)

Barbaresso J, et al. (2014) USDOT's Intelligent Transportation Systems ITS. In: Strategic Plan 2015-2019

Bertino E, Jabal AA, Calo SB, Makaya C, Touma M, Verma DC, Williams C (2017) Provenance-Based Analytics Services for Access Control Policies. In: 2017 IEEE World Congress on Services, SERVICES 2017, Honolulu, HI, USA, June 25-30, 2017. pp 94–101

Bhatt S, Patwa F, Sandhu R (2017) Access Control Model for AWS Internet of Things. In: Yan Z, Molva R, Mazurczyk W, Kantola R (eds). Network and System Security. Springer, Cham. pp 721–736

Bhatt S, Patwa F, Sandhu R (2018) An Access Control Framework for Cloud-Enabled Wearable Internet of Things. In: 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC). pp 328–338

Byun JW, Li N (2008) Purpose based access control for privacy protection in relational database systems. VLDB J 17(4):603–619

Carminati B, Colombo P, Ferrari E, Sagirlar G (2016) Enhancing User Control on Personal Data Usage in Internet of Things Ecosystems. In: 2016 IEEE International Conference on Services Computing (SCC). pp 291–298

Carminati B, Ferrari E, Cao J, Tan KL (2010) A Framework to Enforce Access Control over Data Streams. ACM Trans Inf Syst Secur 13(3):28–12831

Cattell R (2011) Scalable SQL and NoSQL Data Stores. SIGMOD Rec 39(4):12–27

Chamberlin D (2003) XQuery: A Query Language for XML. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. SIGMOD '03. ACM, New York (USA). pp 682–682

Clark T, Warmer J (2002) Object Modeling with the OCL. The Rationale behind the Object Constraint Language. LNCS, Volume 2263. Springer, Berlin

Colombo P, Ferrari E (2014a) Enforcement of Purpose Based Access Control within Relational Database Management Systems. IEEE Trans Knowl Data Eng (TKDE) 26(11):2703–2716

Colombo P, Ferrari E (2014b) Enforcing Obligations within Relational Database Management Systems. IEEE Tran Dependable Sec Comput (TDSC) 11(4):318–331

Colombo P, Ferrari E (2015a) Efficient Enforcement of Action aware Purpose Based Access Control within Relational Database Management Systems. IEEE Trans Knowl Data Eng (TKDE) 27(8):2134–2147

Colombo P, Ferrari E (2015b) Privacy Aware Access Control for Big Data: A Research Roadmap. Big Data Res 2(4):145–154

Colombo P, Ferrari E (2016) Towards Virtual Private NoSQL datastores. In: 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016. pp 193–204

Colombo P, Ferrari E (2017a) Enhancing MongoDB with purpose-based access control. IEEE Trans Dependable Sec Comput 14(6):591–604

Colombo P, Ferrari E (2017b) Towards a unifying attribute based access control approach for nosql datastores. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017. pp 709–720

Colombo P, Ferrari E (2018) Access Control Enforcement Within MQTT-based Internet of Things Ecosystems. In: 23Nd ACM on Symposium on Access Control Models and Technologies. SACMAT '18. ACM, New York (USA). pp 223–234

Cugola G, Margara A (2012) Processing Flows of Information: From Data Stream to Complex Event Processing. ACM Comput Surv 44(3):1–62

Cugola G, Margara A (2015) The Complex Event Processing Paradigm(Colace F, De Santo M, Moscato V, Picariello A, Schreiber FA, Tanca L, eds.). Springer, Cham

Dayarathna M, Perera S (2018) Recent Advancements in Event Processing. ACM Comput Surv 51(2):33–13336

Dean J, Ghemawat S (2004) MapReduce: Simplified Data Processing on Large Clusters. In: Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6. OSDI'04. USENIX Association, Berkeley. pp 10–10

den Hartog J, Zannone N (2016) A Policy Framework for Data Fusion and Derived Data Control. In: Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control. ABAC '16. ACM, New York. pp 47–57

Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R (2001) Proposed NIST Standard for Role-based Access Control. ACM Trans Inf Syst Secur 4(3):224–274

Ferrari E (2010) Access Control in Data Management Systems. Synthesis Lectures on Data Management. Morgan & Claypool Publishers. ISBN: 1608453758 9781608453757

Florescu D, Fourny G (2013) JSONiq: The History of a Query Language. IEEE Internet Comput 17(5):86–90

Guelev DP, Ryan M, Schobbens PY (2004) Model-Checking Access Control Policies. In: Zhang K, Zheng Y (eds). Information Security. Springer, Berlin, Heidelberg. pp 219–230

Gupta M, Patwa F, Sandhu R (2017) Object-tagged RBAC model for the hadoop ecosystem. In: Livraga G, Zhu S (eds). Data and Applications Security and Privacy XXXI. Springer, Cham. pp 63–81

Gupta M, Sandhu RS (2018) Authorization framework for secure cloud assisted connected cars and vehicular internet of things. In: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018. pp 193–204

Gusmeroli S, Piccione S, Rotondi D (2013) A capability-based security approach to manage access control in the Internet of Things. Math Comput Model 58(5):1189–1205. The Measurement of Undesirable Outputs: Models Development and Empirical Analyses and Advances in mobile, ubiquitous and cognitive computing

Hemdi M, Deters R (2016) Using REST based protocol to enable ABAC within IoT systems. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp 1–7

Hernández-Ramos JL, Jara AJ, Marin L, Skarmeta AF (2013) Distributed capability-based access control for the internet of things. J Internet Serv Inf Secur (JISIS) 3(3/4):1–16

Hu H, Ahn G, Kulkarni K (2013) Discovery and resolution of anomalies in web access control policies. IEEE Trans Dependable Sec Comput 10(6):341–354

Hu H, Ahn GJ, Jorgensen J (2013) Multiparty Access Control for Online Social Networks: Model and Mechanisms. IEEE Trans Knowl Data Eng 25(7):1614–1627

Hu VC, Cogdell MM (2013). Guide to Attribute Based Access Control (ABAC) Definition and Considerations, National Institute of Standards and Technology, Jan. 2014, [online] Available: http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

Hu VC, Kuhn DR, Ferraiolo DF (2015) Attribute-Based Access Control. Computer 48(2):85–88

Jin X, Wah BW, Cheng X, Wang Y (2015) Significance and Challenges of Big Data Research. Big Data Res 2(2):59–64

Kaiwen S, Lihua Y (2014) Attribute-Role-Based Hybrid Access Control in the Internet of Things. In: Han W, Huang Z, Hu C, Zhang H, Guo L (eds). Web Technologies and Applications. Springer, Cham. pp 333–343

Katz J, Sahai A, Waters B (2013) Predicate encryption supporting disjunctions, polynomial equations, and inner products. J Cryptol 26(2):191–224

Kencana Ramli CDP, Nielson HR, Nielson F (2013) XACML 3.0 in Answer Set Programming. In: Albert E (ed). Logic-Based Program Synthesis and Transformation. Springer, Berlin, Heidelberg. pp 89–105

Kulkarni D (2013) A fine-grained access control model for key-value systems. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy (CODASPY '13). ACM, New York. pp 161–164. https://doi.org/10.1145/2435349.2435370

La Marra A, Martinelli F, Mori P, Rizos A, Saracino A (2017) Improving MQTT by Inclusion of Usage Control. In: Wang G, Atiquzzaman M, Yan Z, Choo K-KR (eds). Security, Privacy, and Anonymity in Computation, Communication, and Storage. Springer, Cham. pp 545–560

La Marra A, Martinelli F, Mori P, Rizos A, Saracino A (2018) Introducing Usage Control in MQTT. In: Katsikas SK, Cuppens F, Cuppens N, Lambrinoudakis C, Kalloniatis C, Mylopoulos J, Antón A, Gritzalis S (eds). Computer Security. Springer, Cham. pp 35–43

LeFevre K, Agrawal R, Ercegovac V, Ramakrishnan R, Xu Y, DeWitt D (2004). Limiting disclosure in hippocratic databases. In Proceedings of the Thirtieth international conference on Very large data bases,Toronto (Canada), Volume 30 (VLDB '04), Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer (Eds.), Vol. 30. VLDB Endowment 108-119.

Lin D, Rao P, Bertino E, Li N, Lobo J (2010) EXAM: a comprehensive environment for the analysis of access control policies. Int J Inf Secur 9(4):253–273

Longstaff JJ, Noble J (2016) Attribute based access control for big data applications by query modification. In: Second IEEE International Conference on Big Data Computing Service and Applications, BigDataService 2016, Oxford, United Kingdom, March 29 - April 1, 2016. pp 58–65

Marra AL, Martinelli F, Mori P, Saracino A (2017) Implementing Usage Control in Internet of Things: A Smart Home Use Case. In: 2017 IEEE Trustcom/BigDataSE/ICESS. pp 1056–1063

Migliavacca M, Papagiannis I, Eyers DM, Shand B, Bacon J, Pietzuch P (2010) DEFCON: High-performance Event Processing with Information Security. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference. USENIXATC'10. USENIX Association, Berkeley, CA, USA. pp 1–1

Nabeel M, Bertino E (2014) Privacy preserving delegated access control in public clouds. IEEE Trans Knowl Data Eng 26(9):2268–2280

Nehme RV, Lim HS, Bertino E (2010) FENCE: Continuous access control enforcement in dynamic data stream environments. In: 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). pp 940–943

Ong KW, Papakonstantinou Y, Vernoux R (2014) The SQL++ unifying semi-structured query language, and an expressiveness benchmark of SQL-on-Hadoop, NoSQL and NewSQL databases. CoRR. https://doi.org/abs/1405.3631

Ouaddah A, Bouij-Pasquier I, Elkalam AA, Ouahman AA (2015) Security analysis and proposal of new access control model in the Internet of Thing. In: 2015 International Conference on Electrical and Information Technologies (ICEIT). pp 30–35

Ouaddah A, Mousannif H, Elkalam AA, Ouahman AA (2017) Access control in the Internet of Things: Big challenges and new opportunities. Comput Netw 112:237–262

Paci F, Squicciarini A, Zannone N (2018) Survey on Access Control for Community-Centered Collaborative Systems. ACM Comput Surv 51(1):6–1638

Pasarella E, Lobo J (2017) A Datalog Framework for Modeling Relationship-based Access Control Policies. In: Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies (SACMAT '17 Abstracts). ACM, New York. pp 91–102. https://doi.org/10.1145/3078861.3078871

Puthal D, Nepal S, Ranjan R, Chen J (2015) Dpbsv – an efficient and secure scheme for big sensing data stream. In: 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1. pp 246–253

Rajpoot QM, Jensen CD, Krishnan R (2015) Integrating Attributes into Role-Based Access Control. In: Samarati P (ed). Data and Applications Security and Privacy XXIX. Springer, Cham. pp 242–249

Rizvi S, Mendelzon A, Sudarshan S, Roy P (2004) Extending query rewriting techniques for fine-grained access control. In: ACM SIGMOD 2004. pp 551–562

Shalabi Y, Gudes E (2017) Cryptographically Enforced Role-Based Access Control for NoSQL Distributed Databases. In: Livraga G, Zhu S (eds). Data and Applications Security and Privacy XXXI. Springer, Cham. pp 3–19

Tsankov P, Marinovic S, Dashti MT, Basin D (2014) Decentralized Composite Access Control. In: Abadi M, Kremer S (eds). Principles of Security and Trust. Springer, Berlin, Heidelberg. pp 245–264

Ulusoy H, Colombo P, Ferrari E, Kantarcioglu M, Pattuk E (2015) GuardMR: Fine-grained Security Policy Enforcement for MapReduce Systems. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. ASIA CCS '15. ACM, New York. pp 285–296

Ulusoy H, Kantarcioglu M, Pattuk E, Hamlen K (2014) Vigiles: Fine-Grained Access Control for MapReduce Systems. In: 2014 IEEE International Congress on Big Data. pp 40–47

Warmer JB, Kleppe AG (1998) The object constraint language: Precise modeling with uml (addison-wesley object technology series)

Zhang G, Tian J (2010) An extended role based access control model for the Internet of Things. In: 2010 International Conference on Information, Networking and Automation (ICINA), vol. 1. pp 1–3191323

Zhang N, Ryan M, Guelev DP (2005) Evaluating Access Control Policies Through Model Checking. In: Zhou J, Lopez J, Deng RH, Bao F (eds). Information Security. Springer, Berlin, Heidelberg. pp 446–460

Zhang X, Parisi-Presicce F, Sandhu R, Park J (2005) Formal Model and Policy Specification of Usage Control. ACM Trans Inf Syst Secur 8(4):351–387