# Issues in Access Control and Privacy for Big Data

Sylvia L. Osborn†, Daniel Servos, and Motahera Shermin

### Glossary

**Big Data** The large scale collection, processing and analysis of diverse sets of information from rapidly growing sources.

**Privacy** The protection and proper handling of data with respect to sharing with third parties, conforming to regulatory obligations and handling user consent to information collection and storage.

**Access Control** The selective restriction of access to a physical or virtual resource under some predefined model or set of policies.

**Discretionary Access Control (DAC)** An access control model that allows individual users to govern access to objects and resources they own by granting or revoking specific permissions to other users or groups.

**Mandatory Access Control (MAC)** A label-based model of access control where access is decided based on comparing the label of the subject (clearance) with the label (security level) of the object being accessed.

**Role-Based Access Control (RBAC)** A model of access control in which permissions are granted based on the organizational roles assigned to users.

**Attribute-Based Access Control (ABAC)** A policy-based access control model in which access control decision are made based on the attributes of the users, resources and the system's environment.

**Definition of the Subject** This paper explores issues in access control and privacy for big data. We highlight the differences we see between access control and privacy. We examine the requirements introduced by having big data, specifically by the V's of big data, rather than data that can be dealt with on a simple platform. Then we look at the additional challenges introduced when this big data is housed in a cloud.

## Introduction

With the advent of massive storage and massively parallel computation, coupled with the increasing adoption of internet platforms and applications, we are now in the era of big data. *Big Data* is generally considered to encompass massive

Department of Computer Science, Western University, London, Ontario, Canada, e-mail: osborn@uwo.ca, dservos5@uwo.ca, mshermin@uwo.ca
† Deceased, May 23, 2018

collections of data and the processing/analysis required on it. The data is so massive that traditional relational databases or desktop analytical tools are not adequate to perform the capture, or required analysis, with acceptable speed. The processing envisaged varies widely from traditional integration and querying tasks to complex analyses and visualizations. Sources of big data are varied, ranging from widely deployed sensors to commercial websites and social media.

Since this big data is probably not under the control of a traditional database package, there is not an obvious way to specify access control restrictions on it, and no traditional way to ensure privacy requirements are met. Both access control and privacy protection are required to protect data from unwanted access and breaches of privacy. Having these concerns dealt with in individual applications is not adequate. We are reminded of the days before database systems when multiple copies of the same data might exist within a company, and each application had its own version of the schema and its own copy of the data. Database packages brought the schema out of the individual applications so that a single version of the description of the data (the schema), and the data itself, could be managed independently of various applications. Having protection for a big data repository independent of the applications accessing it, is the goal of access control and privacy protection for big data. This goal has not yet been achieved.

We begin with a review of what we mean by access control and privacy, and how they differ. We then, discuss the four *V*'s of big data and what requirements they raise as far as achieving access control and privacy protection. Lastly we bring in the cloud, provide a summary and discuss future directions.

## Access Control and Privacy

### Access Control

Access control in traditional operating system and database environments deals with which users can perform which operations on what data. The traditional access control models are Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-based Access Control (RBAC). Discretionary access control is based on the concept of ownership of data: each user can, at their discretion, give access permissions to other users to access data which they own. This model has inherently decentralized control in that every user is making access control decisions concerning their own data, and it is difficult to formulate a universal picture of what is going on. DAC is provided by relational database packages and its administrative GRANT and REVOKE commands are part of the SQL standard (ISO, 2011). Although DAC is inherently decentralized, it is possible with a relational database system to assume that the database administrator owns all the data, and thus the database administrator can use the DAC commands to design access control for applications such that the design is, in fact, centrally administrated.

In MAC, all objects and subjects (users/applications) have a label, and access (usually restricted to read and write) is decided by comparing the label of the subject (sometimes referred to as its clearance) with the label (security level) of the object they are trying to access (Landwehr, 1981; Sandhu, 1993). MAC is generally considered to be too rigid for commercial applications.

In RBAC (Sandhu et al., 1996), permissions are gathered into roles, and roles are assigned to users. It is centrally managed and designed, and can be configured to simulate DAC and MAC (Osborn et al., 2000). Designing a system using roles is called role engineering, and can be achieved either both bottom-up or top-down (a good introduction can be found in the work by Coyne and Davis (2007)). Bottom-up design works from existing user-permission assignments to construct roles, whereas top-down design proceeds from a security designer's understanding of the job functions in a company or the requirements of various types of users in a software application. Top-down design is performed by humans and is considered slow and expensive. With bottom-up design, sometimes called role mining, it is difficult to attach semantics (or even a role name) to the results (Molloy et al., 2010). Because roles are at a coarser granularity than individual users and individual permissions, RBAC models are considered to be more efficient to manage, and easier to design. However, since an RBAC system has to be centrally designed, it may not be agile enough for a quickly changing, unpredictable application area.

More recently, Attribute-based Access Control (ABAC) has been proposed (Servos and Osborn, 2017; Jin et al., 2012; Servos and Osborn, 2014). With attribute-based access control, users acquire attributes through various means, which are then matched with object attributes to decide whether access should be allowed. It is inherently fine-grained, and allows a decentralized administration of security policy.

Once an access control system has been designed and deployed, its implementation usually includes a reference monitor, that part of a system through which all access requests must pass. One technique for implementing a reference monitor is to recode the DAC or RBAC policy as rules in XACML, and have an XACML engine act as the reference monitor (Anderson, 2004; Ferrini and Bertino, 2009).

## Privacy

Access control is used to manage all kinds of data; data about inventory, weather patterns, ocean temperatures as well as bank accounts and on-line shopping habits. Considerations of privacy deal with "sensitive" information relating to people, or perhaps organizations. Many countries and jurisdictions have regulations governing the privacy of data concerning individuals stored in on-line systems. Whereas access control for databases usually protects facts as they exist in the database, privacy concerns can require the protection of facts, summaries or partial information (Adam and Worthmann, 1989; Barker et al., 2009). Barker et al. (2009) also point out that when, for example, a customer of an on-line shopping site gives some of their

information to the shopping site, they no longer "own" it in the customary sense of DAC. It is now owned by the shopping site, which they refer to as the "house".

Dwork (2008) has used the term *curator* to refer to the process that releases information from a statistical database. Statistical databases do not allow retrieval of facts, but only allow retrieval of statistical information like averages, frequency counts, etc. In general, the curator may: (a) limit what queries are answered, (b) may limit what data is used to compute the answers or (c) may introduce "falsehoods" to comply with the privacy requirements of the statistical database. For case (a), consider a relational database tracking medical diagnoses and treatments within a hospital. Some columns, such as the name or date of birth of a patient, would be used for regular administration of the hospital, whereas other columns, the diagnosis of a patient would be a prime example, might be the subject of statistical operations for research purposes. This column would also be considered one that must be kept private. Only statistical summaries such as a frequency count would be allowed to be released for this column. Queries can be posed which read other columns but only report statistics (such as counting the frequency of each diagnosis for female patients over 65). Limiting queries can be done by limiting the size of the tuple subset which is used to compute any answer to a query, so that it never, for example, is the result of summarizing k tuples or n - k tuples, where n is the size of the table and k is quite small. Any query which involves too few, or too near the total number of tuples, is simply not answered. Even with this, Denning has shown that, using something called a tracker, and asking several well constructed queries, individual data can still be revealed (Denning and Denning, 1979).

An example of (b), limiting what data is used to compute the answer to a query, would be using random samples before calculating the statistical quantity. An example of (c), introducing "falsehoods", is computing random perturbations of the raw data before calculating the statistics. These last two techniques may result in poor quality of the resulting statistics. A summary of the trade-offs in querying statistical databases is given by Adam and Worthmann (1989).

Further to the above discussions of statistical databases is the concept of *k-anonymity* (Sweeney, 2002), which says that statistics about data should not be released unless the information for each person contained in the release cannot be distinguished from at least $k - 1$ other individuals. Additionally, any group of records which has *k-anonymity* can also be required to exhibit *l-diversity*, which means that the group contains $l$ "well represented" values for sensitive attributes (Machanavajjhala et al., 2007). A more recently proposed concept, called *differential privacy* (Dwork, 2008) says, informally, that the presence or absence of a single element will not affect the output of a query in a significant way. This again refers to release of (statistical) information from a statistical database.

Not only are statistical applications able to infer sensitive information about people which they would prefer to keep secret, many current systems track users' behaviours, say by tracking their movements through their cell phone, or tracking their interests by monitoring the web sites they visit. This data obtained through monitoring is data the user has no idea exists, so they are not able to express preferences about this type of data. Another example is a large database of medical information, which might be

| | Access Control | Privacy |
|---|---|---|
| 1 | who can access what in what manner | "sensitive" information with extra protection requirements |
| 2 | data can be about anything | data is about a person or an organization |
| 3 | usually protecting facts | can concern specific facts, partial information or existence of information; inferences |
| 4 | no purpose specification | purpose can be attached at arbitrary granularity |
| 5 | in DAC, decisions are individual, MAC and RBAC are centrally managed | decisions can be individual |

**Table 1** Differences between Access Control and Privacy Protection

used for research. The data is definitely no longer under any control of the patients who are described in it. Controls are needed so that inferences made from the data cannot be used to release private information.

Still relating to privacy, models have been introduced (such as the model by Byun and Li (2008)) to allow individual users to attach an intended purpose to parts of data about them stored in a database which is no longer under their control (the "house" situation from Barker et al. (2009)). Using these purpose specifications, if a customer's data is stored on the computers of an on-line shopping website for example, the customer could indicate that their address can be used for the shipping purpose but not for marketing. In (Byun and Li, 2008), examples show that there are cases where the labels should be attached to individual attribute values, to tuples, to columns or to whole tables. A discussion of trade-offs in implementing attribute labels is given by Mahmud and Osborn (2012).

## Access Control vs. Privacy

Access control is a straightforward protection of facts in a database and controlling who can access and manipulate these facts. When the facts pertain to data which is considered sensitive for some reason, this protection has been called privacy protection. Some contrasts between what we mean by access control and privacy are highlighted in Table 1. The first point emphasizes that privacy concerns add the idea that some information is sensitive to what would otherwise be considered access control. The second point suggest that data that is considered "sensitive" usually concerns people or human organizations, whereas things like massive sensor data, say ocean temperatures or weather data, would not be considered sensitive and that for this data, traditional access control mechanisms should be adequate. The third point suggests that for systems which can infer information from the raw data, such as statistical databases or some on-line tracking of one's web browsing habits, then extra provisions beyond traditional access control are needed. Point 4 suggests that if a system is to allow data providers to give additional privacy preferences in the form of purposes for the use of the data about them, then extra mechanisms beyond traditional access control are needed. The final point deals with how centralized/decentralized

the administration of access control and privacy protection (ACPP) is. In an on-line social network with millions of users, allowing each user to specify their access control facts is a huge challenge. If privacy purposes are allowed to be given, then a mechanism to adhere to these requirements of users can be developed, but checking these preferences in a very big database, be they just access control permissions or privacy purpose labels, can still be very challenging simply because of the volume of data.

One of the most commonly used types of applications today is on-line social networks. In an on-line social network, protection from disclosure about some of the data concerning an individual is usually at the fact level, no inferences or statistical summaries are being calculated. Within these systems, deciding how to protect ones' data is popularly referred to as a privacy setting. Since it is at the fact level, access control technologies can be used directly. However, the decisions vary widely from individual to individual, so that they present a highly decentralized challenge for traditional access control methods.

Some newer access control models blur the lines between access control and privacy. P-RBAC extends the RBAC model to support complex privacy policies that include purposes and obligations (Ni et al., 2010). Kolter et al. (2007) show how to use ABAC to incorporate the privacy preferences of the users in access control decisions.

## 4 V's of Big Data

In this section, we will discuss 4 *V*'s of big data. The basic *V*'s, Volume, Variety and Velocity, are always part of a discussion of big data (Laney, 2001; Hurt, 2012), so these can be considered as basic *V*'s. We also talk about Veracity, as it seems to be becoming an essential *V*. Others have proposed Value as a *V* (see e.g. the work by Demchenko et al. (2013)), but we do not include value here, as all data stored can be assumed to have some value, and this does not change because it is big. A diagram of the three basic *V*'s, showing how the characteristics of the data change as we move from small, centralized systems to cloud systems, can be found in the article by Hurt (2012). We have adapted this diagram to include Veracity (see Figure 1), and show some of the characteristics these *4V*'s introduce, as we move out from the origin, which represents a centralized system. It should be noted that properties mentioned in one ring in one quadrant do not necessarily go together with those in the same ring in another quadrant.

As we describe each *V* , we will also discuss what we consider to be the challenges to access control and privacy protection (ACPP) posed by this *V* .

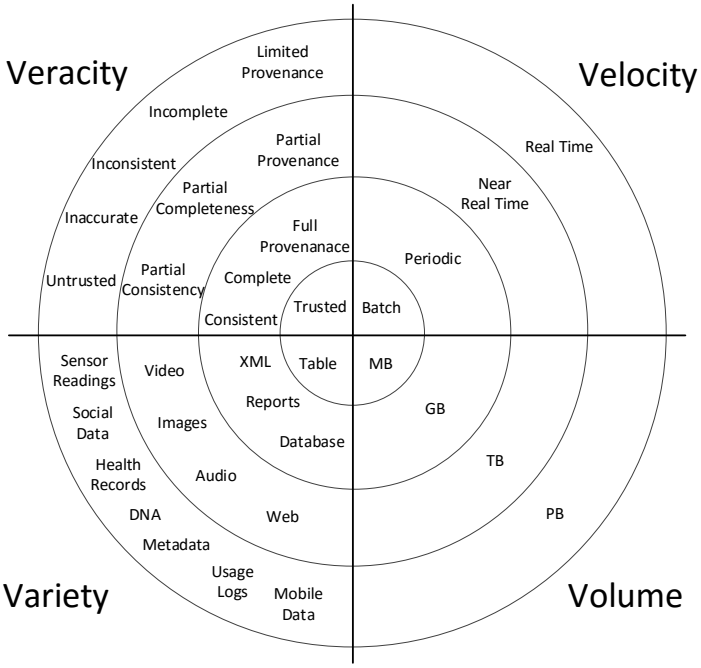


**Fig. 1** Characteristics Introduced by Increases in the *4V*'s of Big Data

## Volume

With the increasing connectedness of the world, the amount of data available for analytics is constantly increasing.

Whereas massive volumes of data pose problems for storage, processing and analytics of the data, we do not believe they add any specific requirements for ACPP, other than to say that the implementation of the ACPP mechanisms needs to be very efficient. The huge volume of data implies that the platform for the applications will not be a traditional system but probably some form of distributed architecture, commonly leveraging the cloud service model. Issues introduced by this platform will be discussed in the next Section.

## Variety

Variety indicates that the kinds of data vary widely, from traditional tuple-like database data to text, images, documents, streams of readings from sensors, results from DNA analyses, key-value pairs, etc. Within a single, large application, say

dealing with many streams of sensor readings, there might not be much variety in the types of data, but others, e.g. a news stream with images, blogs, text summaries, video etc., would have large variety to deal with. It is also unlikely that the same software package will deal with both of these examples, so rather than having a business world where everyone is using relational databases, we now have a world in which there is a large variety of software dealing with massive amounts of data. Variety also suggests a large variety of so-called privacy requirements when a large, on-line social network has millions of users.

Because of the wide variety of data and processing, ACPP mechanisms need to be available beyond what is provided by traditional database systems. Much of the data will be managed by other systems which are not traditional databases, so these systems need to have ACPP mechanisms themselves, or a way of providing ACPP outside of these applications needs to be available. To deal with a Facebook-like environment, a very decentralized access control model is required. These models need to be agile, flexible and capable of dealing with a large variety of requirements.

## Velocity

Velocity as a characteristic of big data means that in some environments, the data arrives very quickly, possibly by some streaming interface. Velocity also suggests that data types and usage patterns can change very quickly.

Considerations of access control for streaming data are not well developed. Some work on privacy for streaming data has been done (e.g. (Cao et al., 2010; Zakerzadeh and Osborn, 2013)). A centralized design of the access control model may not work well if data and usage patterns change quickly; quickly changing requirements mean the AACP mechanisms need to be agile. Velocity also requires that the implementation of the ACPP mechanisms needs to be very efficient.

## Veracity

Veracity indicates a high level of data quality, also sometimes referred to as provenance, is required. With massive amounts of data arriving at a data center, knowing that the data is relatively free of error can be an important requirement.

Traditionally in a database system, ensuring that the data is updated by qualified individuals can be ensured by restricting who can write to and update the database. This is part of an access control system. The Biba model also speaks to this (Sandhu, 1993). Others focus on the sequence of operations that have been performed on data, and its possible migration from one platform to another. Through all of these steps, trusted agents need to be handling the data to ensure its integrity. This history of the data is called its provenance. To ensure data quality, provenance and auditing of these activities is important.

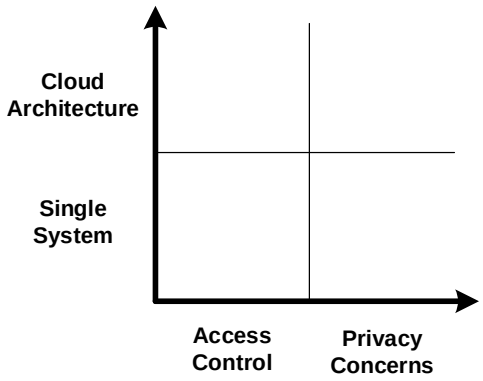


**Fig. 2** Moving From Traditional Database Access Control to Big Data Access Control and Privacy

## Putting it All Together

Big data implies that the data it too big to fit on a traditional computing platform, and that it probably resides in a cloud. NIST defines a cloud as a "model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell et al., 2011). Clouds can be private, community, public or hybrid clouds. Private clouds serve a single organization and are no different from traditional computing systems in their consideration of ACPP. Community, public and hybrid clouds may store data in a location not under control of the consumer, and be managed and controlled by entities outside of the organization that provided the data.

As documented by Mell et al. (2011), cloud computing environments are available with different service models. The most bare-bones is Infrastructure as a Service (IaaS), which just provides machines, storage and networks and requires the customer to load all their software, including the operating systems and applications. Platform as a Service (Paas) cloud providers give the customer operating systems on top of the bare-bones cloud, but the customer must provide their own applications. Software as a Service (SaaS) clouds provide other software configurations such as database packages. Many new software packages aimed at handling large amounts of data which is not necessarily relational, the so-called NoSQL databases, are focused on speed and availability, with less focus on transactional consistency and access control (Grolinger et al., 2013).

Consider the grid shown in Figure 2. The bottom, left quadrant represents traditional access control on a single system. This is well understood, and handled by traditional database management systems. What happens when we move from the bottom, left quadrant of the figure to the other quadrants? Moving to the bottom,

right quadrant means that we are more focused on privacy, still with a single system. We have more concern about the types of inference that may take place, and the labelling of parts of the data for different purposes. In some contexts we move from having a reference monitor to having a curator or auditor. The previously discussed mechanisms for handling data privacy go a long way towards satisfying concerns for this quadrant.

As we move from a single system to a private cloud, we observe that this is basically a distributed database system under the control of a single entity, for which ACPP should be achievable using well-known techniques. In moving to a non-private cloud, many new concerns are added. Many issues are introduced because the data is now outsourced. Rather than using a commercial relational database package which offers some straightforward access control, the application may now be running on one of the NoSQL platforms, which have very little ACPP protection (Shermin, 2013). We no longer know who all the users are who may be threatening our data. Cloud systems often replicate data for performance reasons. Data may cross international boundaries and be subjected to different regulations. Some mechanisms are available for this quadrant. For example, Amazon Simple Storage Service provides access control lists for data in a simple storage system (Amazon, 2020). Office 365 and App fabric are, respectively, the SaaS and PaaS platforms from Microsoft's cloud solution called Azure (Microsoft, 2020). Azure's access control system provides many mechanisms such as identity management, a rule engine and RBAC.

Moving on to the top right quadrant, the challenges are much greater. Some privacy issues are handled by access control mechanisms; if these exist for access control in cloud environments they can be applied directly to deal with these privacy concerns. Solutions dealing with things like *k-anonymity* and differential privacy can be deployed at the point of the curator in a cloud environment. Most recent publications (Colombo and Ferrari (2015) offer a good example) examined in this quadrant give lists of issues but few solutions. Cavoukian et al. (2015), on the other hand, strongly urge practitioners to use ABAC to ensure privacy for big data.

## Summary & Future Directions

In this paper we have discussed what we mean by access control and privacy protection, and compared these two requirements to highlight their differences. We have considered four *V*'s of big data, and how they add requirements to be satisfied by ACPP solutions.

Summarizing the requirements highlighted by the *V*'s, solutions for big data need to be agile, efficient, flexible, and need to scale well to very large systems. Privacy concerns suggest that the design/control of ACPP will be decentralized. Variety suggests that many different software platforms might be involved to handle a wide variety of situations, so we can no longer rely on a single technology, such as a relational database package, to handle access control for us. Due to Velocity and Variety, models might be very complicated. Velocity suggests that developers might

need to rush to a solution without incorporating mechanisms for access control or privacy.

We then looked briefly at some issues arising when data moves to a cloud computing platform. Some solutions for access control in the cloud have been developed, but mechanisms to ensure privacy in all its dimensions for very big data repositories are still lacking. New models such as ABAC are flexible and more agile than traditional access control models, but it is not yet clear that they can be made efficient for a complicated situation.

## Bibliography

N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.

Amazon. *Amazon Simple Storage Service (S3)*, 2020. URL https://aws.amazon.com/s3/. (accessed January 23, 2020).

A. Anderson. XACML profile for role based access control (RBAC). *OASIS Access Control TC committee draft*, 1:13, 2004.

K. Barker, M. Askari, M. Banerjee, K. Ghazinour, B. Mackas, M. Majedi, S. Pun, and A. Williams. A data privacy taxonomy. In *British National Conference on Databases*, pages 42–54. Springer, 2009.

J.-W. Byun and N. Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17(4):603–619, 2008.

J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan. Castle: Continuously anonymizing data streams. *IEEE Transactions on Dependable and Secure Computing*, 8(3): 337–352, 2010.

A. Cavoukian, M. Chibba, G. Williamson, and A. Ferguson. The importance of ABAC: attribute-based access control to big data: privacy and context. *Privacy and Big Data Institute, Ryerson University, Toronto, Canada*, 2015.

P. Colombo and E. Ferrari. Privacy aware access control for big data: A research roadmap. *Big Data Research*, 2(4):145–154, 2015.

E. J. Coyne and J. M. Davis. *Role engineering for enterprise security management*. Artech House, Inc., 2007.

Y. Demchenko, C. Ngo, C. de Laat, P. Membrey, and D. Gordijenko. Big security for big data: Addressing security challenges for the big data infrastructure. In *Workshop on Secure Data Management*, pages 76–94. Springer, 2013.

D. E. Denning and P. J. Denning. The tracker: A threat to statistical database security. *ACM Transactions on Database Systems (TODS)*, 4(1):76–96, 1979.

C. Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

R. Ferrini and E. Bertino. Supporting RBAC with XACML + OWL. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 145–154, 2009.

K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz. Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: advances, systems and applications*, 2(1):22, 2013.

J. Hurt. The three vs of big data as applied to conferences, 2012. URL https://velvetchainsaw.com/2012/07/20/three-vs-of-big-data-as-applied-conferences/.

ISO. ISO/IEC 9075-1-4:2011, information technology-database languages-SQL, part 1-4, 2011.

X. Jin, R. Krishnan, and R. Sandhu. A unified attribute-based access control model covering DAC, MAC and RBAC. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 41–55. Springer, 2012.

J. Kolter, R. Schillinger, and G. Pernul. A privacy-enhanced attribute-based access control system. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 129–143. Springer, 2007.

C. E. Landwehr. Formal models for computer security. *ACM Computing Surveys (CSUR)*, 13(3):247–278, 1981.

D. Laney. 3D data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1, 2001.

A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

M. S. Mahmud and S. L. Osborn. Tradeoff analysis of relational database storage of privacy preferences. In *Workshop on Secure Data Management*, pages 1–13. Springer, 2012.

P. Mell, T. Grance, et al. The NIST definition of cloud computing. 2011.

Microsoft. *Microsoft Azure Documentation*, 2020. URL https://docs.microsoft.com/en-us/azure/. (accessed January 23, 2020).

I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. Calo, and J. Lobo. Mining roles with multiple objectives. *ACM Transactions on Information and System Security (TISSEC)*, 13(4):1–35, 2010.

Q. Ni, E. Bertino, J. Lobo, C. Brodie, C.-M. Karat, J. Karat, and A. Trombeta. Privacy-aware role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):1–31, 2010.

S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):85–106, 2000.

R. S. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, 1993.

R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

D. Servos and S. L. Osborn. HGABAC: Towards a formal model of hierarchical attribute-based access control. In *International Symposium on Foundations and Practice of Security*, pages 187–204. Springer, 2014.

D. Servos and S. L. Osborn. Current research and open problems in attribute-based access control. *ACM Computing Surveys (CSUR)*, 49(4):1–45, 2017.

M. Shermin. An access control model for NoSQL databases, 2013. Master's thesis, The University of Western Ontario, 2013. Electronic Thesis and Dissertation Repository. Paper 1797.

L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

H. Zakerzadeh and S. L. Osborn. Delay-sensitive approaches for anonymizing numerical streaming data. *International journal of information security*, 12(5): 423–437, 2013.