Draft NIST Special Publication 800-210

General Access Control Guidance for Cloud Systems

Vincent C. Hu
Michaela Iorga
Wei Bao
Ang Li
Qinghua Li
Antonios Gouglidis

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-210-draft

C O M P U T E R S E C U R I T Y

National Institute of Standards and Technology
U.S. Department of Commerce

Draft NIST Special Publication 800-210

General Access Control Guidance for Cloud Systems

Vincent C. Hu
Michaela Iorga
Computer Security Division
Information Technology Laboratory

Wei Bao
Ang Li
Qinghua Li
Department of Computer Science and Computer Engineering
University of Arkansas

Antonios Gouglidis
School of Computing and Communications
Lancaster University

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-210-draft

April 2020

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-210
Natl. Inst. Stand. Technol. Spec. Publ. 800-210, 34 pages (April 2020)
CODEN: NSPUE2

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-210-draft

Public comment period: *April 1, 2020* to *May 15, 2020*

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: sp800-210-comments@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

This document presents cloud access control characteristics and a set of general access control guidance for cloud service models: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). Different service delivery models require managing different types of access on offered service components. Such service models can be considered hierarchical, thus the access control guidance of functional components in a lower-level service model are also applicable to the same functional components in a higher-level service model. In general, access control guidance for IaaS is also applicable to PaaS and SaaS, and access control guidance for IaaS and PaaS is also applicable to SaaS. However, each service model has its own focus with regard to access control requirements for its service.

Keywords

access control; access control mechanism; Cloud; cloud systems.

Acknowledgements

The authors, Vincent C. Hu of the National Institute of Standards and Technology (NIST), Bao Wei, Ang Li, and Qinghua Li of Department of Computer Science and Computer Engineering University of Arkansas, and Antonios Gouglidis of School of Computing and Communications Lancaster University wish to thank Isabel Van Wyk and David Ferraiolo (NIST) who reviewed drafts of this document. The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

   i) under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

   ii) without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to sp800-210-comments@nist.gov.

Executive Summary

Cloud systems have been developed over time and conceptualized through the combination of software, hardware components, and virtualization technologies. Characteristics of the cloud, such as resource pooling, rapid elasticity, and pay-as-you-go services, accelerated its wide adoption by industry, government, and academia. Specifically, cloud systems offer application services, data storage, data management, networking, and computing resources management to consumers over a network (the internet in general). Despite the great advancements of cloud systems, concerns have been raised about the offered level of security and privacy. The importance of these concerns becomes more evident when considering the vast number of users who have adopted cloud services.

This document presents cloud access control (AC) characteristics and a set of general access control guidance for cloud service models—IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service)—without considering deployment models (e.g., public cloud, private cloud), which require another layer of access control that depends on the security requirements of the business function or the organization of deployment for which the cloud system is implemented. Different service delivery models need to consider managing different types of access on offered service components. Such considerations can be hierarchical, such as how the access control considerations of functional components in a lower-level service model (e.g., networking and storage layers in the IaaS model) are also applicable in the same functional components in a higher-level service model (e.g., networking and storage in PaaS and SaaS models). In general, access control considerations for IaaS are also applicable to PaaS and SaaS, and access control considerations for IaaS and PaaS are also applicable to SaaS. Therefore, AC guidance for IaaS is applicable to PaaS and SaaS, and AC guidance for IaaS and PaaS is also applicable to SaaS. However, each service model has its own focus with regard to access control requirements for its service.

193 # Table of Contents

224 # List of Appendices

226

List of Figures

List of Tables

1 Introduction

1.1 Purpose

Access control (AC) dictates how principals (i.e., users and processes) can access resources based on defined AC policies to protect sensitive data and critical computing resources in the cloud. Considering the heterogeneity and remote nature of the cloud service models, AC and its general concepts should be revisited. In recent years, many works have focused on AC in cloud systems [22, 24, 25, 26]. However, these are primarily ad hoc solutions targeted at specific cloud applications and do not provide comprehensive views of cloud AC.

Cloud deployment models (e.g., public cloud, private cloud, community cloud, hybrid cloud, etc.) are configured by the scope of cloud users, services, and resources based on service requirements. This document presents a set of general AC guidance for cloud service models independent from its deployment models because it requires another layer of access control that depends on the security requirements of the business function for which the cloud system is used. As shown in Figure 3, different service models require the management of different types of access for the components of the offered service. Since such service models can be considered hierarchical, the AC considerations of functional components in a lower-level (according to Figure 2) service model (e.g., networking and storage layers in the IaaS model) are also applicable to the same functional components in a higher-level service model (e.g., networking and storage in PaaS and SaaS models). In general, AC considerations for IaaS are also applicable to PaaS and SaaS, and AC considerations for IaaS and PaaS are also applicable to SaaS. Thus, AC guidance for IaaS is applicable to PaaS and SaaS, and AC guidance for IaaS and PaaS is also applicable to SaaS. However, each service model has its own focus with regard to AC. For instance, an IaaS provider may put more effort into virtualization control, and in addition to the virtualization control, an SaaS provider needs to consider data security and the privacy of services it provides.

1.2 Scope

This document focuses on providing guidance for access control systems that are applied to an organization's cloud implementation. It does not prescribe the internal cloud access control standards that an organization may need in their enterprise systems or within a community other than the organization itself.

1.3 Audience

The intended audience for this document is an organizational entity that implements access control solutions for sharing information in cloud systems. This document assumes that readers are familiar with the cloud and access (authorization) control systems and have basic knowledge of operating systems, databases, networking, and security. Given the constantly changing nature of the information technology (IT) industry, readers are strongly encouraged to take advantage of other documents—including those listed in this document—for more current and detailed information.

1.4 Document Structure

The sections and appendices presented in this document are as follows:

- Section 1 states the purpose and scope of access control and cloud systems.
- Section 2 gives overviews of cloud access control characteristics.
- Section 3 discusses guidance for access control systems for IaaS (Infrastructure as a Service).
- Section 4 discusses guidance for access control systems for PaaS (Platform as a Service).
- Section 5 discusses guidance for access control systems for SaaS (Software as a Service).
- Section 6 discusses guidance for inter- and intra-cloud operations.
- Section 7 concludes the document with future directions.

2 Cloud Access Control Characteristics

With the support of different service models, cloud systems can provide a wide range of services to its end-users, developers, and system administrators. Cloud systems have been developed over time and conceptualized through the combination of software, hardware components, and virtualization technologies. Characteristics of the cloud, such as resource pooling, rapid elasticity, and pay-as-you-go services, have accelerated its wide adoption by industry, government, and academia. Specifically, cloud systems offer application services, data storage, data management, networking, and computing resources management to consumers[1] over a network (and the internet in general). Examples of popular cloud applications include web-based email services (e.g., Google's Gmail, Microsoft's Office 365 Outlook), data storage (e.g., Google Drive, Microsoft's OneDrive, Dropbox) for end-users, and customer relationship management and business intelligence systems (e.g., CRM Cloud, Workday) for business management. Despite the great advancements of cloud systems, concerns have been raised about offered levels of security and privacy. The importance of these concerns becomes more evident when considering the vast number of users that have adopted cloud services [1].

According to NIST, cloud computing is defined 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" [2]. Cloud computing systems may be deployed privately, hosted on the premises of a cloud customer or a provider's dedicated infrastructure, or hosted publicly by one or more cloud service providers. The system may be configured and used by one consumer or a group of trusted partners or support multi-tenancy and be used publicly by different end-users that acquire the service. Depending on the type of cloud deployment model, the cloud may have limited private computing resources or access to large quantities of remotely accessed resources. The different deployment models present a number of trade-offs in how customers can control their resources as well as the scale, cost, and availability of those resources [3]. As depicted in Figure 1, the architecture of a cloud system is composed, in general, by layers of functions:

- VM (Virtual Machine), including:
  - Applications
  - Application Programming Interface (API)
  - Operating System (OS)
- Hypervisor
- Storage
- Networking
- Hardware

A cloud service can provide access to software applications such as email or office productivity tools (i.e., the Software as a Service, or SaaS, service model), an environment for customers to build and operate their own software (i.e., the Platform as a Service, or PaaS, service model), or network access to virtualized computing resources such as processing power and storage (i.e., the

[1] In this document, **consumers** refer to system planners, program managers, technologists, and others adopting cloud computing as clients of cloud service for their **end users**. **Users** are generally applicable to both **consumers** and **end users**.

Infrastructure as a Service, or IaaS, service model). The different service models have different strengths and are suitable for different customers and business objectives [3], as illustrated in Figure 2.

A cloud system that deploys the SaaS model can be accessible over a network by an end user utilizing various client devices (e.g., a thin client interface, such as a web browser, for accessing a web-based email application) or via a program with the correct set of interfaces whose execution would enable communication with a cloud application. In the SaaS model, an application user is limited to user-specific application configuration settings and does not manage or control the underlying cloud infrastructure, which typically includes the network, servers, operating systems, storage, or individual applications.

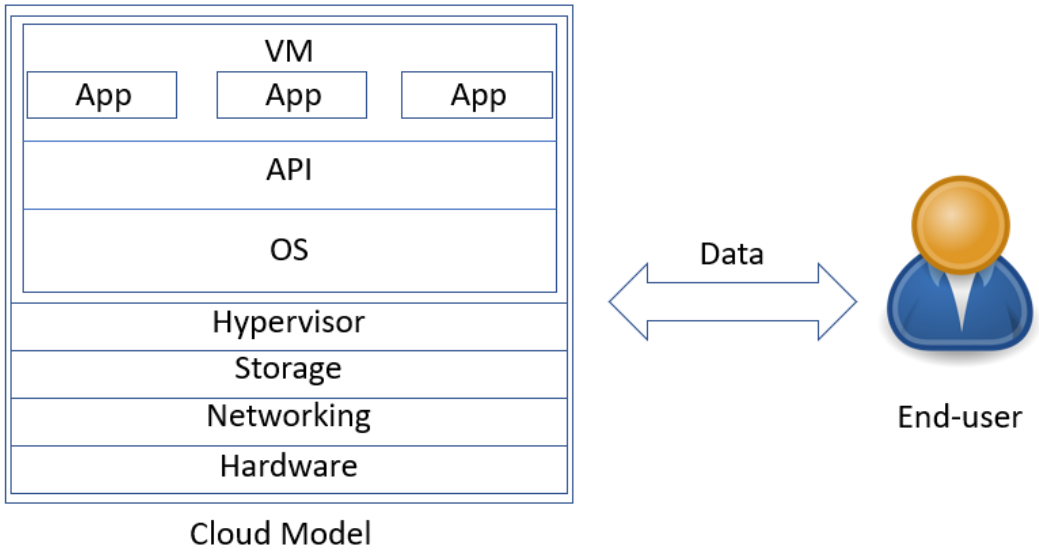

Figure 1: The general architecture of a cloud system

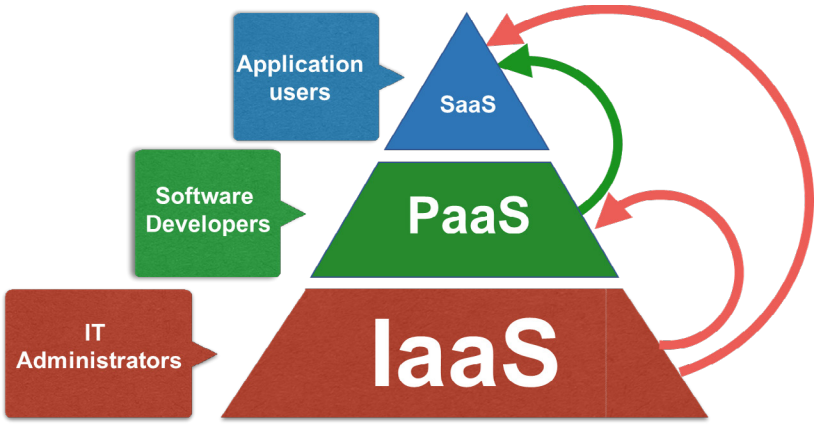

Figure 2: The service models of a cloud system

The PaaS model in a cloud system allows developers to create and deploy applications onto the cloud infrastructure using programming languages, libraries, services, and tools. A software developer does not manage or control the underlying cloud infrastructure but has control over the deployed applications (software) and, possibly, configuration settings for the application-hosting environment.

An IaaS cloud service provides computation, virtualized storage, and network resources to consumers for deploying and running arbitrary software, including operating systems and applications. Consumers may have control over virtual storage, virtualized network components, and the ability to deploy their own VMs and applications.

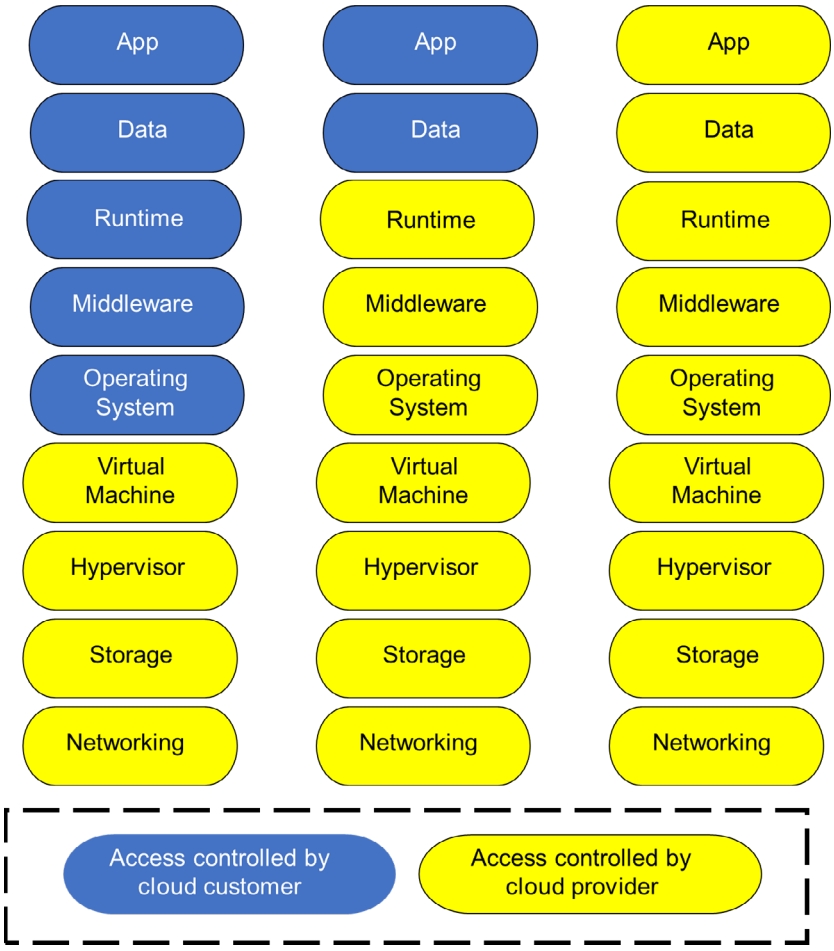

Figure 3: Accesses managed by the cloud provider and the consumer

The five essential characteristics that affect AC system design are summarized as follows [2]:

1. *Broad network access*: Cloud services are available over the network and accessible through standard mechanisms that promote use by heterogeneous thick and thin client platforms (e.g., mobile phones, tablets, laptops, workstations). This raises security concerns with regard to network access. For example, denial of service (DoS) attacks can

be launched against a cloud system, rendering its resources unavailable to legitimate users. Thus, AC for network access should be managed.

2. *Resource pooling*: The computing resources of a cloud system (e.g., storage, memory, processing, network bandwidth) are pooled to serve multiple consumers using a multi-tenant model through different physical and virtual resources, each dynamically assigned and reassigned according to consumer demands. Information may be leaked if the resource allocated to a consumer can be accessed by another co-located consumer or if the allocated resource, such as memory, is not wiped before being reallocated to another consumer. There is also a sense of location independence in that the consumer generally has no control over or knowledge of the exact location of the provided resources. Location may be specified at a higher level of abstraction (e.g., country, state, data center) that brings security concerns. Therefore, methods for implementing resource pooling while ensuring the isolation of shared resources should be considered in the AC design.

3. *Rapid elasticity*: Cloud services can be elastically provisioned and released—automatically, in some cases—to rapidly scale outward and inward commensurate with demands. To the consumer, services available for provisioning often appear to be unlimited and appropriated in any quantity at any time and are supported by adding new *virtual machines* (VMs) with specified computing resources. A challenge for AC design involves the capability to rapidly verify the security of new VMs and determine whether the newly added VMs are qualified to execute a specific task.

4. *Measured service*: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, active end user accounts). Resource usage is monitored, controlled, and reported to provide transparency to both the provider and consumer of the utilized service. To maintain resource usage, cloud consumers should be authorized to review but not modify their own metering data since this could lead to the falsification of payments required for cloud services. Thus, it is reasonable for AC to consider the protection of metering data.

5. *Data sharing*: Sharing information among different organizations is not a trivial task since a cloud system needs to meet the same security requirements of organizations to achieve that. To facilitate data sharing, concepts such as trust of federated identities and AC attributes need be considered, and building that trust is paramount. In this document, it is assumed that trust and federated identities/attributes are already established, and further discussion on that topic will be considered in another document. Regardless of the service model, consumers are entitled to be responsible for the security of their cloud-based data and, implicitly, of who has access to it [4]. For this reason, data is never controlled by cloud providers but rather always stays with the cloud customers. (The exception to this is log data, but consideration should still be given to how privacy and security is affected by such data.) Although a cloud provider might become the custodian of consumers' data, it should not have access to that data. If consumers' data is not encrypted, then cloud administrators might be able to read it. In this case, accessing data is a red flag, and customers should be aware when it is happening.

Guidance for each cloud service model, as described in Sections 3, 4, and 6 of this document, can be further extended to system requirements by referring to AC control elements listed in NIST SP 800-53, Revision 4, *Security and Privacy Control for Federal Information Systems and Organizations* [5] based on the operation requirements of the cloud service. The Appendix section maps the guidance to the AC control elements listed in the NIST SP 800-53, Revision 4.

3 Access Control Guidance for IaaS

IaaS is the cornerstone of all cloud services that offer computing and storage through a network such as the internet. Through virtualization technology, IaaS enables end users to dynamically allocate computing resources by instantiating new *virtual machines* (VMs) or releasing them based on their requirements. A VM is a software container that behaves like a physical machine with its own operating system (OS) and virtual resources (e.g., CPU, memory, hard disk, etc.). Leasing VMs is more cost-effective than purchasing new physical machines. The virtualization technology is composed of VMs and a *hypervisor,* as shown in Figure 1. VMs are managed by the hypervisor, which controls the flow of data and instructions between the VMs and the physical hardware. At the consumer side, system administrators are usually the major users of IaaS services since IaaS services are flexible to configure resources (e.g., network, data storage).

Cloud virtualization adds additional security management burdens by introducing security controls that arise from combining multiple VMs onto a single physical computer, which can have potential negative impacts if a security compromise occurs. Some cloud systems make it easy to share information among VMs by, for instance, allowing users to create multiple VMs on top of the same hypervisor if multiple VMs are available. However, this convenience can also become an attack vector since data leakage could occur among VMs. Additionally, virtualized environments are transient since they are created and vanish frequently, thereby making the creation and maintenance of necessary security boundaries more complex.

As shown in Figure 3, data in the middleware, data, applications, and OS layers is owned and controlled by the customer. The IaaS system and the customer need to ensure that access to the data is not granted to IaaS system administrators or any other IaaS customers in these layers unless any of them are permitted. IaaS administrators are responsible for access control on the virtual machine, hypervisor, storage, and networking layers and should consider Sections 3.1 – 3.5 below.

3.1 Guidance for Network

The network is shared among IaaS clients, and it is important to secure the network traffic and the cloud's environment from being exploited by unauthorized clients. Thus, access control for network boundaries and whitelists for network communications are required and may be applied through, for example, dedicated virtual local area networks (VLANs) leveraging automated access control lists (ACLs). Using the Institute of Electrical and Electronics Engineers (IEEE) 802.1Q VLAN tagging for network traffic with a cloud data center will result in routing only traffic tagged with the server's unique VLAN identifier to or from that server [6].

3.2 Guidance for Hypervisor

A hypervisor plays an important role in the security of the entire virtualized architecture since it manages customer loads and guest operating systems (OSs),[2] creates new guest OS images, and controls hardware resources. The security implications of actions like managing guest OS and hardware resources means that access to the hypervisor should be restricted to authorized cloud administrators only. Otherwise, a cloud end user could potentially obtain a VM from the cloud

[2] An OS that is secondary to the originally installed OS.

service provider and install a malicious guest OS that compromises the hypervisor by gaining unauthorized access to and altering the memory of other VMs [7]. Moreover, an attacker in a VM with lower access rights may be able to escalate their access privilege to a higher level by compromising the hardware resources allocation within the hypervisor [8]. Protecting the hypervisor from unauthorized access is therefore critical to the security of IaaS services.

3.3 Guidance for Virtual Machines

VMs that are created by different end users allow resources to be shared among multiple end users. In such a case, it must be ensured that no application from one VM can directly access other VMs since covert channels [9, 10] may leak information between VMs by accessing shared physical resources (e.g., memory). Similarly, although the ability to copy and paste information between VMs via the clipboard is a convenient feature, such a capability could be made available on other VMs running on the same hypervisor and thus introduce an attack vector (i.e., information can be leaked to other VMs through the clipboard). Organizations should have policies regarding the use of shared clipboards. Isolation between VMs is necessary to keep VMs running independently of each other, and quotas on VM resource usage should be regulated so that a malicious VM can be prohibited from exhausting computation resources. If a malicious application consumes the majority of computation resources, legitimate applications may not be able to obtain sufficient resources to perform their operations. Moreover, end users might terminate the execution of their tasks before they are finished. The state and data of the current VM would then be saved as a guest OS image, and when the task is resumed, the VM might be migrated from a different hypervisor. In such scenarios, guest OS images must be protected from unauthorized access, tampering, or storage. Furthermore, VMs that are not active may also store sensitive data. Monitoring access to the sensitive data in inactive VMs should be considered.

3.4 Guidance for APIs

There are several popular open-source platforms for deploying an IaaS cloud [11, 12, 13]. These solution platforms enable APIs to manage access control of VMs, hypervisors, and networks (note that a consumer cannot control hypervisors and networks in a multi-tenant environment unless it is a private cloud). For example, [13] consists of control components, including API, communication, lifecycle, storage, volume, scheduler, network, *API server* for managing AC policies for hypervisors, and *network Controller* for constructing network bridges and firewall AC rules. The lack of monitoring AC within these APIs might result in unenforced or wrongly enforced AC policies by the hypervisors, VMs, and networks. Thus, a service for monitoring the AC APIs in cloud platforms should also be taken into consideration.

3.5 Recommendations for IaaS Access Control

As shown in previous sections, the security of an IaaS cloud system is heavily dependent on the virtualization (hypervisor). One of the most widely adopted solutions for protecting them is a *virtualization management system* [14], which lies between the underlying hardware and the hypervisor. The virtualization management system enforces AC on both hypervisors and VMs in different ways. Virtualization management systems enforce different levels of access on different users. Some users are given read-only access to the administrative interface of a guest OS; some are allowed to control particular guest OSs; and some are given complete administrative control.

There are existing solutions for providing AC for hypervisors and VMs. For example, the approach in [15] secures the hypervisor against control hijacking attacks by protecting its code from unauthorized access and offering isolation of VMs with flexible security of mandatory access control (MAC). To enforce AC on interoperations, a well-designed service-level agreement can be applied to secure external interoperations. Other isolation mechanisms [16, 17] are helpful in ensuring the security of internal interoperations.

Guideline rules for IaaS AC policy that consider the main elements in AC (i.e., subject, object, and action) are listed in Table 1. While each row indicates a possible AC rule, the AC designer should ultimately decide whether the access in each rule is permitted or denied based on system requirements. For example, if a legitimate IaaS end user requires the use of cloud services, a login action in the hypervisor for the end user should be granted; otherwise, it should be denied.

Table 1: Potential policy rules expressed by Subject, Action, Object for IaaS AC policy

Subjects	Actions	Objects
IaaS end user	Login, Read, Write, Create	Hypervisor
IaaS end user	Read, Write, Create	VMs
VM	Write	Hypervisor
VM	Read, Write	Other VMs within the same host
VM	Read, Write, Create	Guest OS images
VM	Read, Write	Other VMs from different hosts but within the same IaaS provider
VM	Read, Write	Other VMs from different IaaS providers
Hypervisor	Read, Write, Create	Guest OS images
Hypervisor	Read, Write	Hardware resources
Hypervisor	Read, Write, Create	VMs

4 Access Control System for PaaS

PaaS is a platform that provides a framework for developers to create and deploy customized applications. As shown in Figure 3, any security assurance considerations below the data level and starting from the runtime level should be offered by the PaaS provider. The primary focus of AC in the PaaS model is to protect data during runtime, which is managed by middleware and OS. Applications have to rely on the security and privacy offered by the PaaS provider to protect their data from leaks through a covert channel introduced by unsecure shared memory. Therefore, enforcing AC over data during runtime in the PaaS is critical for the security of PaaS services.

The PaaS system administrator is responsible for the access control of runtime, middleware, OS, virtual machine, hypervisor, storage, and networking layers, as described by the guidance in Sections 4.1-4.6 below.

4.1 Guidance for Memory Data

The PaaS model permits users to deploy tasks in a provider-controlled middleware and host OS, which may be shared with other PaaS applications. As such, PaaS typically leverages OS-based techniques (e.g., Linux Containers and Docker for isolating applications) [18]. However, numerous existing memory-related attacks can compromise sensitive application-related data by hacking through the shared OS memory in PaaS [19]. Thus, AC for OS memory, such as AC of different processes on top of processor caches [20], should be considered.

4.2 Guidance for APIs

As the PaaS model allows developers to build applications on top of the platform, APIs should control the scope of each user's application such that user data remains inaccessible between different applications. In addition, packaged API can be serviced as microservices in a PaaS Cloud. A centralized architecture for provisioning and enforcement of access policies governing access to all microservices is required due to the sheer number of services needed for service composition to support real-world business transactions (e.g., customer order processing and shipping). Since each of the microservices may be implemented in a different language, policy provisioning and computation of access decisions may require the use of an authorization server [21].

4.3 Recommendations for PaaS Access Control

An efficient method should be established for protecting memory data by flushing processor caches during context switches. However, in order to avoid significant performance degradation, only highly sensitive memory data should be flushed.

Guideline rules for PaaS AC policy are listed in Table 2 with respect to the three basic elements of AC (i.e., subject, object, and action). Each row indicates a possible AC rule, but the AC designer should decide whether access should be granted or denied based on the system requirements. For example, if a user of an application needs to access memory data related to their application, permission to read memory data will be granted. However, access to that memory data will be denied to other users.

565 **Table 2: Potential policy rules expressed by Subject, Action, Object for PaaS AC policy**

Subjects	Actions	Objects
Application user	Read	Memory data
VM of a hosted application	Read, Write	Other applications' data within the same host
Application developer	Create, Read, Write	Middleware data, memory data
Cloud provider	Replicate	Application-related data

566

5 AC System for SaaS

In SaaS, a cloud provider delivers an application as a service to end users through a network such as the internet. Thus, there is no need for users to install and execute applications locally on their own computers. As shown in Figure 4, multiple applications and users can be supported simultaneously by the cloud to share common resources, including applications and underlying databases.

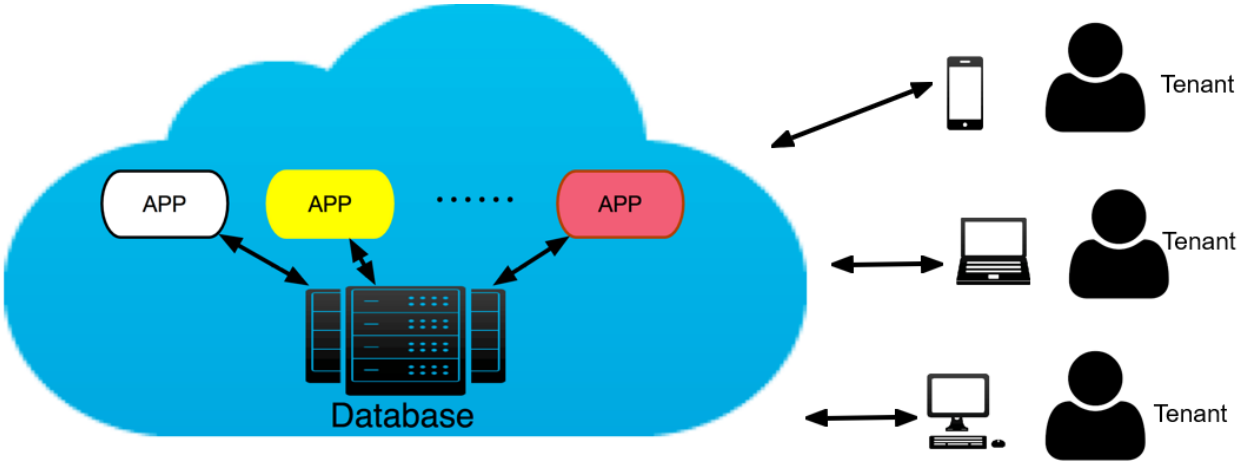

Figure 4: The multi-tenant architecture of the SaaS model

If a developer deploys a third-party application, data in that application and other unrelated applications might be stored. End users have to rely on the security and privacy offered by the cloud provider to protect their data from unauthorized access introduced by those unrelated applications. Note that data managed by the application layer is owned and controlled by the customer. The SaaS system and customer need to ensure that access to application data in these layers is not granted to the SaaS system administrator, customers, or other users unless they are trusted. SaaS administrators are responsible for the access control of all operation layers in Figure 3 and should consider the guidance in Sections 3, 4, and 5.1- 5.4.

5.1 Guidance for Data Owner's Control

A data provider is the creator or source of application data owned by consumer organizations. Application data is typically stored in the SaaS service provider's database. How a data provider manages access to its data is a challenge. Example questions to be addressed are related to data retention by the provider (e.g., where data is kept and for how long) and whether the provider has any permission to determine access rights to the data it hosts. If a data provider has the capability to determine access rights on data it holds, consideration should be given to ensure that an up-to-date AC policy is always enforced within the SaaS model.

5.2 Guidance for Confidentiality

In the application deployment model, the integrity of sensitive data residing within the data owner's domain must be protected. Protection mechanisms for application data include data encryption schemes by which data can be encrypted through certain cryptographic primitives, and

598 decryption keys will only be disclosed to authorized users [22]. For such enforcement, attribute-
599 based access control (ABAC) [23] and attribute-based encryption (ABE) schemes can be used to
600 control access to SaaS data [22, 24, 25, 26, 27] since these schemes can use the identity of users
601 through attributes to manage, encrypt, and decrypt application data. However, considering the high
602 volume of data in the SaaS model, the involved encryption and decryption significantly reduce
603 performance. Hence, when encryption is used, consideration should be given to ensure the
604 confidentiality of data while offering good performance.
605

606 ## 5.3 Guidance for Privilege Management

607 In addition to AC enforcement, privilege management involves adding, removing, and changing
608 the privileges of a subject. It is crucial to design a flexible mechanism for assigning and revoking
609 privileges to maintain the usability of the SaaS service [28].
610

611 ## 5.4 Guidance for Multiple Replicas of Data

612 To maintain high availability, the cloud provider may replicate data at multiple locations, even
613 across countries. Thus, it is important to make sure that all data replicas are protected under the
614 same AC policy. In other words, the same AC policy for the replicated data object should be
615 populated to all hosts that process the same data. The technology for policy synchronization upon
616 changes must also be considered for inclusion.
617

618 ## 5.5 Guidance for Multi-tenancy

619 The SaaS model introduces additional considerations with regard to the management of access to
620 applications. An immediate necessity is to focus on users' access to applications. The access rights
621 are granted to end users through AC policies based on predefined attributes or roles. This
622 requirement can be specified by attribute-based access control (ABAC) policy models [29, 30],
623 role-based access control [31] (RBAC), and context-based access control [32] (CBAC).
624

625 A tenant hosts a service application. The SaaS model is a typical, multi-tenancy platform that
626 supports multiple end users accessing an application simultaneously and with data of different
627 users' applications residing at the same location. Exploiting vulnerabilities in the application or
628 injecting client code into the SaaS system might expose data to other users [33]. Therefore,
629 consideration should to be given to implementing multi-tenancy while segregating data from
630 different users' applications during the design of an AC system.
631

632 ## 5.6 Guidance for Attribute and Role Management

633 In the SaaS service model, attribute and role-based AC management employs policies and
634 predefined roles to manage access rights to applications and underlying databases. The primary
635 challenge of deploying attribute or role-based AC management is reaching an agreement on what
636 types of attributes or roles should be used and what should be taken into account when designing
637 the AC systems [34]. If the set of considered attributes or roles is too small, flexibility will be
638 reduced. However, if the number of attributes or roles is too large, the complexity of policies will
639 increase.
640

5.7 Guidance for Policies

SaaS applications provide application-specific access control configurations for different user applications, and in this case, user policies for each application are enforced by the SaaS provider. This configuration does not support collaboration between the SaaS provider and the consumer's access control infrastructure. For example, while large organizations often employ on-premises access control systems for managing their users centrally and efficiently, SaaS applications typically provide organizations with an AC configuration interface for managing AC policies, which forces the AC policies to be stored and evaluated on the SaaS provider's side. This approach might result in disclosing sensitive data required for evaluating the AC policies to the SaaS provider. Therefore, methods for enforcing authorization in the SaaS provider while not disclosing sensitive access control data to the SaaS provider should be considered. Federated authorization [35] is an efficient technique that utilizes a middleware layer to transfer the management of access control policies from the SaaS provider to the consumer side and enforce policies on the SaaS applications without disclosing sensitive data required for evaluating the policies.

5.8 Guidance for APIs

An API in the SaaS model serves as an interface between the cloud server and its users. The API should be designed to protect against both accidental and malicious attempts to circumvent any AC policy. Applications for organizations and third parties often build upon the APIs, which introduce the AC complexity of the new layered API. For example, if the APIs do not require memory access for their tasks, then the AC policy for the APIs should enforce the non-memory access. Additionally, AC policies should be specified to manage the authorization process for web APIs. For example, when APIs connect through SOAP and REST protocols, the AC should control whether to allow end users to interface between Microsoft or non-Microsoft tools and technologies. For authorized API connections through SOAP and REST protocols, the AC should grant all related access requested by the protocols. For unauthorized API connections through these protocols, no access or partial access should be granted by the AC.

5.9 Recommendations for SaaS Access Control

With regard to multi-tenancy, authorization may be enforced using a *centralized*, *decentralized*, or *hybrid* authorization system. In a centralized authorization system, the SaaS provider manages a central authorization database for every end user and their accounts [36]. In a decentralized or hybrid authorization system, individual tenants are responsible for all or part of the authorization process. Note that different tenants may require different systems. Considering the attributes or roles of tenants is crucial when selecting the most suitable system. There are many ways to specify attributes or roles, such as in ABAC and RBAC models [30,31]. Attributes or roles must be well-designed and take into account hierarchy relationships when implementing AC policies for different tenants.

Authorization federation [35] is an efficient way to enforce AC policies in the SaaS provider. A generic middleware architecture that incorporates access control requirements from consumers and handles local and remote attributes or roles can be used to extend and shift AC policy management from the SaaS provider to the consumer side. This approach centralizes consumer AC policy management and lowers the required trust in the SaaS provider. In addition, the AC for VM-

supporting federation operations should also be specified (e.g., an end user may create a VM to run different applications). Within the VM of the same host, one application may need to access the application code of other applications to fulfill its task. Unlike the PaaS architecture, where consumers can fully manage the design, testing, and development of the software, SaaS consumers have limited control of the applications hosted in the cloud server.

To achieve the application data owner's control, a security class agreement (SCA) [27] may be of use. SCA is mutually agreed upon by both the data provider of PaaS subscribers and the PaaS service provider and is used for defining the security class of data providers. Multiple replicas of the same data share the same security level as its data provider. This means that given data from a particular data provider, the security class for multiple replicas of the data should be identical. As a result, the host within the PaaS service that is qualified for executing the access request can be determined by referring to the SCA. The data provider can manage access to its data by specifying security classes for the SCA to keep the data provider and the cloud host synchronized in determining the access right of data. For example, in a Bell-LaPadula model [37], assuming a patient's report is written by a doctor with confidential clearance, the report can only be read by a host with the same or higher security clearance. Additionally, when multiple data sources that are not intended to be accessed in the same cloud system are accessed, the privacy of data should not be leaked due to different security classes of these data sources and their data in the SCA. However, due to the high computation complexity of encryption and decryption, cryptographic schemes should be carefully designed to maintain the performance of cloud systems while protecting data confidentiality.

A privilege management infrastructure (PMI) [38] can be employed to dynamically manage assigning and revoking privileges through the use of attributes or role specification certificates in the PaaS model. PMI specifies the privileges for different users and links the privileges with different attribute or role specification certificates, which contain different attribute or role assignments to enforce privilege management.

To handle access control of multiple replicas of data, a method to manage the central AC policy system should be introduced. Thus, once the data within a PaaS provider is duplicated across PaaS providers, any change in the policy should result in an appropriate update to the central AC policy system. Moreover, the AC policy related to the replicated data in other PaaS providers should be synchronized accordingly based on an AC policy in the central system.

Guideline rules for SaaS AC policy are listed in Table 3. The AC designer should decide whether access in each rule is permitted or denied based on the system requirements. For example, during federation operation, VM read/write to other application code within the same host is permitted; otherwise, it is denied.

725 **Table 3: Potential policy rules expressed by Subject, Action, Object for SaaS AC policy**

Subjects	Actions	Objects
Application user	Read, Write	Application-related data
Application user	Read	Memory
Application user	Execute	Application
Application user	Read, Write	Application data
Application user	Execute	Application code
VM of a hosted application	Execute	Other application code within the same host

726

727

6 Guidance for Inter and Intra Operation

In general, collaboration (i.e., two or more systems that work together as a combined system) in the context of the cloud may lead to a seamless exchange of data and services among various cloud infrastructures. There are two types of collaborations: *inter-operation* and *intra-operation*. Inter-operation refers to the capability of using multiple cloud infrastructures. For example, as shown in Figure 4, a customer may purchase IaaS services from two different cloud providers, *Cloud A* and *Cloud B*, and the collaboration between them should be allowed due to data processing requirements.

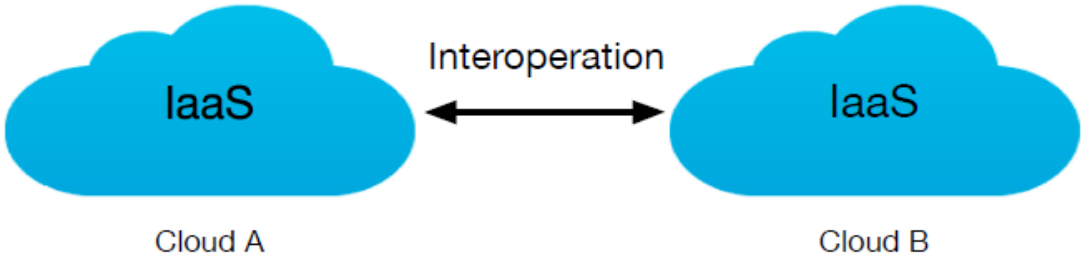

Figure 5: The external collaboration (inter-operation) between different Clouds

With regard to intra-operation, two scenarios must be considered, as shown in Figure 5. First, a customer may own multiple VMs in a single cloud host (*VM A* and *VM B*), and collaboration among those VMs may be required. Second, a customer may rent multiple hosts within the same IaaS service, and collaboration among VMs from these different hosts may be required (e.g., an interoperation between *VM B* and *VM C*).

There are some access control policy integration issues for inter-operation. For instance, different cloud providers using different sets of subject attributes for AC may cause potential conflicts or leak access permissions [39]. Attributes with the same name may result in different privileges when switching providers. Enforcing AC among different cloud providers without incurring conflicts or blocks of privilege for individual users/VMs is a challenge. This would require examining how to achieve secure inter-operation among the cloud providers [1]. Some cloud AC systems adopt centralized mechanisms to create global AC policies that manage policy integration among different cloud providers [40]. However, the cloud inter-operation is transient and thus inefficient to manage global AC policies as frequent updates for individual cloud AC policies.

With regard to intra-operation, the AC policy should enable the operations of VMs for the same customer to access each other as needed during the collaboration period and disable the access when the collaboration period ends. There are two primary cases in intra-operation: inter-host case (i.e., VMs from different cloud hosts are operating collaboratively) and intra-host case (i.e., VMs are from the same cloud host and must exchange data and services). Additionally, for some applications, VMs might be distributed in multiple host computers, so the AC policy should cover both intra-host and inter-host cases.

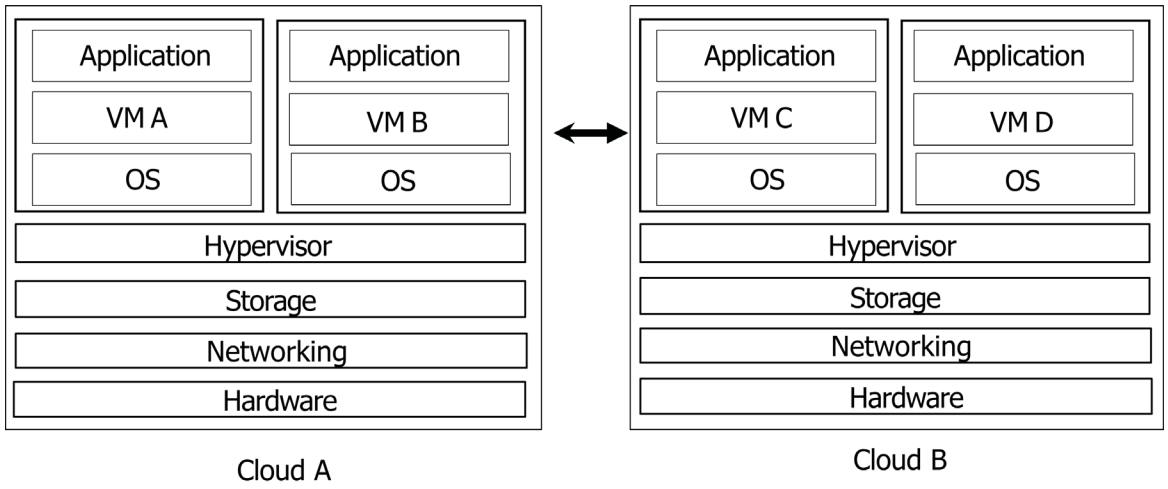

Figure 6: The internal collaboration (intra-operation) within the same Cloud

7 Conclusions

This document presents an initial step toward understanding security challenges in cloud systems by analyzing the access control (AC) considerations in all three cloud service delivery models—IaaS, PaaS, and SaaS. Essential characteristics that would affect the Cloud's AC design are also summarized, such as broad network access, resource pooling, rapid elasticity, measured service, and data sharing. Various guidance for AC design of IaaS, PaaS, and SaaS are proposed according to their different characteristics. Recommendations for AC design in different cloud systems are also included to facilitate future implementations. Additionally, potential policy rules are summarized for each cloud system. However, many issues remain open, such as AC management across different devices and platforms as well as new challenges that have yet to emerge with the wide adoption of the cloud.

References

[1] Gouglidis A, Mavridis I, Hu VC (2014) Security policy verification for multi-domains in Cloud systems. *International Journal of Information Security* 13(2):97-111. https://doi.org/10.1007/s10207-013-0205-x

[2] Mell PM, Grance T (2011) The NIST Definition of Cloud Computing. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-145. https://doi.org/10.6028/NIST.SP.800-145

[3] Badger ML, Grance T, Patt-Corner R, Voas JM (2012) Cloud Computing Synopsis and Recommendations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-146. https://doi.org/10.6028/NIST.SP.800-146.

[4] Federal Information Security Modernization Act of 2014, Pub. L. 113-283, 128 Stat. 3073. https://www.govinfo.gov/app/details/PLAW-113publ283

[5] Joint Task Force Transformation Initiative (2013) Security and Privacy Controls for Federal Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-53, Rev. 4, Includes updates as of January 22, 2015. https://doi.org/10.6028/NIST.SP.800-53r4

[6] Bartock MJ, Souppaya MP, Scarfone KA, Carroll D, Masten R, Scinta G, Massis P, Prafullchandra H, Malnar J, Singh H, Yeluri R, Shea T, Dalton M, Dukes A, Phoenix C Swarts B (2018) Trust Cloud: Security Practice Guide for VMware Hybrid Cloud Infrastructure as a Service (IaaS) Environments. (National Institute of Standards and Technology, Gaithersburg, MD), Preliminary Draft NIST Special Publication (SP) 1800-19B. Available at https://www.nccoe.nist.gov/projects/building-blocks/trusted-cloud

[7] Szefer J, Lee RB (2011) A case for hardware protection of guest VMs from compromised hypervisors in cloud computing. *2011 31st International Conference on Distributed Computing Systems Workshops (ICDCSW)* (IEEE, Minneapolis, MN), pp 248–252. https://doi.org/10.1109/ICDCSW.2011.51

[8] Krutz RL, Vines RD (2010) *Cloud security: A comprehensive guide to secure cloud computing* (Wiley Publishing, Indianapolis, IN).

[9] Wu J, Ding L, Wu Y, Min-Allah N, Khan SU, Wang Y (2014) C2detector: a covert channel detection framework in cloud computing. *Security and Communication Networks* 7(3):544–557. https://doi.org/10.1002/sec.754

[10] Rushby J (1992) Noninterference, transitivity, and channel-control security policies. (SRI International, Menlo Park, CA), Technical Report CSL-92-02. Available at http://www.csl.sri.com/papers/csl-92-2/

[11] Change ATC, Foster JL, Hall DK (1987) Nimbus-7 SMMR derived global snow cover parameters. *Annals of Glaciology* 9:39-44. https://doi.org/10.3189/S0260305500200736

[12] Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, Zagorodnov D (2009) The Eucalyptus open-source cloud-computing system. *9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09)* (IEEE, Shanghai, China), pp 124-131. https://doi.org/10.1109/CCGRID.2009.93

[13] Sefraoui O, Aissaoui M, Eleuldj M (2012) OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications* 55(3):38-42. https://doi.org/10.5120/8738-2991

[14] Scarfone KA, Souppaya MP, Hoffman P (2011) Guide to Security for Full Virtualization Technologies. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-125. https://doi.org/10.6028/NIST.SP.800-125

[15] Wang Z, Jiang X (2010) Hypersafe: A lightweight approach to provide lifetime hypervisor control-flow integrity. *2010 IEEE Symposium on Security and Privacy (SP)* (IEEE, Berkeley/Oakland, CA), pp 380–395. https://doi.org/10.1109/SP.2010.30

[16] Berger S, Cáceres R, Pendarakis D, Sailer R, Valdez E, Perez R, Schildhauer W, Srinivasan D (2008) TVDc: managing security in the trusted virtual datacenter. *ACM SIGOPS Operating Systems Review* 42(1):40–47. https://doi.org/10.1145/1341312.1341321

[17] Sailer R, Valdez E, Jaeger T, Perez R, Doorn LV, Griffin JL, Berger S (2005) sHype: Secure hypervisor approach to trusted virtualized systems. (IBM Research Division, Yorktown Heights, NY) IBM Research Report RC23511. Available at https://domino.research.ibm.com/library/cyberdig.nsf/papers/265C8E3A6F95CA8D85256FA1005CBF0F/$File/rc23511.pdf

[18] Zhang Y, Juels A, Reiter MK, Ristenpart T (2014) Cross-tenant Side-channel Attacks in PaaS Clouds. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (ACM, Scottsdale, AZ), pp 990–1003. https://doi.org/10.1145/2660267.2660356

[19] Osvik DA, Shamir A, Tromer E (2006) Cache attacks and countermeasures: the case of AES. Pointcheval D. (eds) Topics in Cryptology – CT-RSA 2006. CT-RSA 2006. Lecture Notes in Computer Science 3860 (Springer, Berlin), pp 1–20. https://doi.org/10.1007/11605805_1

[20] Tromer E, Osvik DA, Shamir A (2010) Efficient cache attacks on AES, and countermeasures. *Journal of Cryptology* 23(1):37–71. https://doi.org/10.1007/s00145-009-9049-y

[21] Chandramouli R (2019) Security Strategies for Microservices-based Application Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-204. https://doi.org/10.6028/NIST.SP.800-204

[22] Yu S, Wang C, Ren K, Lou W (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. *INFOCOM, 2010 Proceedings* (IEEE, San Diego, CA), pp 1-9. https://doi.org/10.1109/INFCOM.2010.5462174

[23] Hu VC, Ferraiolo DF, Kuhn DR, Schnitzer A, Sandlin K, Miller R, Scarfone KA (2014) Guide to Attribute Based Access Control (ABAC) Definition and Considerations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-162, Includes updates as of August 02, 2019. https://doi.org/10.6028/NIST.SP.800-162

[24] Sahai A, Waters B (2005) Fuzzy identity-based encryption. *Advances in Cryptology – EUROCRYPT 2005*. Lecture Notes in Computer Science 3494 (Springer, Berlin), pp 457–473. https://doi.org/10.1007/11426639_27

[25] Nali D, Adams CM, Miri A (2005) Using threshold attribute-based encryption for practical biometric-based access control. *International Journal of Network Security* 1(3):173–182. Available at http://ijns.jalaxy.com.tw/download_paper.jsp?PaperID=IJNS-2005-06-30-2&PaperName=ijns-v1-n3/ijns-2005-v1-n3-p173-182.pdf

[26] Zhu Y, Hu H, Ahn G-J, Huang D, Wang S (2012) Towards temporal access control in cloud computing. *INFOCOM, 2012 Proceedings* (IEEE, Orlando, FL), pp 2576–2580. https://doi.org/10.1109/INFCOM.2012.6195656

[27] Hu VC, Grance T, Ferraiolo DF, Kuhn DR (2014) An access control scheme for big data processing. *2014 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (IEEE, Miami, FL), pp 1–7. https://doi.org/10.4108/icst.collaboratecom.2014.257649

[28] Hu VC, Scarfone KA (2012) Guidelines for Access Control System Evaluation Metrics. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 7874. https://doi.org/10.6028/NIST.IR.7874

[29] Vipul G, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)* (ACM, Alexandria, VA), pp 89-98. https://doi.org/10.1145/1180405.1180418

[30] Hu VC, Kuhn DR, Ferraiolo DF, Voas J (2015) Attribute-based access control. *Computer* 48(2):85-88. http://doi.org/10.1109/MC.2015.33

[31] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. *Computer* 29(2):38-47. https://doi.org/10.1109/2.485845

[32] Rubart J (2005) Context-based access control. *Proceedings of the 2005 Symposia on Metainformatics (MIS '05)*. (ACM, New York, NY), pp 13-18. https://doi.org/10.1145/1234324.1234337

[33] Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications* 34(1), pp 1–11. https://doi.org/10.1016/j.jnca.2010.07.006

[34] Jin X, Krishnan R, Sandhu R (2012) A unified attribute-based access control model covering DAC, MAC, and RBAC. *Data and Applications Security and Privacy XXVI, DBSec 2012*. Lecture Notes in Computer Science 7371 (Springer, Berlin), pp 41-55. https://doi.org/10.1007/978-3-642-31540-4_4

[35] Decat M, Lagaisse B, Van Landuyt D, Crispo B, Joosen W (2013) Federated authorization for software-as-a-service applications. *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*. Lecture Notes in Computer Science 8185 (Springer, Berlin), pp 342–359. https://doi.org/10.1007/978-3-642-41030-7_25

[36] Dimitrios Z, Lekkas D (2012) Addressing cloud computing security issues. *Future Generation Computer Systems* 28(3):583-592. https://doi.org/10.1016/j.future.2010.12.006

[37] McLean J (1985) A comment on the 'basic security theorem' of Bell and LaPadula. *Information Processing Letters* 20(2):67-70. https://doi.org/10.1016/0020-0190(85)90065-1

[38] Blobel B, Nordberg R, Davis JM, Pharow P (2006) Modelling privilege management and access control. *International Journal of Medical Informatics* 75(8), pp 597–623. https://doi.org/10.1016/j.ijmedinf.2005.08.010

[39] Bertino E, Federica P, Rodolfo F, Shang N (2009) Privacy-preserving digital identity management for cloud computing. *IEEE Data Engineering Bulletin* 32(1):21-27. Available at http://sites.computer.org/debull/A09mar/bertino.pdf

[40] Catteddu D (2010) Cloud Computing: benefits, risks and recommendations for information security. *Web Application Security*. Communications in Computer and Information Science 72 (Springer, Berlin), pp 17-17. https://doi.org/10.1007/978-3-642-16120-9_9

Appendix A—Guidance and SP 800-53 Revision 4 Access Control (AC) Family Mapping

The following table maps the cloud access control guidance to the AC controls listed in NIST SP 800-53, Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations* [5].

Guidance	AC Control in 800-53
3.1 Guidance for Network	AC-1, AC-3, AC-4, AC-5, AC-10, AC-17, AC-21, AC-22
3.2 Guidance for Hypervisor	AC-1, AC-3, AC-5, AC-17, AC-21
3.3 Guidance for Virtual Machine	AC-1, AC-3, AC-4, AC-5, AC-11
3.4 Guidance for API	AC-1, AC-3, AC-4, AC-5, AC-11, AC-17, AC-21, AC-22
4.1 Guidance for Memory Data	AC-1, AC-3, AC-4, AC-5, AC-10, AC-11, AC-21
4.2 Guidance for APIs	AC-1, AC-3, AC-4, AC-5, AC-10, AC-11, AC-21
5.1 Guidance for Data Owner's Control	AC-1, AC-3, AC-5
5.2 Guidance for Confidentiality	AC-3, AC-6, AC-21
5.3 Guidance for Privilege Management	AC-2, AC-11, AC-14, AC-22
5.4 Guidance for Multiple Replicas of Data	AC-1, AC-3, AC-4, AC-5, AC-17, AC-21
5.5 Guidance for Multi-tenancy	AC-1, AC-2, AC-3, AC-4, AC-5, AC-10, AC-11, AC-21
5.6 Guidance for Attribute and Role Management	AC-6, AC-1, AC-3
5.7 Guidance for Policies	AC-1, AC-3
5.8 Guidance for APIs	AC-1, AC-2, AC-3, AC-4, AC-5, AC-6, AC-11, AC-14, AC-17, AC-21

AC-1: Access Control Policy and Procedures

AC-2: Account Management

AC-3: Access Enforcement

AC-4: Information Flow Enforcement

AC-5: Separation of Duties

AC-6: Lease Privilege

AC-10: Concurrent Session Control

AC-11: Session Lock

AC-14: Permitted Actions without Identification or Authentication

AC-17: Remote Access

AC-21: Collaboration and Information Sharing

AC-22: Publicly Accessible Content