SECURITY GUIDANCE

For Critical Areas of Focus In Cloud Computing **V4.0**

EXCERPT

Domain 11 for

Hewlett Packard
Enterprise

cloud security alliance®

FOREWORD

As we move towards a data driven economy, data, and how an organisation uses it, are easily becoming an organisation's business critical asset. Accompanying this shift is the increasing statutory emphasis being placed on the data privacy of individuals. Organisations wanting to move to the cloud are faced with the prospect of putting such critical assets in the hands of third party cloud providers. Naturally, that brings about lots of trepidation.

With the right approach, we at the Cloud Security Alliance, feel that such fears can be alleviated when one realises the large amount of tools, resources and knowledge that has been made available to organisations that want to move to the cloud. When it comes to data security and encryption, one such resource is Domain 11 of the CSA Security Guidance. This domain aims to provide clarity and guidance on topics such as data security controls as well as how cloud data in transit and storage can be protected.

We hope that this and the 13 other domains in the *CSA Security Guidance* will aid your journey to the cloud.

The CSA Research Team

DOMAIN 11
Data Security and Encryption

11.0 Introduction

Data security is a key enforcement tool for information and data governance. As with all areas of cloud security its use should be risk-based since it is not appropriate to secure everything equally.

This is true for data security overall, regardless of whether or not the cloud is involved. However, many organizations aren't as accustomed to trusting large amounts of their sensitive data—if not all of it—to a third party, or mixing all their internal data into a shared resource pool. As such, the instinct may be to set a blanket security policy for "anything in the cloud" instead of sticking with a risk-based approach, which will be far more secure and cost effective.

For example, encrypting everything in SaaS because you don't trust that provider at all likely means that you shouldn't be using the provider in the first place. But encrypting everything is not a cure-all and may lead to a false sense of security, e.g. encrypting data traffic without ensuring the security of the devices themselves.

By some perspectives information security is data security, but for our purposes this domain will focus on those controls related to securing the data itself, of which encryption is one of the most important.

11.1 Overview

11.1.1 Data Security Controls

Data security controls tend to fall into three buckets. We cover all of these in this section:

- Controlling what data goes into the cloud (and where).
- Protecting and managing the data in the cloud. The key controls and processes are:
  - Access controls
  - Encryption
  - Architecture

- Monitoring/alerting (of usage, configuration, lifecycle state, etc.)
- Additional controls, including those related to the specific product/service/platform of your cloud provider, data loss prevention, and enterprise rights management.

- Enforcing information lifecycle management security
  - Managing data location/residency.
  - Ensuring compliance, including audit artifacts (logs, configurations).
  - Backups and business continuity, which are covered in Domain 6.

11.1.2 Cloud Data Storage Types

Since cloud storage is virtualized it tends to support different data storage types than used in traditional storage technologies. Below the virtualization layer these might use well-known data storage mechanisms, but the cloud storage virtualization technologies that cloud consumers access will be different. These are the most common:

Object storage: Object storage is similar to a file system. "Objects" are typically files, which are then stored using a cloud platform specific mechanism. Most access is through APIs, not standard file sharing protocols, although cloud providers may also offer front-end interfaces to support those protocols.

Volume storage: This is essentially a virtual hard drive for instances/virtual machines.

Database: Cloud platforms and providers may support a variety of different kinds of databases, including existing commercial and open source options as well as their own proprietary systems. Proprietary databases typically use their own APIs. Commercial or open source databases are hosted by the provider and typically use existing standards for connections. These can be relational or non-relational—the latter includes NoSQL and other key/value storage systems, or file system-based databases (e.g. HDFS).

Application/platform: Examples of these would be a content delivery network (CDN), files stored in SaaS, caching, and other novel options.

Most cloud platforms also use redundant, durable storage mechanisms that often utilize *data dispersion* (sometimes also known as data *fragmentation of bit splitting*). This process takes chunks of data, breaks them up, and then stores multiple copies on different physical storage to provide high durability. Data stored in this way is thus physically dispersed. A single file, for example, would not be located on a single hard drive.

11.1.3 Managing Data Migrations to the Cloud

Before securing the data in the cloud most organizations want some means of managing what data is stored in private and public cloud providers. This is often essential for compliance as much or more than for security.

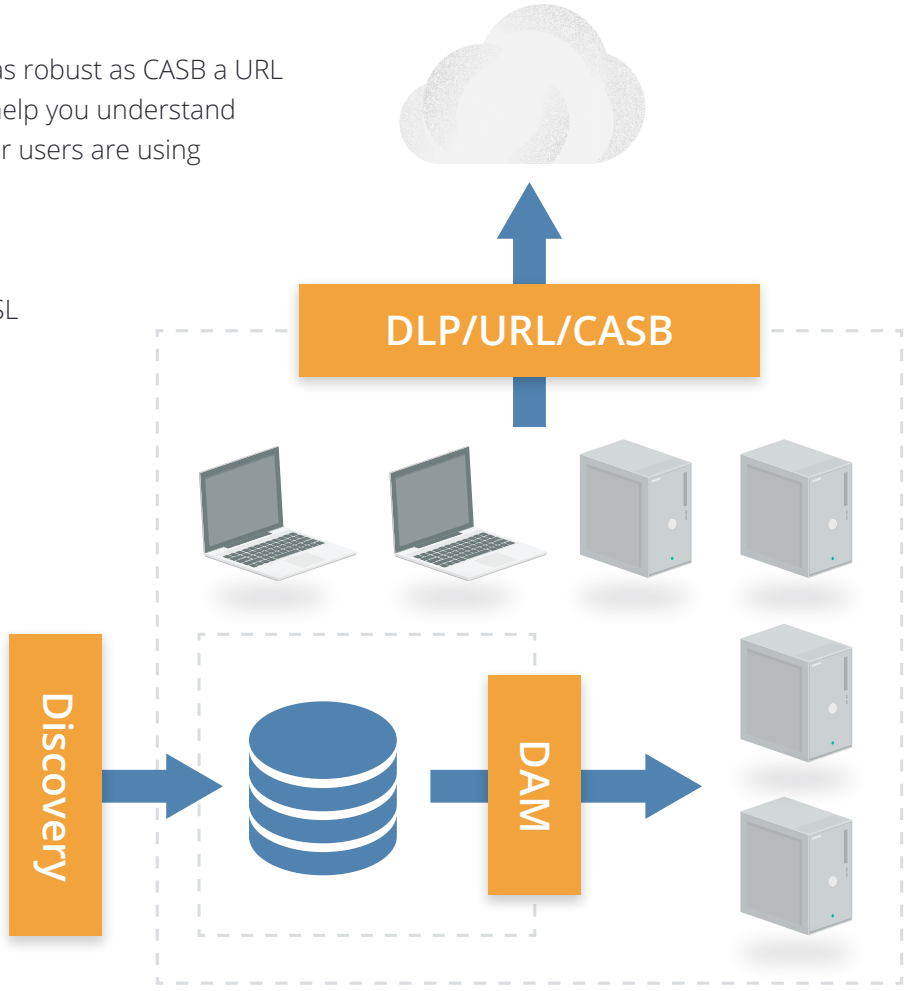
To start, define your policies for which data types are allowed and where they are allowed, then tie these to your baseline security requirements. For example, "PII is allowed on x services assuming it meets y encryption and access control requirements."
Then identify your key data repositories. Monitor them for large migrations/activity using tools like Database Activity Monitoring and File Activity Monitoring. This is essentially building an "early warning system" for large data transfers, but it's also an important data security control to detect all sorts of major breaches and misuse scenarios.

To detect actual migrations monitor cloud usage and any data transfers. You can do this with the help of the following tools:

CASB: Cloud Access and Security Brokers (also known as Cloud Security Gateways) discover internal use of cloud services using various mechanisms such as network monitoring, integrating with an existing network gateway or monitoring tool, or even by monitoring DNS queries. After discovering which services your users are connecting to, most of these products then offer monitoring of activity on approved services through API connections (when available) or inline interception (man in the middle monitoring). Many support DLP and other security alerting and even offer controls to better manage use of sensitive data in cloud services (SaaS/ PaaS/and IaaS).

URL filtering: While not as robust as CASB a URL filter/web gateway may help you understand which cloud services your users are using (or trying to use).

DLP: If you monitor web traffic (and look inside SSL connections) a DLP tool may also help detect data migrations to cloud services. However, some cloud SDKs and APIs may encrypt portions of data and traffic that DLP tools can't unravel, and thus they won't be able to understand the payload.



Managing data migrations to the cloud.

11.1.3.1 Securing Cloud Data Transfers

Ensure that you are protecting your data as it moves to the cloud. This necessitates understanding your provider's data migration mechanisms, as leveraging provider mechanisms is often more secure and cost effective than "manual" data transfer methods like SFTP. For example, sending data to a provider's object storage over an API is likely much more reliable and secure than setting up your own SFTP server on a virtual machine in the same provider.

There are a few options for in-transit encryption depending on what the cloud platform supports. One way is to encrypt before sending to the cloud (client-side encryption). Network encryption (TLS/SFTP/etc.) is another option. Most cloud provider APIs use TLS by default; if not, pick a different provider, since this is an essential security capability. Proxy-based encryption may a third option, where you place an encryption proxy in a trusted area between the cloud consumer and the cloud provider and the proxy manages the encryption before transferring the data to the provider.

In some instances you may have to accept public or untrusted data. If you allow partners or the public to send you data, ensure you have security mechanisms in place to sanitize it before processing or mixing it with your existing data. Always isolate and scan this data before integrating it.

11.1.4 Securing Data in the Cloud

Access controls and encryption are the core data security controls across the various technologies.

11.1.4.1 Cloud Data Access Controls

Access controls should be implemented with a minimum of three layers:

- *Management plane:* These are your controls for managing access of users that directly access the cloud platform's management plane. For example, logging into the web console of an IaaS service will allow that user to access data in object storage. Fortunately, most cloud platforms and providers start with default deny access control policies.
- *Public and internal sharing controls:* If data is shared externally, to the public or partners that don't have direct access to the cloud platform, there will be a second layer of controls for this access.
- *Application level controls:* As you build your own applications on the cloud platform you will design and implement your own controls to manage access.

Options for access controls will vary based on cloud service model and provider-specific features. Create an entitlement matrix based on platform-specific capabilities. An entitlement matrix documents what users, groups, and roles should access which resources and functions.

Entitlement	Super-Admin	Service-Admin	Storage-Admin	Dev	Security-Audit	Security-Admin
Volume Describe	X	X		X	X	X
Object Describe	X		X	X	X	X
Volume Modify	X	X		X		X
Read Logs	X				X	X

Frequently (ideally continuously) validate that your controls meet your requirements, paying particular attention to any public shares. Consider setting up alerts for all new public shares or for changes in permissions that allow public access.

Fine-Grained Access Controls and Entitlement Mappings

The depth of potential entitlements will vary greatly from technology to technology. Some databases may support row-level security, others little more than broad access. Some will allow you to tie entitlements to identity and enforcement mechanisms built into the cloud platform, while others rely completely on the storage platform itself merely running in virtual machines.

It's important to understand your options, map them out, and build your matrix. This applies to more than just file access, of course; it also applies to databases and all your cloud data stores.

11.1.4.2 Storage (At-Rest) Encryption and Tokenization

Encryption options vary tremendously based on service model, provider, and application/deployment specifics. Key management is just as essential as encryption, and is thus covered in a subsequent section.

Encryption and tokenization are two separate technologies. Encryption protects data by applying a mathematical algorithm that "scrambles" the data, which then can only be recovered by running it through an unscrambling (decryption) process with a corresponding key. The result is a blob of ciphertext. Tokenization, on the other hand, takes the data and replaces it with a random value. It then stores the original and the randomized version in a secure database for later recovery.

Tokenization is often used when the *format* of the data is important (e.g. replacing credit card numbers in an existing system that requires the same format text string). Format Preserving Encryption encrypts data with a key but also keeps the same structural format as tokenization, but it may not be as cryptographically secure due to the compromises.

There are three components of an encryption system: data, the encryption engine, and key management. The data is, of course, the information that you're encrypting. The engine is what performs the mathematical process of encryption. Finally, the key manager handles the keys for the encryption. The overall design of the system focuses on where to put each of these components.

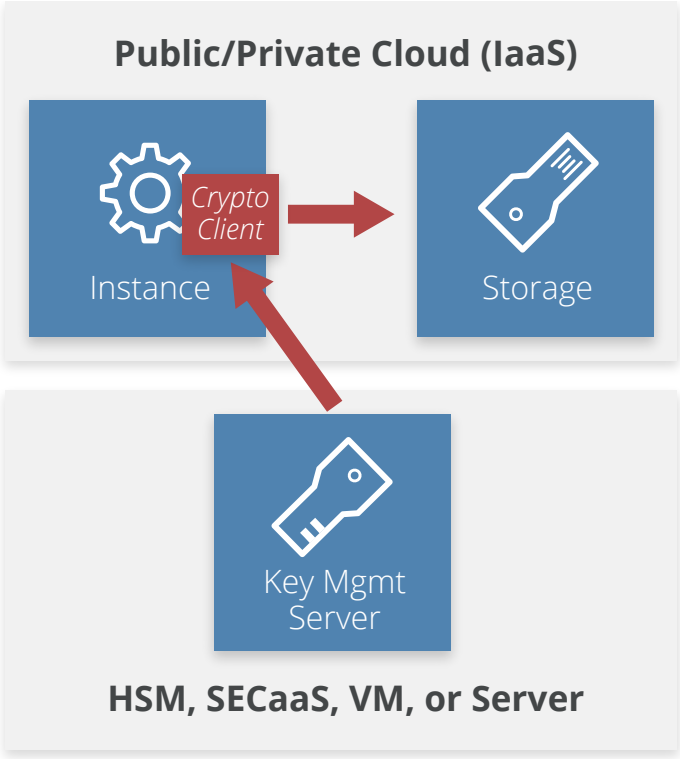
When designing an encryption system, you should start with a threat model. For example, do you trust a cloud provider to manage your keys? How could the keys be exposed? Where should you locate the encryption engine to manage the threats you are concerned with?

IaaS Encryption

IaaS volumes can be encrypted using different methods, depending on your data.

Volume storage encryption

- *Instance-managed encryption:* The encryption engine runs within the instance, and the key is stored in the volume but protected by a passphrase or keypair.
- *Externally managed encryption:* The encryption engine runs in the instance, but the keys are managed externally and issued to the instance on request.



Externally managed volume encryption

Object and file storage

- *Client-side encryption:* When object storage is used as the back-end for an application (including mobile applications), encrypt the data using an encryption engine embedded in the application or client.
- *Server-side encryption:* Data is encrypted on the server (cloud) side after being transferred in. The cloud provider has access to the key and runs the encryption engine.
- *Proxy encryption:* In this model you connect the volume to a special instance or appliance/software, and then connect your instance to the encryption instance. The proxy handles all crypto operations and may keep keys either onboard or externally.

PaaS Encryption

PaaS encryption varies tremendously due to all the different PaaS platforms.

- *Application layer encryption:* Data is encrypted in the PaaS application or the client accessing the platform.
- *Database encryption:* Data is encrypted in the database using encryption that's built in and is supported by a database platform like Transparent Database Encryption (TDE) or at the field level.
- *Other:* These are provider-managed layers in the application, such as the messaging queue. There are also IaaS options when that is used for underlying storage.

SaaS Encryption

SaaS providers may use any of the options previously discussed. It is recommended to use per-customer keys when possible, in order to better enforce multitenancy isolation. The following options are for SaaS consumers:

- *Provider-managed encryption:* Data is encrypted in the SaaS application and generally managed by the provider.
- *Proxy encryption:* Data passes through an encryption proxy before being sent to the SaaS application.

11.1.4.3 Key Management (Including Customer-Managed Keys)

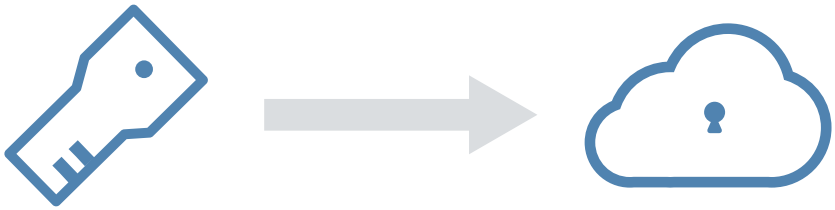
The main considerations for key management are performance, accessibility, latency, and security. Can you get the right key to the right place at the right time while also meeting your security and compliance requirements?

There are four potential options for handling key management:

- *HSM/appliance:* Use a traditional hardware security module (HSM) or appliance-based key manager, which will typically need to be on-premises, and deliver the keys to the cloud over a dedicated connection.
- *Virtual appliance/software:* Deploy a virtual appliance or software-based key manager in the cloud.
- *Cloud provider service:* This is a key management service offered by the cloud provider. Before selecting this option, make sure you understand the security model and SLAs to understand if your key could be exposed.
- *Hybrid:* You can also use a combination, such as using a HSM as the root of trust for keys but then delivering application-specific keys to a virtual appliance that's located in the cloud and only manages keys for its particular context.

Customer-Managed Keys

A customer-managed key allows a cloud customer to manage their own encryption key while the provider manages the encryption engine. For example, using your own key to encrypt SaaS data within the SaaS platform. Many providers encrypt data by default, using keys completely in their control. Some may allow you to substitute your own key, which integrates with their encryption system. Make sure your vendor's practices align with your requirements.



Customer managed keys.

Some providers may require you to use a service within the provider to manage the key. Thus, although the key is customer-managed, it is still potentially available to provider. This doesn't necessarily mean it is insecure: since the key management and data storage systems can be separated it would require collusion on the part of multiple employees at the provider to potentially compromise data. However, keys and data could still be exposed by a government request, depending on local laws. You may be able to store the keys externally from the provider and only pass them over on a per-request basis.

11.1.5 Data Security Architectures

Application architecture impacts data security. The features your cloud provider offers can reduce the attack surface, but make sure to demand strong metastructure security. For example, gap networks by using cloud storage or a queue service that communicates on the provider's network, not within your virtual network. That forces attackers to either fundamentally compromise the cloud provider or limit themselves to application-level attacks, since network attack paths are closed.

An example would be using object storage for data transfers and batch processing, rather than SFTP-ing, to static instances. Another is message queue gapping—run application components on different virtual networks that are only bridged by passing data through the cloud provider's message queue service. This eliminates network attacks from one portion of the application to the other.

11.1.6 Monitoring, Auditing, and Alerting

These should tie into overall cloud monitoring. (See Domains 3, 6, and 7.) Identify (and alert about) any public access or entitlement changes on sensitive data. Use tagging to support alerting, when it's available.

You'll need to monitor both API and storage access, since data may be exposed through either—in other words, accessing data in object storage via an API call or via a public sharing URL. Activity monitoring, including Database Activity Monitoring, may be an option. Make sure to store your logs in secure location, like a dedicated logging account.

11.1.7 Additional Data Security Controls

11.1.7.1 Cloud Platform/Provider-Specific Controls

A cloud platform or provider may have data security controls that are not covered elsewhere in this domain. Although typically they will be some form of access control and encryption, this Guidance can't cover all possible options.

11.1.7.2 Data Loss Prevention

Data Loss Prevention (DLP) is typically a way to monitor and protect data that your employees access via monitoring local systems, web, email, and other traffic. It is not typically used within data centers, and thus is more applicable to SaaS than PaaS or IaaS, where it is typically not deployed.

- *CASB (Cloud Access and Security Brokers):* Some CASBs include basic DLP features for the

sanctioned services they protect. For example, you could set a policy that a credit card number is never stored in a particular cloud service. The effectiveness depends greatly on the particular tool, the cloud service, and how the CASB is integrated for monitoring. Some CASB tools can also route traffic to dedicated DLP platforms for more robust analysis than is typically available when the CASB offers DLP as a feature.
- *Cloud provider feature:* The cloud provider themselves may offer DLP capabilities, such as a cloud file storage and collaboration platform that scans uploaded files for content and applies corresponding security policies.

11.1.7.3 Enterprise Rights Management

As with DLP, this is typically an employee security control that isn't always as applicable in cloud. Since all DRM/ERM is based on encryption, existing tools may break cloud capabilities, especially in SaaS.

- *Full DRM:* This is traditional full digital rights management using an existing tool. For example, applying rights to a file before storing it in the cloud service. As mentioned, it may break cloud provider features, such as browser preview or collaboration, unless there is some sort of integration (which is rare at the time of this writing).
- *Provider-based control:* The cloud platform may be able to enforce controls very similar to full DRM by using native capabilities. For example, user/device/view versus edit: a policy that only allows certain users to view a file in a web browser, while other users can download and/or edit the content. Some platforms can even tie these policies to specific devices, not just on a user level.

11.1.7.4 Data Masking and Test Data Generation

These are techniques to protect data used in development and test environments, or to limit real-time access to data in applications.

- *Test data generation:* This is the creation of a database with non-sensitive test data based on a "real" database. It can use scrambling and other randomization techniques to create a data set that resembles the source in size and structure but lacks sensitive data.
- *Dynamic masking:* Dynamic masking rewrites data on the fly, typically using a proxy mechanism, to mask all or part of data delivered to a user. It is usually used to protect some sensitive data in applications, for example masking out all but the last digits of a credit card number when presenting it to a user.

11.1.8 Enforcing Lifecycle Management Security

- *Managing data location/residency:* At certain times, you'll need to disable unneeded locations. Use encryption to enforce access at the container or object level. Then, even if the data moves to an unapproved location, the data is still protected unless the key moves with it.
- *Ensuring compliance:* You don't merely need to implement controls to maintain compliance, you need to document and test those controls. These are "artifacts of compliance"; this includes any audit artifacts you will have.
- *Backups and business continuity:* (see Domain 6)

11.2 Recommendations

- Understand the specific capabilities of the cloud platform you are using.
- Don't dismiss cloud provider data security. In many cases it is more secure than building your own, and comes at a lower cost.
- Create an entitlement matrix for determining access controls. Enforcement will vary based on cloud provider capabilities.
- Consider CASB to monitor data flowing into SaaS. It may still be helpful for some PaaS and IaaS, but rely more on existing policies and data repository security for those types of large migrations.
- Use the appropriate encryption option based on the threat model for your data, business, and technical requirements.
- Consider use of provider-managed encryption and storage options. Where possible, use a customer-managed key.
- Leverage architecture to improve data security. Don't rely completely on access controls and encryption.
- Ensure both API and data-level monitoring are in place, and that logs meet compliance and lifecycle policy requirements
- Standards exist to help establish good security and the proper use of encryption and key management techniques and processes. Specifically, NIST SP-800-57 and ANSI X9.69 and X9.73.