Trusted Cyber Technologies

THALES
Building a future we can all trust

The Case for Centralized Multicloud Encryption Key Management

White Paper

Contents

According to data gathered by 451 Research in support of the 2021 Thales Data Threat Report, some 19% percent of all corporate data is sensitive and stored in the Cloud. Further, the data indicated that about 41% of the data is protected by encryption.

Is the glass half-full or half-empty? We've been publishing the Data Threat Report, and some form of the above statistics, for many years. From the optimistic perspective, the percentage of sensitive data protected by encryption is always increasing. But let's hope that, perhaps within a few years, all sensitive data in the cloud will be protected by encryption. After all, 63% of those surveyed said that encryption is the top choice for securing sensitive data in the cloud.

63%

selected encryption as the top choice to secure sensitive data in the cloud.

Cloud data protection

Data encryption and keys

Cloud consumers face a choice: using the cloud service provider's (CSP) encryption or bringing their own encryption. Cloud providers make their native encryption offerings as simple as they can. For many providers, the cloud consumer can simply turn on encryption and not bother with the encryption keys. However, keeping the keys secure and separate from the data store is essential to securing the data, because encrypted data can be decrypted if the keys are available. Arguably, securely managing the keys is what digital security is all about.

The cloud is still a young industry, and reliable sources regarding cloud security are few. One we trust for cloud security best practices is the Cloud Security Alliance and their Cloud Controls Matrix, which states, in section EKM-04:

> " **Keys shall not be stored in the cloud (i.e., at the cloud provider in question), but maintained by the cloud consumer or trusted key management provider. Key management and key usage shall be separated duties."**

A leading unbiased source for cloud security best practices is telling enterprises to manage their cloud provider encryption keys separately from the CSP holding the encrypted data.

According to the *2021 Thales Data Threat Report*, only 12 percent of enterprises control all their encryption keys, and another 21 percent say they "all or mostly control their encryption keys." 60 percent say their CSP "all or mostly ... controls encryption keys." Something is clearly missing.

Best practice or critical imperative?

It is a best practice to control your encryption keys, but should it be a critical imperative? Let's review an area of IT history at a high level to help understand this:

- The oldest data-at-rest protection tool is data backup, realized as a "best practice" as far back as the early 1950's, driving the rapid evolution of tape backup devices.
- By the 1980's, regulations emerged requiring "data retention" and "data backup." For example, data retention for legal discovery is enforced by laws in many countries, states, and provinces.
- Debuting meaningfully after 2000, enterprise-class encryption (requiring secure encryption key management) for data protection starts being narrowly deployed, especially compared with ubiquitous backup.
- After 2000, regulations that mandate encryption but don't mention key management began and continued to emerge.
- During this time as well, centralized key lifecycle management recommendations, such as those from the National Institute of Standards and Technologies (NIST) and from PCI DSS for payment card data started emerging.

Based on the history of backup, we expect centralized and secure key management recommendations to become mandates before long, perhaps even with legal enforcement. If it's going to be a mandate soon, it's probably a critical imperative right now.

This is just the compliance argument. There is also the reputational argument. If you want to be as safe as possible from data breaches or other compromises to sensitive data for which you are responsible on your watch, then you want to store and manage your cloud keys separately from your CSPs. As we've seen, data breaches can lead to lost market share, depressed stock prices, and executive dismissals at the highest levels.

Managing your encryption keys separately from your CSP is obviously what you want to do. The question is: How?

Encryption key management

According to the Infosec Institute:

Encryption Key Management is the management of cryptographic keys in the cryptosystem. Key management concerns itself with keys at the user level, either between user or system. Therefore, a robust key management system is important, and policies must include the following:

- Key lifecycle: Key generation, key activation, expiration, destruction and backup/restore policies
- Physical and logical access to the key server, on a strictly need-to-know basis for business
- Role-based access control to encryption keys

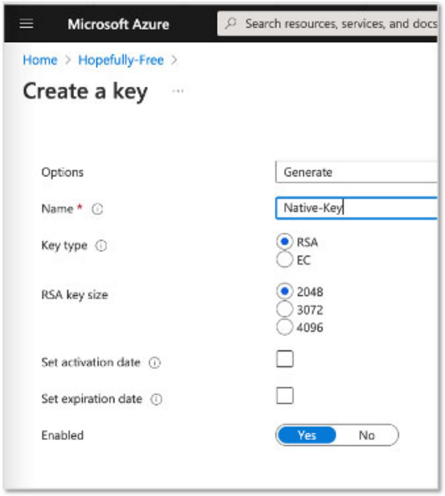
The NIST 800-57 document triplet (part 1) largely covers the same concepts. Enterprises must manage all these tasks to securely manage their keys in the cloud.

Cloud key lifecycle management defined

In the cloud, things get a bit tricky with respect to and compared with definitions provided by Infosec Institute and NIST. For example, cloud customers have no physical access to cloud provider key sources. To have a clear understanding of cloud key lifecycle management let's define a few more elements and operations:

- "Key vaults" refer to the "secure databases" where cloud providers store encryption keys (and often other information such as secrets).
- "Native keys" refer to keys that are created by the cloud provider either autonomically when needed for data encryption or upon request from a cloud customer either programmatically or in the customer-facing cloud management console.
- "Bring Your Own Key" (BYOK) could be defined as facilities offered by the cloud provider to enable customers to use their own key material instead of native but, once uploaded to the key vault, may be accessible to the CSP.
- "Hold You Own Key" (HYOK) refers to emerging technologies wherein a cloud provider requests keys from the customer's own key source, holding and using them ephemerally.
- Revocation, temporary or permanent, refers to the ability to disable use of a key in the cloud. In the case of BYOK, revocation removes the key material from the cloud key vault, rather than disabling use of it.

The following key lifecycle management operations apply to on-premises, native, BYOK, and HYOK (xYOK):



Create a key

Options: Generate

Name *: Native-Key

Key type: ● RSA ○ EC

RSA key size: ● 2048 ○ 3072 ○ 4096

Set activation date: □

Set expiration date: □

Enabled: [✓ Yes] No

Creating a native key in Microsoft Azure

- Key rotation refers to the capability to create new key material for an existing, named key.
- Key usage monitoring can vary in "depth" but consider these use cases
  - Key usage monitoring can be limited to visibility into key management operations (as above) or
  - Can go deeper, tracking actual use of keys at the level of who, when, what and, in some cases, why.
- Finally, expiration refers to key metadata governing the usable lifetime of a key.

With these definitions in mind, let's start looking at why cloud key lifecycle management can be such a challenge.

Multicloud key lifecycle managment challenges

It's a multicloud world

According to the 2021 Thales Data Threat Report, 84 percent of respondent organizations use more than 10 software as a service (SaaS) applications, and 74 percent use more than one platform as a service (PaaS) provider.

Working with multiple cloud consoles

Because it's a multicloud world, to manage your own keys in multiple clouds you'll need to log into each individual cloud console every time you need to generate a new key for a new workload and (in some cases) every time you need to rotate a key. And, if you're a large agencies, some number of people around the world will need to learn to do those key management tasks on all those different consoles to manage workloads globally.

BYOK key generation

For BYOK, you also need to determine where and how you will get your keys. The CSPs explain how to BYOK. But they do not tell you how to generate a key; they only tell you what kind of key to generate. You could use an open source tool such as OpenSSL, which is built into Linux, to generate BYOK keys. You can type a command in OpenSSL to generate a key. Unfortunately, it might not be a very strong key. The challenge with an open source technology is that if a hacker can read the source code, the hacker then understands the algorithm used to generate the random number(s) needed to create keys. If the hacker can understand the random number algorithm, the hacker is at least somewhat likely to be able to compromise open source keys. This means that the only truly safe keys originate in secure key sources, such as hardware security modules (HSMs), using the newest, secure key generation algorithms, such as elliptical curve, which are known for the best random number generation.

Cloud provider unique BYOK interfaces and processes

Generally CSPs provide a (typically proprietary) command-line interface (CLI) and/or RESTful APIs to enable you to bring your keys to the cloud. For CLI use, one must install either PowerShell for Azure, the AWS CLI, or the Google SDK (including the CLI) to bring a key to any of those clouds. So, anyone who is going to manage keys for your enterprise will have to learn those CLI mechanisms and/or the RESTful APIs for more advanced key management. And, they will have to learn them across however many cloud providers your enterprise uses. This leads to duplication of high-level key management skills distributed across people skilled with each cloud provider. This can be expensive and fraught with risks due to the well-known shortage of skilled IT security resources.

It can be maddening to learn the unique key management processes for each cloud. With kudos to the providers, we see that unique processes arise from their efforts to continue to enhance cloud data security. And, components of each process can be complex with unique acronyms or expressions that describe key management and use of key vaults. Some examples: Customer Master Key (CMK), Soft-Delete, Tenant Secret, Key Alias, Tenant, Service Principle.

To support data sovereignty needs, cloud key vaults have to be distributed around the world in data centers within a country or region. Some cloud provider consoles can require changing regions before seeing the key vault. For a globally distributed organization, this can be time-consuming.

Also, there are advanced capabilities in major CSPs that allow you to share the keys for encrypted datasets with other accounts, subscriptions, or projects, depending on the cloud provider. This requires learning even more CLI commands and APIs.

Key storage

Even if you are satisfied with the security of those open-source keys, once you use them either on-premises or in the cloud, you face the challenge of where to store those keys. Spreadsheets sound tempting for this purpose, since they act as personal databases. You can even add multiple columns for varying key metadata. And most spreadsheets can be protected with a password. However, storing keys for terabytes of data in the cloud in a spreadsheet most likely does not conform to common-sense or security best practices.

One more thing about key storage: For higher security, some CSPs take the uploaded BYOK key and combine it with one of their keys to create a "derived" key. This means that the original BYOK key sent to the cloud provider is useless without the CSP's key. For disaster recovery, providers using derived keys offer an encrypted data structure containing the combined key. This is sometimes referred to as a backup key. But the backup key needs a secure, long-term storage location or there can be no disaster recovery.

Key rotation

Best practices from the National Institute of Standards and Technology (NIST) (in the still-evolving NIST 800-57 triplet) provide guidance on **cryptoperiods** for encryption keys. Actual instantiation of the concept of cryptoperiods is visible in the operation known as **key rotation**. The NIST triplet generally refers to key **material** when talking about keys. Key rotation creates new key material, usually for a given key name or alias. This enables any operation to continue to refer to the key by the same name or alias while the key material changes. The NIST content on cryptoperiods can be utilized by organizations to define the rotation interval for key material.

Logging key access and usage

Many internal compliance policies require logging key activities. This requires more CLIs, APIs, or visits to cloud consoles. For example, you can use CLIs or APIs or visit CloudTrail in AWS. There are also tools similar to CloudTrail available from most large cloud providers.

You may need to direct your cloud key activities to your favorite security information and event management (SIEM) tool. This may require either simple or complex configuration in each cloud. And you'll face the decision of whether to pre-filter key management or filter in the SIEM. Plus, given that key management activities vary across cloud providers, it might be difficult to correlate, for example, key management activity in Microsoft Azure with the same in Amazon Web Services.

Summarizing the challenges

In summary, here are the challenges with multicloud key lifecycle management:

- Multiple cloud consoles for managing native or BYOK encryption keys
- Unique xYOK methods and corresponding unique APIs
- The need for secure key generation and storage
- Rotating keys according to both your organization's and your industry's best practices
- Keeping track of who is doing what with encryption keys using a SIEM or cloud provider tools

A crucial way to interpret the above list is to consider one of the fundamental challenges facing both the IT team generally and the IT security team specifically: complexity. Complexity is expensive, and, for the security team, complexity can induce risk. Multiple cloud consoles and unique methods create operational complexity. The need for secure key generation, storage and rotation create security complexity. Your organization manages flexibility in other IT domains with the right tools. Logging complexity is resolved with a SIEM. Backup complexity is resolved with cloud backups and backup managers.

Overcoming multicloud key lifecycle management challenges

Home grown? Let's do it ourselves.

Despite unique APIs and processes, it is clearly possible to write a home-grown software solution binding an on-premises HSM with each API and process needed for each supported cloud. But this introduces a range of considerations to add to the above list:

- Are your developers cloud environment experts? Can they keep pace with API changes and advances such as the newest HYOK offerings?
- Does your organization have early development partner program access to the cloud providers you utilize for xYOK?
- Do your developers have experience programming secure key sources? Multicloud key management is a two-sided problem with PKCS#11 key sources on one end and multiple cloud provider APIs and processes on the other.

- Complexity takes time. You'll face operational complexity and expense while your first internally-created multicloud key manager is under development
- With endlessly evolving security mandates and best practices, a complex multicloud key manager is going to be more than "v1". Once in production, you face a challenging and expensive software maintenance regimen

CipherTrust Cloud Key Manager

CipherTrust Cloud Key Manager from Thales helps overcome the challenges captured above by simplifying key management complexity and reducing operational costs with centralized key lifecycle management and visibility for cloud data encryption keys. CipherTrust Cloud Key Manager aggregates encryption key management from multiple environments, presenting all supported clouds and even multiple cloud accounts, in a single management interface. Advanced cloud key management capabilities include automated key rotation, key expiration handling, and cloud key vault synchronization. Automation tools dramatically reduce the time required for cloud key lifecycle management. CipherTrust Cloud Key Manager offers full key lifecycle management of native cloud keys – cloud key management not requiring BYOK!

Enhanced IT efficiency

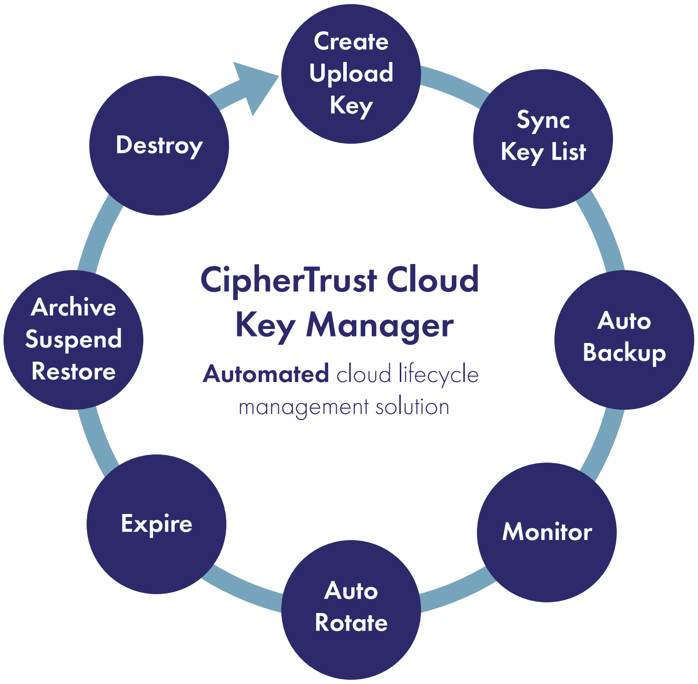
CipherTrust Cloud Key Manager offers multiple capabilities that enhance IT efficiency:

- Centralized cloud key management provides access to each cloud provider from a single browser window, including across multiple accounts or subscriptions.
- Full management of native cloud keys enables multi-cloud key management even without BYOK.
- Automated synchronization ensures that cloud console-specific key operations are reflected in centralized key management.
- Automated key rotation, including support for expiring keys, can ensure compliance while potentially saving thousands of hours per year.
- Key operation presentation in the semantics of the cloud provider, saving time and training[1].

The compliance tools you need

CipherTrust Cloud Key Manager records key activity and offers prepackaged reports enable fast compliance reporting. Log records may also be directed to a syslog server or SIEM.

Cloud Key Lifecycle Management



[1] How? While the section named, "Cloud provider unique BYOK interfaces and processes" presented challenges potentially expensive to overcome, it did not delve into cloud provider unique keys, charging methods (accounts, subscriptions, organizations) or other cloud-specific constructs. CipherTrust Cloud Key Manager brings these disparate elements together. One employee might be an Azure expert and know very little about AWS but CipherTrust Cloud Key Manager allows them to see Azure and AWS in a similar console. The AWS expert might use "key aliases" in AWS and not be clear on "soft delete" in Azure, but at least both experts see these features largely together. Seeing them together allows a person with key management responsibility to gain knowledge and skills about multicloud key management without having to become an expert in each cloud, saving time and training. In multicloud environments, cloud-specific semantics must be retained but CipherTrust Cloud Key Manager brings different semantics together.

APIs support your automation

CipherTrust Cloud Key Manager capabilities are available programmatically using RESTful APIs, enabling the power of centralized cloud encryption key management to work with your DevSecOps initiatives. The product's graphical user interface includes an "API playground" enabling you to discover many RESTful calls for automating functions.

Strong encryption key security

CipherTrust Cloud Key Manager leverages the security of CipherTrust Manager, the Vormetric Data Security Manager, or Thales Luna Network HSMs as the key source to create trusted FIPS 140-2-certified keys complying with regulations and requirements for secure key generation and storage. Such keys are more secure than those generated by OpenSSL because of the crucial need for quality random numbers to make encryption keys less penetrable.

CSP sourced keys

Some users wish to use keys sourced by their CSP. CipherTrust Cloud Key Manager provides comprehensive key management for native cloud keys for single or multiple CSPs. Enterprises can mix and match native keys with BYOK keys in a single cloud or across multiple clouds. Support for native keys means that for one cloud or all, CipherTrust Cloud Key Manager enables cloud key management without BYOK!

Secure key storage

CipherTrust Cloud Key Manager securely saves original xYOK keys. Such keys are always available in case of disaster such as accidental deletion of a key by a human being. As mentioned above, some clouds deliver a backup key, and some don't, because they use the original key. CipherTrust Cloud Key Manager manages both and keeps them straight and safe in the key sources.

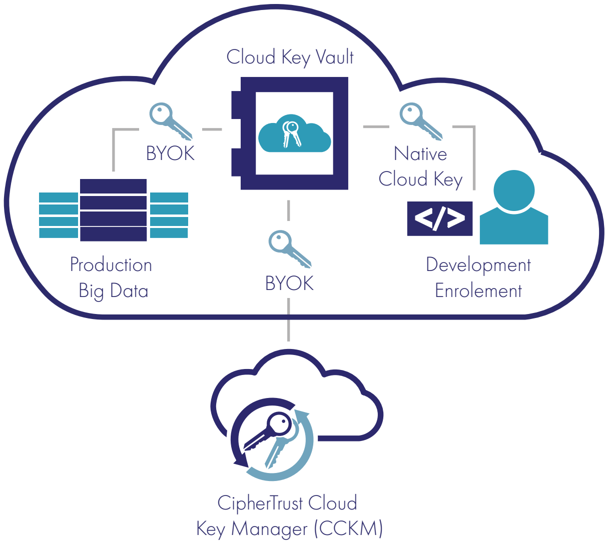
Full key lifecycle management

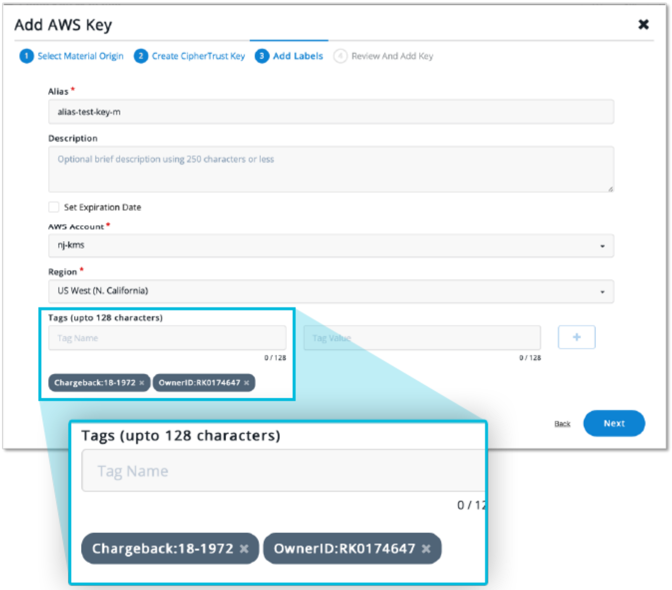
In addition to managing backup keys, CipherTrust Cloud Key Manager enables additional key operations that are not readily available in BYOK from either the command line or each provider's cloud console:

- Revocation: the use cases for removing key material from the cloud using a central console are not limited to a global cyber threat. Some examples: a key created for a development operation that's ended. Or a key created for an employee or contractor whose tenure has ended. Convenient, centralized revocation is a crucial component of cloud key lifecycle management
- Key Metadata: Nearly every cloud provider offers metadata fields which can be used to help keep your keys organized. Consider the illustration at right, where one tag identifies a chargeback department and the other the employee ID of the designated key owner

Use case: Mixing cloud BYOK and native keys

An enterprise has a massive production database in a cloud, but the enterprise also uses that cloud for its DevOps workspace. The production database requires BYOK, but the DevOps team uses native keys for developing and testing. In a situation such as this, CipherTrust Cloud Key Manager can provide BYOK keys for operations and allocate native keys for developers.





Alias *
alias-test-key-m

Description
Optional brief description using 250 characters or less

□ Set Expiration Date

AWS Account *
nj-kms

Region *
US West (N. California)

Tags (upto 128 characters)
Tag Name 0 / 128
Tag Value 0 / 128
Chargeback:18-1972 ✖ OwnerID:RK0174647 ✖

Tags (upto 128 characters)
Tag Name
0 / 12
Chargeback:18-1972 ✖ OwnerID:RK0174647 ✖

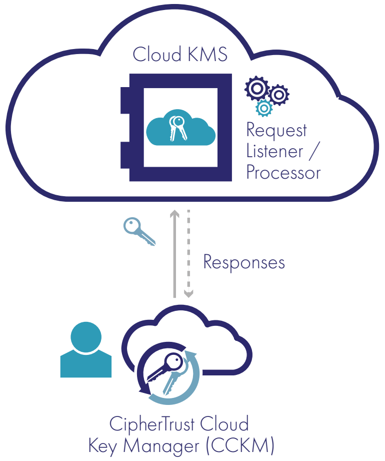
Hold your own key

HYOK has been cautiously deployed, because if the key source is unavailable, work cannot proceed. CipherTrust Cloud Key Manager provides a high availability cluster solution, which makes it ideal for both BYOK and HYOK. An increasing number of HYOK solutions are now on the market, and CipherTrust Cloud Key Manager supports many of them:

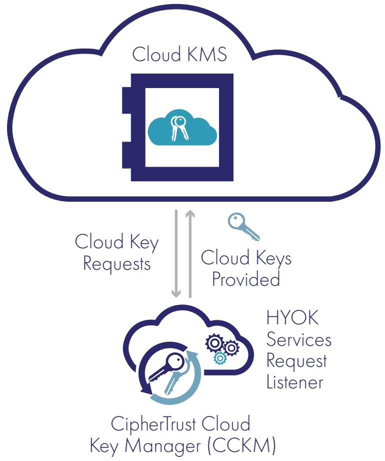
- SalesForce Cached Keys is an HYOK application, and CipherTrust Cloud Key Manager is the only solution identified by Salesforce to support it
- Google Cloud External Key Management (EKM)
- Google Workspace Client-side Encryption
- Microsoft 365 Double-Key Encryption (DKE)[2]

How HYOK and BYOK work on CipherTrust Cloud Key Manager: In BYOK, envision the cloud provider's key management system (KMS) as a **listener** to requests from CipherTrust Cloud Key Manager to manage a key. In HYOK, the situation is reversed. A range of services on CipherTrust Cloud Key Manager listen for requests from cloud providers for keys to use temporarily. The notion of **services** in support of HYOK is an example of the flexibility and power of CipherTrust Cloud Key Manager which can be difficult to implement in a home-grown solution, because both BYOK and HYOK processes differ between cloud providers.

BYOK


Cloud KMS
Request Listener / Processor
Responses
CipherTrust Cloud Key Manager (CCKM)

HYOK


Cloud KMS
Cloud Key Requests
Cloud Keys Provided
HYOK Services Request Listener
CipherTrust Cloud Key Manager (CCKM)

Flexible deployment options

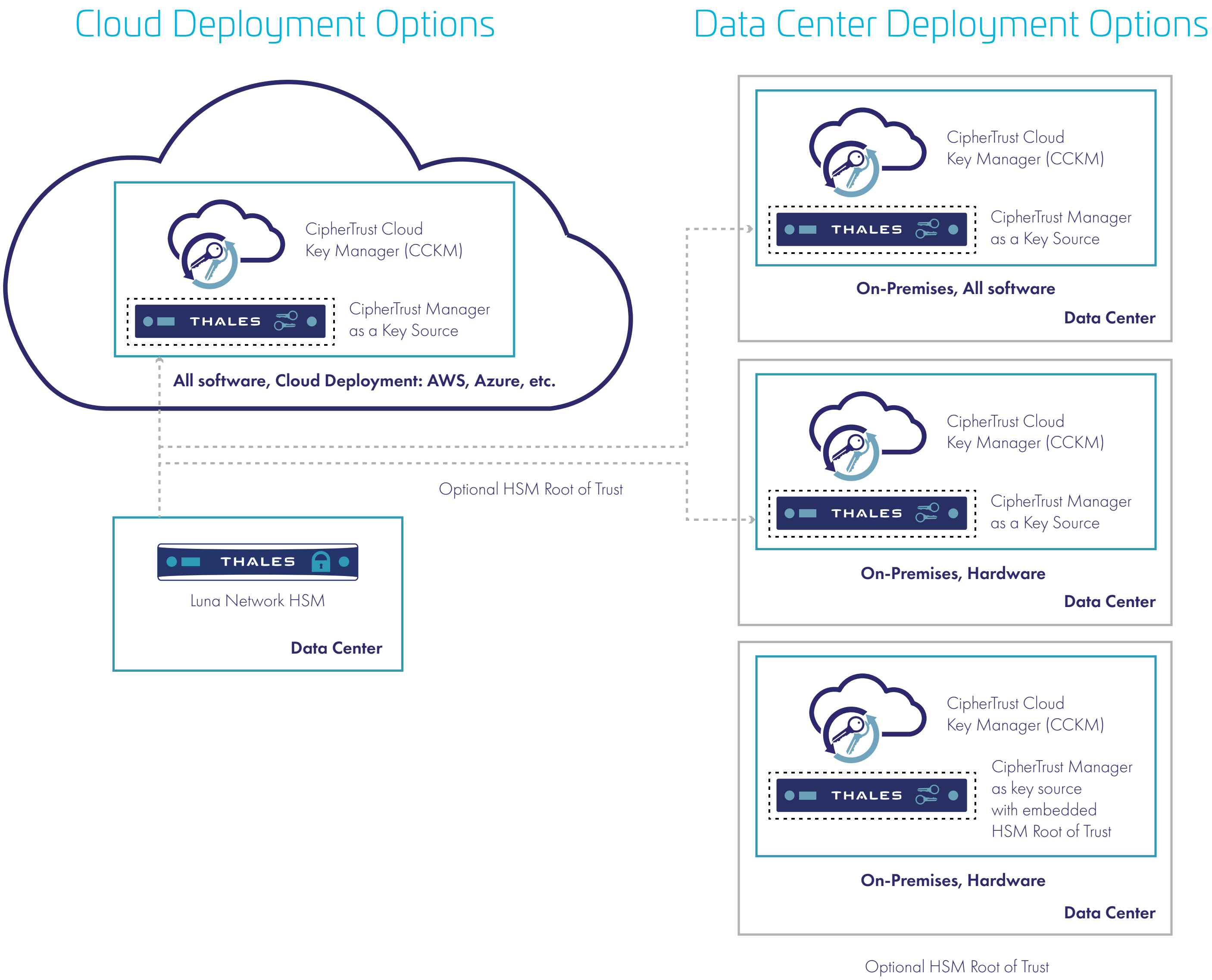
CipherTrust Cloud Key Manager is available in multiple form factors to meet any organization's needs. Both CipherTrust Cloud Key Manager and its key sources are available in all-software, cloud-friendly offerings and may be found in several cloud provider marketplaces for fast instantiation. Further, deployment in any cloud is wholly separated from cloud provider access, and keys can be managed in the cloud in which the solution is deployed as well as any other reachable, supported cloud. For example:

- A key source may be on-premises for compliance
- A CipherTrust Cloud Key Manager instance may be deployed in Amazon Web Services or any other cloud supported for deployment
- From where it is deployed it can manage keys in AWS, Salesforce or Azure or other supported clouds

Many other deployment architectures are available. For example, one edition of CipherTrust Cloud Key Manager can reside inside a secure physical appliance with its key source.

[2] On the CipherTrust Cloud Key Manager roadmap and supported on both Thales Luna Cloud HSM and Luna Network HSM.

Flexible deployment options for CipherTrust Cloud Key Manager



Cloud Deployment Options
Data Center Deployment Options
CipherTrust Cloud Key Manager (CCKM)
THALES
CipherTrust Manager as a Key Source
All software, Cloud Deployment: AWS, Azure, etc.
Optional HSM Root of Trust
THALES
Luna Network HSM
Data Center
CipherTrust Cloud Key Manager (CCKM)
THALES
CipherTrust Manager as a Key Source
On-Premises, All software
Data Center
CipherTrust Cloud Key Manager (CCKM)
THALES
CipherTrust Manager as a Key Source
On-Premises, Hardware
Data Center
CipherTrust Cloud Key Manager (CCKM)
THALES
CipherTrust Manager as key source with embedded HSM Root of Trust
On-Premises, Hardware
Data Center
Optional HSM Root of Trust

Conclusion

Multicloud consumption is pervasive, with sensitive data stored across many clouds. There are numerous challenges around managing the keys for data protected with cloud provider encryption. CipherTrust Cloud Key Manager simplifies creating, deploying, and managing the full lifecycle of cloud encryption keys and help fulfill industry and organizational data protection mandates. In addition, Thales multi-cloud security products, including Bring Your Own Advanced Encryption, all with centralized key management, enable you to encrypt and control cloud storage to help comply with government regulations and industry mandates and eliminate reputational and financial losses in the event of a cloud data breach.

About Thales Trusted Cyber Technologies

Thales Trusted Cyber Technologies, a business area of Thales Defense & Security, Inc., protects the most vital data from the core to the cloud to the field. We serve as a trusted, U.S. based source for cyber security solutions for the U.S. Federal Government. Our solutions enable agencies to deploy a holistic data protection ecosystem where data and cryptographic keys are secured and managed, and access and distribution are controlled. For more information, visit www.thalestct.com

THALES

Building a future we can all trust

Contact us

Please visit thalestct.com/contact-us

> thalestct.com <