Multi-Cloud Key Management

Version 1.0
Updated: Thursday, May 4, 2017

Securosis, L.L.C. 515 E. Carefree Highway Suite #766 Phoenix, AZ 85085 **T** 602-412-3051
info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog, but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

This report is licensed by Thales eSecurity.

THALES

ABOUT THALES E-SECURITY - Thales e-Security is the leader in advanced data security solutions and services that deliver trust wherever information is created, shared or stored. We ensure that the data belonging to companies and government entities is both secure and trusted in any environment – on-premises, in the cloud, in data centers or big data environments – without sacrificing business agility. Security doesn't just reduce risk, it's an enabler of the digital initiatives that now permeate our daily lives – digital money, e-identities, healthcare, connected cars and with the internet of things (IoT) even household devices. Thales provides everything an organization needs to protect and manage its data, identities and intellectual property and meet regulatory compliance – through encryption, advanced key management, tokenization, privileged user control and high assurance solutions. Security professionals around the globe rely on Thales to confidently accelerate their organization's digital transformation. Thales e-Security is part of Thales Group.

Please visit http://www.thalesesecurity.com/ and find us on Twitter @thalesesecurity

Copyright

Multi-Cloud Key Management

Table of Contents

Encryption Keys and Cloud Computing

Running IT systems on public cloud services is the reality for most companies. Just about every company uses Software as a Service to some degree; many have already migrated back-office systems such as email, collaboration, file storage, and customer relationship management. But we are now seeing the core of the data center — financial systems, databases, supply chain, and enterprise resource planning — moving to public Platform and Infrastructure "as a Service" (PaaS & IaaS) providers. It is now common for medium and large enterprises to run SaaS, PaaS, and IaaS at different providers, all in parallel with on-premises systems. Some small firms we speak with no longer have data centers, instead relying on third parties to host all their applications.

Cloud services offer an alluring cocktail of benefits: cost effectiveness, reliability, agility, and *security*. While several of these advantages have always been accepted, security was the last major hurdle for customers. So cloud service providers focused on customers' security concerns and now offer extensive capabilities for data, network, and infrastructure security. In fact most customers can realize equivalent or better security in the cloud than they can provide in-house. With the removal of this last impediment we see more and more firms embrace IaaS for critical applications.

Infrastructure as a Service entails handing over ownership and operational control of your IT infrastructure to a third party. But responsibility for data security cannot go along with it. Your provider ensures compute, storage, and networking components are secure from external attackers and their other tenants, but you must protect your data and application access to it. Some of you trust your cloud providers, while others do not. Or you might trust one cloud service but not others. Regardless, to maintain control of your data you must engineer cloud security controls to ensure compliance with internal security requirements, as well as regulatory and contractual obligations. In some cases you will leverage security capabilities provided by a cloud vendor, and in others you will bring your own and run them atop the cloud.

Encryption is *the* fundamental security technology in modern computing. So it should be no surprise that encryption technologies are everywhere in cloud computing. The vast majority of cloud service providers enable network encryption by default to protect data in transit and prevent man-in-the-middle attacks. Most cloud providers offer encryption for data at rest, both to protect files and archives from unwanted inspection by authorized infrastructure managers, and in case of data leaks from the cloud service. In many ways encryption is another commodity feature of the cloud service you pay for. But it is only effective when encryption keys are properly protected. Just as with on-

premises systems, when you move data to cloud services, it is critical to properly manage and secure the keys.

Controlling encryption keys — and thus also your data — while adopting cloud services is one of the more difficult puzzles in moving to the cloud. For example you need to decide who creates keys (you or your provider), where they are managed (on-premises or in-cloud), how they are stored (hardware or software), how keys will be maintained, how to scale up in a dynamic environment, and how to integrate with each different cloud model you use (SaaS, PaaS, IaaS, and hybrid). And you still need to either select your own encryption library or invoke your cloud service to encrypt on your behalf. You get an excellent selection of choices to meet any use case, but piecing it all together is a challenge.

This paper addresses the central question we keep getting: How can I manage encryption keys on someone else's hardware? Customers need to know if there is a consistent way to handle this across all their cloud services. We will discuss challenges specific to multi-cloud key management and why customers may need something different than the native encryption options offered by their cloud providers. We will help you select the right strategy from the many possible combinations. Finally, we discuss how each customer requirements maps to different deployment options, and what to look for in a key management system.

Use Cases

There is a common set of data security concerns shared by customers moving into cloud services. But their inquires are often based on on-premises approaches, and less relevant in the cloud. To provide some context, one of the major mental adjustments security folks need to make when moving to cloud services is where their responsibilities begin and end. You are no longer responsible for physical security of cloud systems and cannot control the security of resource pools such as compute, storage, and network, so your areas of concern move "up the stack". With IaaS you control applications, data, user access, and network accessibility. With SaaS you are limited to data and user access. Any cloud service limits the tools at your disposal to those either provided natively by the cloud or third-party tools which work with that cloud service. Fortunately the cloud also reduces your set of responsibilities.

Fielding customer calls on data security over the last decade, we have learned that inquiries regarding on-premises systems typically start with the data repository. Typical questions are, "How can I protect my database?", "Our SAN vendor provides encryption, but what threats does that protect us from?" and "How do I protect sensitive data on my file servers?" In these conversations, once we understand the repository and the threats to address, we can construct a data security plan. They usually center on some implementation of encryption with supporting key management, access management, and possibly masking or tokenization technology. In the cloud, encryption is still the primary to tool for data security, but the starting points of conversations are different. The issues are more about needs than driven by particular threats. The following are the main concerns cited by customers:

- **PII:** Personally Identifiable Information — essentially sensitive data specific to a user or customer — is the top concern. PII includes things like Social Security Numbers, credit card numbers, account numbers, passwords, and other sensitive data types defined by various regulations. And it's very common for what companies move into — or derive inside — the cloud to contain sensitive customer information. Other types of sensitive data are present as well, but PII compliance requirements drive our data protection conversations. The regulation might be GLBA, Mass Privacy Regulation 201 CMR 17, NIST 800-53, FedRAMP, PCI-DSS, HIPAA, or another from the evolving list, and that's just in the USA. The mapping of these requirements to on-premises security controls has always been fuzzy, and it hasn't been clarified by the changes in how cloud services function. So IT staff and external auditors are uncertain whether and how well these requirements are really addressed. Leveraging existing encryption keys and tools helps ensure consistency with current policy and process.

- **Legal Compliance and Nation-State Attacks:** Many companies have information that is attractive to governments and intelligence services. They worry both about their own government as well as foreign ones. Others worry that litigation may result in a subpoena for all their data. In either case the customer needs to ensure their cloud providers cannot be compelled to turn over encryption keys, either within the country they run their servers, or if their applications migrate data or applications to a different country. If the vendor is never provided with your encryption keys, they cannot be compelled to turn them over, ensuring data privacy and control.

- **Trust:** More precisely, the problem is lack of trust. Some customers simply do not trust their vendors. Many security pros, having seen security products and platforms fail repeatedly during their careers, view cloud security with a jaundiced eye. They are especially hesitant with security systems they cannot fully audit. Or they do not have faith that cloud vendors' IT staff cannot access their data. In some cases they do not trust software-based encryption.

- **Vendor Lock-in and Migration:** A common concern is vendor lock-in, and an inability to migrate to another cloud service provider. Services fail, competitors emerge and contractual relationships can become untenable. Some native cloud encryption systems do not allow customer keys to move outside the system, and cloud encryption systems are based on proprietary interfaces. The goal is to maintain protection regardless of where data resides, moving between cloud vendors as needed.

- **Jurisdiction:** Cloud service providers, especially IaaS vendors, offer services in multiple countries, often in more than one region, with redundant data centers. This redundancy is great for resilience, but regulatory concerns arises when moving data across regions which may have different laws and jurisdictions. For example the EU's General Data Protection Regulation (GDPR) governs the personal data of EU citizens, and applies to any foreign company regardless of where data is moved. While similar in intent and data types to the US regulation mentioned above under 'PII', it further specifies that citizen data **must not** be available in foreign countries that do not meet specific requirements. Many SaaS and IaaS security models cannot accommodate such data-centric concerns. In these cases, segregation of duties and access controls can be augmented by key management.

- **Consistency:** Firms often adopt a "best of breed" cloud approach. They leverage multiple IaaS providers, placing each application on the service which best fits that application's particular requirements. Most firms are quite familiar with their on-premises encryption and key management systems, so they often prefer to leverage the same tool and skills across multiple clouds. This minimizes process changes around key management and application changes to support different API.

Obviously the nuances of each cloud implementation guide these conversations as well. Not all services are created equal, so what works in one may not be appropriate in another. But the major

vendors offer very strong encryption implementations. Concerns such as data exfiltration protection, storage security, volume security, database security, and protecting data in transit can all be addressed with provided tools. That said, some firms cannot fully embrace a cloud native implementation, typically for regulatory or contract reasons. These firms have options to maintain control over encryption keys and leverage cloud-native or third-party encryption.

Service Type & Deployment

In this section we discuss basic approaches for managing keys in public cloud services and how it all works. We illustrate deployment options and the components of a solution. We will walk through the process of getting a key from your on-premises Hardware Security Module (HSM) into a cloud HSM. We will discuss variations on using cloud-based HSMs for all encryption operations, as well as when you instead delegate encryption operations to the cloud-native encryption service. We'll also discuss software-based (non-HSM) key management systems running on IaaS cloud services.

There are two basic design approaches for managing your keys in a third-party cloud service. The most common model is generally referred to as 'BYOK' (Bring Your Own Key). As the name implies you place your own keys in a cloud HSM, and use your keys with the cloud HSM service to encrypt and decrypt content. This model requires HSMs but supports all cloud service models (SaaS, PaaS, and IaaS) so long as the cloud vendor offers an HSM service. The alternative is software-based key management. In this case you run the same key management software you use on-premises, but in a multi-tenant IaaS cloud. Your key management vendor supplies either a server or a Docker container containing the software, and you configure and deploy it in your cloud environment.

Let's jump into the specifics of each model, and the different ways each approach is used.

BYOK

Cloud platforms for commercial services offer encryption as an option for data storage and communications. With most cloud environments — especially SaaS — encryption is built-in and occurs by default for all tenants as part of the service. To keep things simple encryption and key management interfaces are not exposed — instead encryption is a transparent function handled on the customer's behalf. For select cloud services where stronger security is required, or regulations demand their use, Hardware Security Modules are offered as an option. These modules are physically and digitally hardened against attack to ensure that keys are secure from tampering and difficult to misuse.

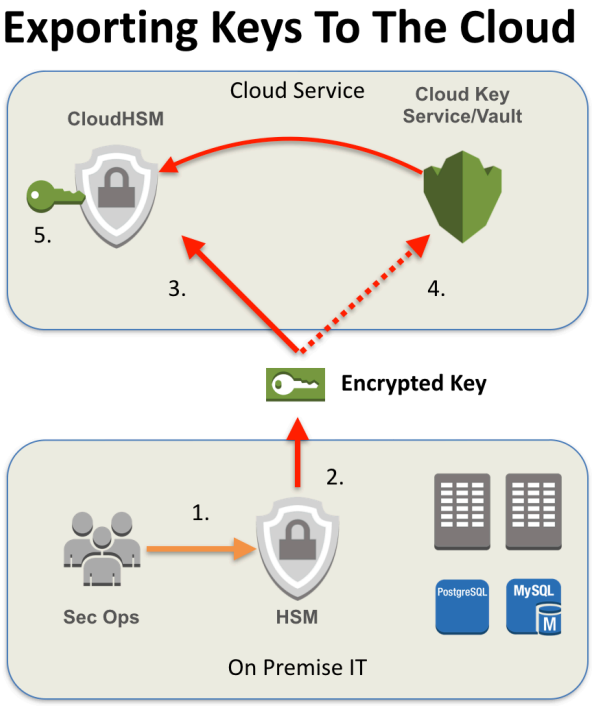
To incorporate HSMs into a cloud service, cloud vendors typically offer an extension to their key management service. In some cases it's a simple set of additional API, but in most cases a dashboard is provided with an API for provisioning and key management. In some cases, particularly when you use the same type of HSM on-premises as your cloud vendor, its full suite of HSM functions may be available. So the amount of work required to set up BYOK varies.

Let's take a closer look at getting keys into the cloud.

Exporting Keys

Those of you using HSMs on-premises understand that typically keys remain fully protected within the HSM, and are *never* extracted. It is possible to extract keys, but extracted keys are encrypted with additional information to render them usable only for import into another compatible HSM. Each customer HSM is seeded by the vendor with information about the vendor and customer, and some configuration regarding permitted uses for its keys. This non-key configuration information, along with keys from the vendor and customer, are used to 'bootstrap' an HSM. This process can be reversed to extract keys, but traditionally not for use outside another HSM — traditionally only to seed another appliance.

Key extraction is a manual process for most HSMs. It typically involves two or more security administrators providing credentials and a smart card or USB stick with a secure enclave to authenticate to the HSM, then requesting a key for extraction. For most HSMs, extraction is similar: Once validation occurs, the HSM takes the customer's master key and bundles it with information regarding the HSM vendor and customer, and in some cases usage rights for the key, then encrypts it all together. These added data elements provide additional protections for the key, dictating where it can be decrypted and how it may be used.



Export of keys does not occur through any specific channel, and is not synchronous with import into a destination HSM. Instead the encrypted information bundle is sent to the cloud service provider. A cloud HSM service likely leverages a cluster of several HSMs, not the traditional model of a 'Active/Standby' pair. And each cloud vendor implements their own integration layer, so details on how keys are imported vary, as does level of effort. In general once a customer has been provisioned for the cloud HSM service they can import their master key via a dashboard, API, or command line. The customer's master key bundle is used to create their intermediate keys as needed for their cloud key hierarchy, and those intermediate keys in turn are used to generate data encryption keys as needed. These encryption keys are copied between cloud HSM as needed for failover and replication.

Each cloud provider scales up and maintains redundancy in its own ways, and they typically do not publish details. Instead they provide service guarantees for uptime and performance. The good news is you no longer need to worry much about the specifics because they are taken care of for you. Additionally, cloud service providers do not as a rule use Active/Standby HSM pairs, preferring a

more scalable ‘cloud’ of many hardware modules, importing customer keys as needed, so resiliency is likely better than whatever you have on-premises today.

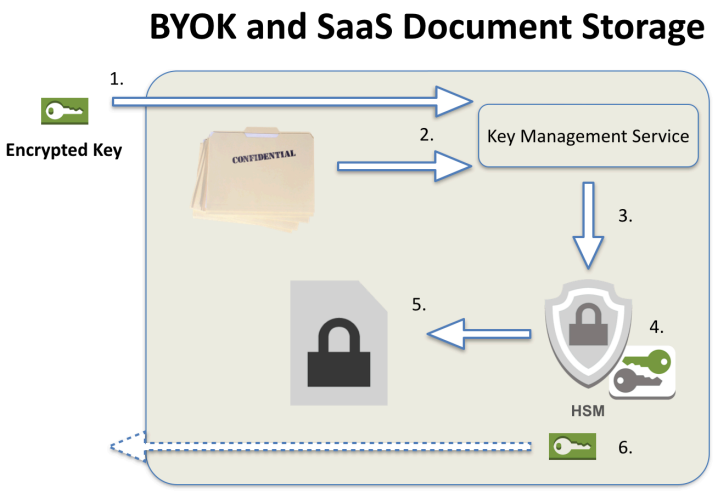
Keep in mind that hardware-based key management support is still considered a special case by cloud service vendors. Not all customers demand it. And it is often not fully available as a self-service feature — there may be a manual sign-up process and availability in only specific regions or zones. Unlike built-in native encryption, HSM capabilities cost extra.

Once you have your keys installed in the cloud HSM service, you can use it to encrypt data. The way this works varies between cloud service models so we will look at a couple common cases.

SaaS with HSM Encryption

With many SaaS services, if you contract for a cloud-based HSM service, all encryption operations on your behalf are performed inside the HSM. The native cloud encryption service may satisfy requests on your behalf so encryption and decryption are transparent, but key access and cryptographic operations are kept within the HSM.

The following graphic illustrates how some cloud document services provide BYOK. (1) Your key is imported into the cloud and the key management interface installs your master key into the HSM, as described earlier. (2) Users upload documents to the cloud, and (3) the key management service requests a new data encryption key for each uploaded document. (4) A new key is generated, and (5) the uploaded document is encrypted prior to being stored in the cloud. (6) In some cases the newly derived encryption key is now encrypted under the customer’s master key and sent back to the customer for on-premises storage in their HSMs.



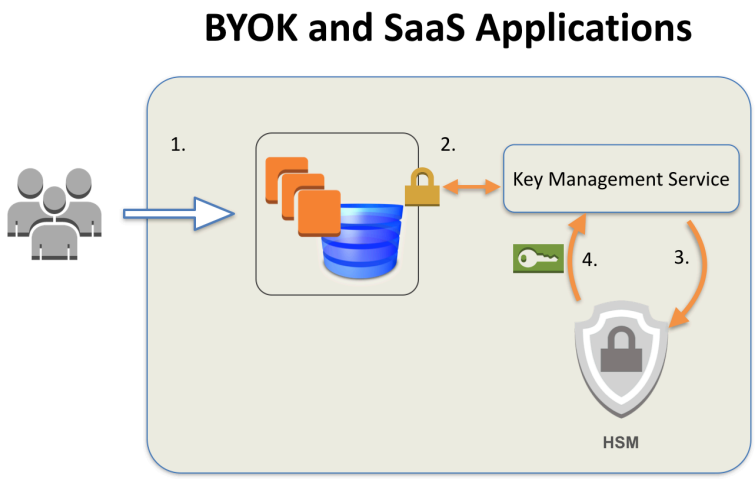
There are differences between providers but this basic framework is common. Other types of SaaS offerings, where encryption and decryption within the HSM is infeasible (for the vendor), use a slightly different process.

SaaS/PaaS HSM with Native Encryption

In some cases customers bring their own 'master' keys to the cloud, but the cloud provider uses data encryption keys derived from the master for actual encryption and decryption outside the HSMs. Consider a SaaS application running complex reports, or a PaaS service where the customer stores terabytes of information encrypted within a database. As the cloud vendor controls all of the underlying hardware and software, they can choose if encryption is done in hardware or software services, while maintaining security of the derived encryption keys. Keep in mind that for PaaS and SaaS, the application and supporting databases are part of the vendor's underlying service, so they have several options for how and where to encrypt and decrypt.

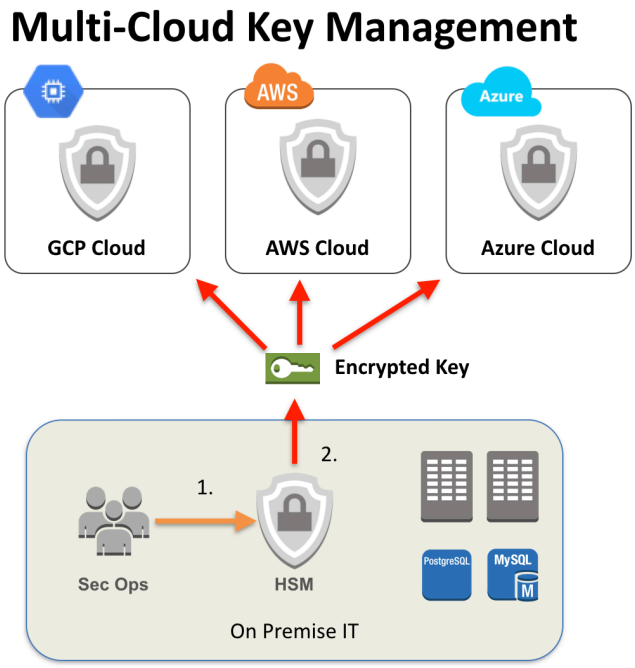
Cloud vendors often use transparent columnar or disk encryption, with a customer encryption key derived from the customer's master key. This encryption key is supplied to memory-resident encryption services so the customer master key never leaves the HSMs. The number and type of data encryption keys each cloud service creates per customer varies, but one or more keys per data repository or user role is common. These in-memory encryption engines are protected from other services, with access tightly restricted; sensitive memory such as key storage is overwritten as soon as it is no longer needed.

This graphic outlines what happens when when a user makes a request to a cloud service. (1) The user is first authenticated to the cloud. (2) After authentication the cloud service requests access on the user's behalf from the key service. (3) The key management service requests a key from the HSM on the user's behalf. (4) Assuming credentials validate and the request is authorized, the HSM returns the requested key to the key management service. What happens at this point is cloud specific. If the cloud service uses columnar data encryption, the application or database decrypts data prior to presentation to the user. If the cloud service uses volume encryption, the key is supplied when the disk is mounted, so the storage service will decrypt disk blocks for all subsequent access as content is read from physical media. Either way, these operations are transparent to users.



Multi-Cloud HSM

With these encryption and key management capabilities it does not much matter whether the customer's application architecture uses a private cloud, a public cloud, a hybrid cloud, or is distributed across multiple clouds — the framework remains the same. You can leverage various cloud services and ensure *you* control data access using your own keys to encrypt and decrypt, allowing only your authorized users to use your keys. For example you could simultaneously run analytics on GCP, SharePoint on Azure, and web applications on AWS — each using data secured by *your* keys, controlled by your on-premises HSM cluster. This is the idea behind multi-cloud key management: leverage the agility and cost-effectiveness of cloud computing while protecting data from cloud administrators, hostile nation-states, law enforcement, and unauthorized third parties. It extends your existing on-premises security controls, and is one of the few cases where on-premises security controls are fully suitable for cloud services.



BYO Key Manager

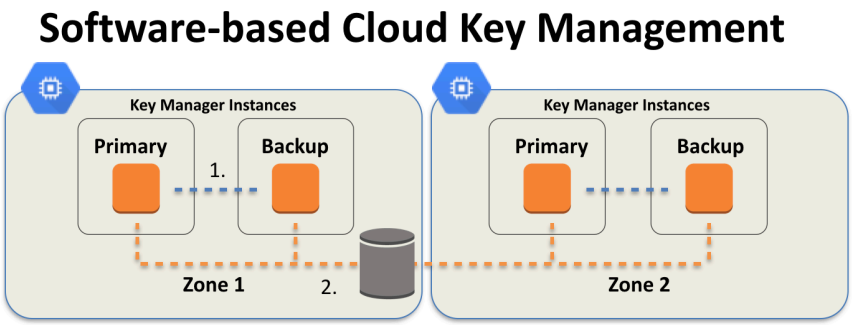
The other option for multi-cloud key management is to install your own key management server in the cloud. The organization is exactly like the illustration above, but purely in software. For customers who do not use advanced hardware for key management on-premises, but want to ensure their cloud providers do not own — and cannot be compelled to turn over — keys to decrypt their data, software-based key management is suitable. If you are already comfortable with software-based key management, bringing your own software to the cloud may be your first choice. Let's take a look at deployment.

Software Key Management

Software-based key management simply means running your own key management application in the cloud. This approach is only suitable for IaaS, as you need to install and configure your own servers to perform key management, but it provides most of the same functions as HSMs, if not the same assurances or security level. These core functions are key generation, key storage, key rotation, and API interfaces to orchestrate encryption in the cloud.

One downside is that you need to handle failover and replication yourself. Nor is the software model compliant with regulatory requirements which specify FIPS-certified hardware. But benefits include lower cost than HSMs and full control of key services, rather than delegating them to your cloud provider. You can import your own master keys, but more often customers generate *new* master keys for each cloud installation. Unique keys are provided to each cloud service by the key manager through API, so volume storage and databases can encrypt and decrypt as needed.

This graphic shows how you can deploy server images and containers within an IaaS region. Typically you have a primary server in one zone or region, with a backup in another. Replication and synchronization details are generally specific to the software you are using, but opening up a software-defined network connection between two servers is a common way for them to communicate.



Another is a shared persistent store, such as a database or "file bucket" with highly restricted access and encryption on the contents, allowing each key server instance to pull updates periodically, and ensuring only the other key manager instance can read. High availability and resilience are provided as a combination of the built in capabilities of the key management software and native IaaS cloud capabilities.

Secret Management

We have recently seen a new take on securing keys and other sensitive data elements in the cloud. A few organizations distribute secrets across multiple user accounts in the cloud. Like software key managers, these secrets are stored in encrypted files or databases, and access is restricted to specific users or users with a specific role assigned to their account. In this way service accounts — generic administrative accounts which may be assumed by one or more users — can access these secrets, but no one else. This is not limited to key management; encryption keys can be assigned to individual accounts, but access tokens, identity certificates, passwords, other access credentials and configuration data are more common.

The principal use case arose from ‘DevOps’ and similar approaches to automating software builds, software testing, and IT deployments. The strong role of automation and orchestration within cloud services and DevOps is driving heavy use of automated software build servers, container management systems, and other tools. These tasks need to run independently, without human intervention; but they need access, and often certificates and passwords as well. Developers and IT often put encryption keys and passwords in configuration files, where they are easily stolen. Rather than allow critically sensitive data to sit in unsecured files, these secrets are better placed in an encrypted ‘vault’. The vault contents can be accessed programmatically, and may be shared by one or more users in a group.

In IaaS clouds this deployment model looks very similar to software key management. There are typically one or more servers for users to connect to; with replication taking place via a dedicated network connection, database, or even secured file. Why not just use key management? Because it does not address most types of secrets, is less flexile, typically offers more limited API integration, and is harder to set up than a simple encrypted vault. Simplicity and suitability are the reasons “secret management” is gaining traction, despite current tools requiring considerable Do It Yourself (DIY) coding and integration. This is still an outlier today, but growing quickly as a simple way to solve real security problems for cloud development and deployment.

Selection and Migration

Cloud services are typically described as sharing responsibility for security, but the reality is that you aren't working shoulder to shoulder with your vendor. Instead you implement security using the building blocks they provide, possibly filling in gaps where they don't provide solutions. One of our central goals for this project was to show that it is possible to take control of data security, replacing embedded encryption and key management services, even when you don't control the environment. With key management you can gain as much security as your on-premises solution provides — perhaps even leveraging familiar tools — with minimal disruption to existing management processes.

That said, if you decided to Bring Your Own Keys (and select cloud HSMs), or bring your own software key management stack, you are signing on for additional set-up. And it's not always simple — the cloud variants of HSMs and software key management services are different than their on-premises counterparts. This section highlights some differences to consider when managing keys in the cloud.

Governance

Let's cut to the heart of the matter: If you need an HSM, you likely have regulatory requirements or contractual obligations driving your decisions. Many of these requirements spell out specific physical and electronic security levels, typically something like FIPS 140-2 Level 2 or Level 3. And these regulations often specify usage models — perhaps requiring periodic key rotation, segregation of administrative duties, and multi-admin requirements for select operations. Cloud vendors usually publish certifications for their HSMs, but not details. You'll likely need to dig through their documentation to understand how to manage the HSMs to satisfy operational requirements, and what interfaces its functions are available through — typically some combination of web application, command-line tool, and API.

It's one thing to have a key rotation capability, for example, but another to prove you are using it consistently and properly. Key management service administrative actions are a favorite audit item. As your HSM is now in the cloud, you need to determine how you will access the HSM logs and move them into your SIEM or compliance reporting tools.

Integration

A key question is whether it is okay for your cloud provider to perform encryption and decryption on your behalf, so long as your master keys are always kept within an HSM. Essentially, if your

requirement is that all encryption and signing operations must happen in hardware, you need to ensure your cloud vendor provides that option. Some SaaS solutions do not: You provide them keys derived from your master key, and the service performs the actual encryption without necessarily using an HSM. Some IaaS platforms let you perform bulk encryption in their HSM platform, or leverage their software service. Find out whether your potential cloud provider offers what you need.

For IaaS migration of applications and databases that encrypt data elements or columns, you may need to change the API calls to leverage the HSMs or software key management instances. Depending on how your application authenticates itself to the key management server, you may also need to change the authentication code in your application. The process of equipping volume encryption services with keys varies between cloud vendors, so your operations team should investigate how initial provisioning works.

Finally, as mentioned under governance, you will need to get log files from the HSMs or software key manager(s). Cloud logs are typically provided on-demand via API, or dumped into a storage repository where you can access raw events as needed. But HSMs are a special service with additional security controls, so you need to check with your vendor for how to access log files and what formats they offer data in.

Management

Whether using hardware or software you can count on the basic services of key creation, secure storage, rotation, and encryption. But a number of concerns pop up when moving to the cloud because things work a bit differently. One is dual-administrator functions, sometimes called 'split-key' authority. Two or more administrators must authorize certain sensitive administrative functions. For cloud-based key management you need to designate HSM operators. These operators are typically issued identity certificates and hardware tokens to authenticate to the HSM or key manager. We recommend that these certificates be stored in password managers on-premises, and the hardware tokens secured on-premises as well. We suggest you do not tie the role of HSM operator to an individual, but instead use a service account so you're not locked out of the HSM when an admin leaves the company.

You should modify existing processes to accommodate changes the cloud brings. And prior to production deployment, should practice key import and rotation to ensure there are no hiccups.

Operations

In NIST's definition of cloud computing, one of the essential characteristics — which separates it from hosting providers and on-premises virtualization — is availability on-demand and through self-service. HSMs in the cloud are new enough that it is not yet always fully self-service. You may need to work through a partially manual process to get set up and vetted before you can use it. This is normally a one-time annoyance which should not affect ongoing agility or access.

HSM services cost more than using native cloud key management. SaaS providers tend to charge a set-up fee and a flat monthly rate so costs are predictable. IaaS charges are generally based on the number of keys used, so if you expect to generate lots of keys — such as one per document — costs can skyrocket. Check to see how keys are generated, how often, and how often they are rotated, for a handle on operating costs. When comparing HSM with software key managers deployed atop IaaS, software offers lower licensing costs, but overall operational expense is on-par given additional set-up and management time software deployments require.

For disaster recovery you need to fully understand your cloud provider's failover and recovery models, and whether you need to replicate keys back to your on-premises HSMs. To provide infrastructure failover you may extend services across multiple zones, or perhaps between different geographic regions. If you need multi-region replication for greater assurances on up-time, make sure it's available from your cloud provider and investigate additional operational and set-up costs and that apply.

Some HSM services offer failover between geographic regions (as opposed to more localized availability zones within the same geographic region), some do not. If you choose *software* key management you will need to determine how the service will deploy and sync between zones or regions. This depends on both crypto vendor capabilities such as service endpoints and database replication, and also on which services your IaaS vendor offers. In any event you likely face additional set-up and configuration.

Finally, if you want to bring your own keys to the cloud, you will likely decide to leverage HSMs. It is the only supported model for SaaS, and this is where both cloud and HSM vendors have focused their efforts. If you want BYOK in conjunction with software key management for IaaS, it is still somewhat early on the evolutionary curve. Most of the features and functions exist, just as if you were running on-premises; but ease of management, deployment, and integration are just not there yet. It is workable but requires more effort.

Summary

There are many good reasons firms want — and often insist on — controlling data security through encryption, atop generally secure cloud services. When you control encryption keys you control data access. Directly managed encryption provides segregation of duties between you and your cloud vendor. It helps to prevent law enforcement and government from compelling your cloud partner to disclose your information. Encryption under your control can buttress application *and* cloud service controls, using your own data usage policies. And it provides auditors some comfort that data usage policies are being enforced.

We have reached a point where major cloud vendors can offer you control over your own encryption keys, while still providing the benefits of cloud computing. Security was the last major sticking point for companies to adopt public cloud services. Based on customer demand, *most* cloud providers who cater to enterprise customers have adopted strong encryption, key management, and HSM support. And they have designed these services to integrate with the rest of their compute, networking, and storage offerings. Not all vendor implementations ensure you are the only one who can access your keys in the cloud, so you need to verify how their systems work. But for the most part you can pick the cloud that best serves your business needs, rather than searching for a provider who offers minimum adequate security.

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com.

About the Analyst

Adrian Lane, Analyst/CTO

Adrian Lane is a Senior Security Strategist with 25 years of industry experience. He brings over a decade of C-level executive expertise to the Securosis team. Mr. Lane specializes in database security, secure application development and data security. With extensive experience as a member of the vendor community (including positions at Ingres and Oracle), in addition to time as an IT customer in the CIO role, Adrian brings a business-oriented perspective to security implementations. Prior to joining Securosis, Adrian was CTO at database security firm IPLocks, Vice President of Engineering at Touchpoint, and CTO of the secure payment and digital rights management firm Transactor/Brodia. Adrian also blogs for Dark Reading and is a regular contributor to Information Security Magazine. Mr. Lane is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University.

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services. Our services include:

- **Primary research publishing**: We publish the vast majority of our research for free through our blog, and package the research as papers that can be licensed for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements, and follow our Totally Transparent Research policy.
- **Cloud Security Project Accelerators**: Securosis Project Accelerators (SPA) are packaged consulting offerings to bring our applied research and battle-tested field experiences to your cloud deployments. These in-depth programs combine assessment, tailored workshops, and ongoing support to ensure you can secure your cloud projects better and faster. They are designed to cut months or years off your projects while integrating leading-edge cloud security practices into your existing operations.
- **Cloud Security Training:** We are the team that built the Cloud Security Alliance CCSK training class and our own Advanced Cloud Security and Applied SecDevOps program. Attend one of our public classes or bring us in for a private, customized experience.
- **Advisory services for vendors**: We offer a number of advisory services to help our vendor clients bring the right product/service to market in the right way to hit on critical market requirements. Securosis is known for telling our clients what they NEED to hear, not what they want to hear. Clients typically start with a strategy day engagement, and then can engage with us on a retainer basis for ongoing support. Services available as part of our advisory services include market and product analysis and strategy, technology roadmap guidance, competitive strategies, etc. Though keep in mind, we maintain our strict objectivity and confidentiality requirements on all engagements.
- **Custom Research, Speaking and Advisory**: Need a custom research report on a new technology or security issue? A highly-rated speaker for an internal or public security event? An outside expert for a merger or acquisition due diligence? An expert to evaluate your security strategy, identify gaps, and build a roadmap forward? These defined projects bridge the gap when you need more than a strategy day but less than a long-term consulting engagement.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors. For more information about Securosis, visit our website: <http://securosis.com/>.