“... there were no flaws in the theory of the machine. It offered all the defenses that its inventor and proponents said it did; the Germans’ reasoning was impeccable, and their confidence was, in theory, not misplaced. But it was in practice. The ways in which men used the machine undermined its defenses. The army cryptographers’ requirement that message keys be duplicated and then enciphered created relationships that vastly reduced the number of trials cryptanalysts would have to make to find the right key. Cipher clerks made up message keys, such as XXX, that were so easy to guess that the number of trials was reduced even more, and signals officers drafted stereotyped plaintexts, also easy to guess, that cut the number still further. The failure was not, as a later generation would say, in the hardware; it was a software problem.”

Contents

INTRODUCTION .................................................................................................................. 4
CRYPTOGRAPHY AND HARD SECURITY ......................................................................... 5
HSM deployment strategy ................................................................................................. 6
Dedicated HSMs and shareable HSMs ............................................................................. 6
Security World as Best Practice ....................................................................................... 7
SECURING USER KEYS IN THE AZURE CLOUD ......................................................... 9
Azure Key Vault Bring-Your-Own-Key Program ............................................................ 9
BYOK Design Goals ....................................................................................................... 10
BYOK and Thales Product Features .............................................................................. 11
Automating the Process ................................................................................................. 12
Securing Imported Keys ............................................................................................... 12
Achieving Multi-tenancy ............................................................................................... 12
Proving Key Encryption Key Authenticity ...................................................................... 13
SUMMARY ..................................................................................................................... 14
Managing cryptographic keys, the large numbers that cryptosystems utilize to feed the algorithms that convert plaintext to ciphertext, is critical to the integrity of any cryptosystem. As the German armed forces discovered to their chagrin when the Allies broke the Enigma in 1943, bad practice in the use of otherwise hardened cryptosystems can lead to the discovery of the keys that form the backbone of the system. Once compromised, the keys are useless and all data protection is lost. If cryptography is the art of making things secret, then key management is the art by which we keep them secret.

Cryptographic keys are very large numbers with special properties. The means by which keys are generated, used, stored, and retired from service is commonly called “key life cycle management”. At its most basic, keys must be fed to their associated encryption algorithms in plain text, readable by both computers and humans, in order to be useful. Like the rotor settings on the Enigma, the process or person performing encryption must be able to see the key to use it. On modern cryptosystems this means that a key must be exposed in plaintext within memory of the system that will perform the encryption. While a key is thus exposed it is vulnerable. The plaintext key can wind up in system cache, on backup stores, on swap partitions, in the hands of hackers.

A core requirement of modern electronic cryptosystems is the separation of keys and devices. The Enigma achieved this goal handily; keys were set via rotors at the time of use. If an enemy agent absconded with an Enigma, ciphertext encoded by it remained intact. The device, while a fine example of modern engineering, is useless without the key. Where keys and their resultant cipher text are mixed together, it is possible to recover both, especially where keys are managed like any other data object, by software processes, on the servers that use them.

Sophisticated modern cryptosystems utilize specialized devices called Hardware Security Modules (HSMs) to create, manage, and use keys. Microsoft Azure Key Vault chose Thales nShield product line HSMs to secure their customers’ keys. The balance of this document presents the features and general theory of Thales HSMs. We further present the methods that Microsoft chose in using Thales HSMs and make connections between product features, the Azure Key Vault application, and the security model that results. Our goal is to inform the reader on the technology and methods deployed by Microsoft in defense of its Azure users’ data in the cloud.
Cryptography and hard security

The movement from “on premises” IT infrastructure to cloud based deployments represents the high point of abstracting infrastructure from the applications and data that depend on it. We have moved over the past 20 years from a device centric view of computing through a network centric posture to a pure abstraction, the cloud. Nothing epitomizes this trend better than the Microsoft (Windows Azure) offering. The promises of the cloud include elasticity, extensibility, flexibility, and an appropriate deployment of resources given the problem set at hand.

However, ‘on-demand’ connections to people and machines, anywhere and at any time challenges the traditional reliance on perimeter security. Cloud security initiatives emphasize strong authentication, protection of intellectual property and customer privacy, protecting information however it moves or wherever it resides. These goals, of course, imply a strong and flexible cryptographic and therefore key management capability.

The appropriate use of cryptography to encrypt information, digitally sign documents and enforce digital rights is well proven and effectively unbreakable. But cryptography relies on the use of keys; failure to protect and manage these cryptographic keys risks shattering an entire layer of security and potentially the system itself. Many organizations make the mistake of relying on “soft security”, leaving keys unprotected on general purpose servers, vulnerable to attack. Wherever cryptography is used to protect sensitive data, organizations must deploy “hard security” controls to manage risk. Central to strong cryptographic security is the protection of keys within an HSM.

Cryptographers have relied on specialized devices that both protect keys and optimize the encryption process for as long as we have been encrypting data as a means of preserving secrets. Take the example of the Enigma. The device replaced a manual process that was previously dependent on people to transform plaintext into ciphertext. The modern computer owes its architecture to an English mathematician names Alan Turing, a name best known for his role in breaking the Enigma.

During the 1990s cryptography spread through the growingly critical networks of the young Internet. The standard “handshake” by which Internet based servers identify themselves (Secure Sockets Layer or SSL) is based on public/private key cryptography and is still the dominant model today. The need for specialized devices to protect keys and to accelerate the mathematically intensive operations of modern ciphers gave rise to a new family of products, the HSM. The core operative principal behind the modern HSM is very simple conceptually; create a device that is impervious to attack so that keys can be generated and used inside the device. This eliminates potential loss of keys due to exposure within server memory, a risk that is too probable for modern security requirements.
Early HSMs were basic but functional single purpose devices. They served as “black boxes” attached to servers wherein it was safe to expose keys in plaintext as required by cryptographic operations. HSM designers created microprocessor based devices and then encased them in neutral material (often called “potting”) that would repel both electronic and physical attempts to compromise key material. Given the tight constraints of memory and microprocessors of the time, HSMs tended to be slow and expensive. As the product genre evolved and hardware prices tumbled, HSMs became much more than simple key stores. As the Thales HSM demonstrates, the modern HSM provides means to assure complete control over the devices, high availability features, separation of duties, and a flexible, fluent native application program interface (API) for fine grained application control of HSM features. HSMs have truly become platforms for trusted computing operations.

**HSM DEPLOYMENT STRATEGY**

Today HSMs form the basis of best practice cryptographic security, but their role and value go far beyond simply protecting a key against physical attack. Deployed correctly, HSMs provide a basis for cryptographic security that can scale easily, improve system performance and yet be robust and flexible enough to handle the dynamics of real-world situations. Every organization has a unique set of applications with specific performance and security requirements. These tend to evolve as new opportunities, threats and technologies emerge. To handle the diverse and dynamic needs of today’s IT infrastructure, Thales provides a comprehensive family of flexible, scalable and interoperable HSMs.

Thales nShield HSMs use a common key management framework, the Security World. This means Thales nShield HSMs are completely compatible with each other, allowing them to be configured in any combination to meet an organization’s management, security and budgetary needs. Flexibility in configuration allows an organization to protect an existing investment, reconfigure and reallocate hardware devices as necessary, and easily extend the use of “hard security” to meet new business needs. Thales conducts extensive interoperability testing to ensure straightforward HSM integration with leading Web server, application server, public key infrastructure (PKI) and other third party software packages. In addition Thales supports industry standard APIs such as PKCS#11 and MS CAPI/CNG.

All Security World HSMs feature specialized cryptographic processors to perform CPU-intensive cryptographic operations, therefore offloading them from the host server and, in turn, dramatically increasing server capacity and optimizing the performance of cryptographic applications.

**DEDICATED HSMs AND SHAREABLE HSMs**

Thales’ HSMs are available in two distinct deployment configurations: dedicated, directly connected cryptographic modules, each attached to individual servers; and network-connected HSMs that can be shared by multiple servers. Thales’ dedicated HSMs (nShield Solo, Solo+, Solo XC, and nShield Edge) provide cryptographic resource to a particular host server, ensuring processing requests are offloaded from the host. Thales’ shareable nShield Connect, Connect+, and Connect XC are a highly secure, network-attached HSMs that provides a shareable cryptographic resource for multiple servers. Applications that require access to hardware protected cryptographic keys can access the nShield Connect, Connect+, or Connect XC over a secured connection.
SECURITY WORLD AS BEST PRACTICE

Thales’ Security World framework delivers cryptographic key management features that can scale, are robust and are flexible enough to handle real-world deployments. This powerful solution gives the ability to handle an unlimited number of keys and provides the functions necessary to manage keys throughout the entire key lifecycle from creation to operational use, back-up, recovery, archival and finally destruction.

Security World delivers a common set of features across all Thales HSMs:

**SECURITY**

All HSMs are designed to ensure that there is no single point of compromise within the key management environment. All cryptographic functions take place within a FIPS 140-2 level 3 and Common Criteria EAL4+ validated HSM meaning that Azure Key Vault based HSMs have been independently validated to the world’s most widely recognized benchmark for cryptographic modules. Central to HSM security is two-factor authentication for administrators and split responsibility or role separation which are supported by threshold sets of smartcards. This means “k” out of a total of “n” cards must be presented to authorize a specific cryptographic function or administrative activity significantly reducing the risk of malicious insiders and super-users within the Azure Key Vault infrastructure.

**SCALABILITY**

The fundamental requirement of any cloud service is flexibility and scalability and the Thales Security World key management framework provides unique ability to provision keys and replicate keys across multiple devices to satisfy the capacity requirements of Azure Key Vault customer. Security World is replete with functions that simplify the process of securely sharing keys among an array of disparate types of Thales HSMs joined to a specific Security World. This enables Microsoft to load and use customer keys in the HSMs appropriate to the task at hand dynamically without customer interaction.

**FINE GRAINED CONTROL OF KEY USAGE**

Every key protected by a Thales HSM has an associated Access Control List (ACL) that defines the allowable uses of that key. Authorization to use individual application keys can be tightly controlled allowing different levels of security to be assigned to individual keys in direct relation to their importance. Together these controls ensure that individual keys or groups of keys can be isolated from one another through logical separation. These flexible controls help to reinforce the individual requirements of a given security policy, allowing access to individual keys only by authorized users or servers, avoiding the need to impose rigid partitioning within an HSM.

**RESILIENCE**

Security World technology ensures that there is no single point of failure in any Thales HSM deployment. Multiple HSMs can be deployed on a single server or across the network to provide secure fail-over. If an HSM is damaged or stolen, keys can be recovered easily by initializing a new module. The Security World key management framework has a range of built-in controls to simplify back-up and recovery – all essential aspects of a robust cloud based service.

---

**nCORE**

A native API, nCore, that allows programmers to interact directly with the HSM at its most basic level. High level APIs like PKCS#11 or MS CNG are of necessity simple and consistent across platforms. The goal of a high level API is to make complex things simpler. With nCore (the language in which Thales’ PKCS#11 is written) the programmer can create functions and capabilities specific to her problem set. These functions can use the full array of cryptographic primitives available “down on the metal”. We will come back to this feature when we review how Microsoft secures its customer keys for import into the Azure cloud.

**DEFINING A FEW KEY TERMS**

The goal of this document is to inform the reader about how Microsoft uses some of the advanced features of Thales’ cryptographic products to secure user tenant keys in the Azure cloud. It is easy to lapse into technical jargon, a temptation which the author eschews on general principal. However, some key technical terms are necessary to a good grasp of the key concepts. Some key terms used in this document include:

- **HSM – Hardware Security Module.**
- **KEK – Key Encryption Key – a key used to encrypt other keys.**
- **ACL – Access Control List – a feature of a key in the Thales system that allows key owners to control how, by whom, and for what purpose a key may be used.**
- **ACS – Administrative Card Set – a set of smart cards that enable administrators to add HSMs to a Security World, change Security World parameters, and other critical administrative functions. Ownership of the core secrets of the Security World is vested in the ACS.**
- **OCS – Operator Card Set – a set of smart cards that enable fine grained control of the key loading process.**
- **nCore – the native application programmer interface (API) for the Thales HSM. This API enables low level control of the HSM and use of its more fine grained features in securing application layer data.**
- **Key Recovery – the ability to restore an encrypted key in the Security World to its original plaintext format.**
- **Key wrapping – the use of a KEK to encrypt a key for transport or storage.**
Securing User Keys in the Azure Cloud

Azure Key Vault service utilizes an important key called the tenant key at the root of its trust model. Azure Key Vault key management service (KMS) stores and uses these tenant keys with extreme security thanks to its reliance on industry proven, FIPS and Common Criteria compliant HSMs from Thales. The KMS also offers related services such as the Bring-Your-Own-Key capability that lets customers generate, import, and delegate use privileges to Microsoft for use in the Azure cloud.

**AZURE KEY VAULT BRING-YOUR-OWN-KEY PROGRAM**

Azure customers often need to use a key generated by, archived at, and under the control of their organization’s security officers. This requirement may be due to compliance, because they are migrating from their on-prem systems, or simply to achieve best practices in key custody. With the Bring-Your-Own-Key (BYOK) feature, Azure users may generate a tenant key on their premises, using tools of their choice, in compliance with their own policies. This key is securely imported into the Thales HSMs used in the Azure data center. The Azure user is promised assurance that Microsoft operators cannot see or leak the key during the import as well as during the running steady state.

Microsoft Azure Key Vault team approached Thales seeking advice and assistance in the design and implementation of a secure key import process that would accomplish several goals:

- Spare Azure customers the necessity of purchasing their own HSM and co-locating it in the Azure service data center. Customer tenant keys can be loaded into Azure HSMs and used on behalf of customers without regard to which HSM is doing the work.
- Securely import end user generated keys without exposure of key plaintext during the import process outside of the boundary of a Thales HSM.
- Automation of the import process without use of unsecure temporary disk files such as password protected PEM format files.
- Elimination to the greatest degree possible of the ability for Microsoft to recover plaintext copies of customer keys.
- Proof acceptable to Azure users that the transport key, the KEK, did indeed come from Microsoft and was indeed created in a Thales HSM.
Accomplishing these goals required investment on the part of Microsoft in developing custom nCore applications and some creative use of Thales’ Security World features. Thales Advanced Solutions Group (ASG) provided training, design review, architectural review, and coding assistance to Azure Key Vault developers as they set about using Thales’ extensive tool kit to design and implement one of the first cloud based HSM offerings to achieve customer ownership of keys, multi-tenancy within the Azure data center, and control of key usage rights by key owners rather than vendors. In addition, Azure customers can use standard digital certificates and their chains of trust to prove that their keys are indeed the ones used by Microsoft on their behalf.

**BYOK Design Goals**

Microsoft set out to create mechanisms that would enable BYOK customers to import keys generated at their sites while offering assurances that customer generated key material cannot readily be recovered, read, or exported by Microsoft Azure hosting services personnel. In some cases, Microsoft may generate and maintain a private key on behalf of its customer. In these cases, Microsoft is a trusted custodian of the private key. In other cases, the BYOK customer will generate her own key and maintain the key for long term key recovery purposes. This customer may generate the key on premise within an nShield HSM or may generate the key in some third party system.

Furthermore, Microsoft required that the importation of copies of customer generated keys be an automated process that allows Azure customers to operate independently to transfer keys over an untrusted network. Azure BYOK customers can take advantage of Thales low cost USB connected HSM, the nShield Edge, to generate their tenant key in their own Security World thus assuring that these critical keys are never exposed outside of compatible Thales HSMs. In these cases Azure customers can assure that custody of the tenant keys is maintained according to the strictest industry best practices without breaking the budget.

Microsoft wanted to be able to give assurances that no one at Microsoft could ever get plaintext access to the keys, either directly or indirectly. Once introduced by the Azure BYOK user in the HSM’s security world, the key would remain there and no artifacts that could enable the key to be extracted, exported or shared are left in the hands of Microsoft. Microsoft also wanted to establish a key upload process that was verifiably secure and the keys always remain opaque to Microsoft throughout the process. What’s more, Microsoft not only wanted to make sure they didn’t leak or share the keys, Microsoft wanted, to the fullest extent possible, to be able to give assurances that they couldn’t do so due to the architecture implemented.
BYOK AND THALES PRODUCT FEATURES

In a Thales Security World, the root of trust belongs to the customer who owns the Security World. This ownership is instantiated technically in several system objects:

- The Security World master keys (KMSW, KNSO, etc.) – AES 256 keys randomly generated by the HSM upon creation of the Security World.
- The World File – a disk file that contains strongly encrypted key tokens for all of the Security World infrastructure keys; (i.e. KMSW, KNSO, etc.).
- The Administrative Card Set – a set of smart cards that contains the tokens that allow an nShield to load the infrastructure keys. The ACS tokens are split with a “k” of “n” scheme.

In the early stages of design, Microsoft used standard product features to accomplish simple key importation. This meant that early Rights Management Services (RMS) users had to:

- Assemble a quorum of ACS holders in Redmond.
- Initialize a Microsoft Thales HSM with their master keys from their ACS.
- Set the HSM up as a member of their Security World.
- Change permissions on the keys to be delegated to prevent Microsoft from recovering them.

Once their keys were imported into the Azure Security World, users would have to initialize the HSM used during the transfer to remove any remnants of the Security World and trust Microsoft to be careful with their delegated keys.

This process had obvious limitations:

- Users had to bring a quorum of card holders to Redmond. This is expensive and card holders tend to be difficult to bring together at the same time, much less in a remote location.
- The process by which Microsoft changed the ACLs on customer tenant keys was manual and, during the process, a temp file containing recovery data was created. Microsoft process administrators had to assure customers that this temp file was destroyed during the process.
- The process required detailed understanding of HSMs and cryptography. This limited the appeal of BYOK to a subset of technically sophisticated Azure users.
AUTOMATING THE PROCESS

The first task undertaken by Microsoft and Thales involved development of an automated tool that would:

- Replace the manual steps of the onsite process with an automated process that can be executed by Azure customers at their premises so that customer-generated tenant keys can be automatically uploaded to Azure cloud without the inconvenience of a trip to Redmond.
- Transform the ACL of the imported key so that Microsoft cannot recover the customer tenant key in plain text. This maintains the integrity of the customer held tenant key while enabling Microsoft to securely use a copy of the key on behalf of the customer.
- Utilize the warrants feature of Thales HSMs to prove to its customers that the key encryption key was generated within a Thales HSM that is owned by Microsoft.

The Azure Security World issues a key encryption key. The KEK is forwarded to the Azure user’s Security World. The customer runs the Microsoft utility which changes the ACL, deletes any recovery data, encrypts the tenant key with the KEK, then sends the wrapped data objects over to the Azure Security World.

SECURING IMPORTED KEYS

Upon receipt, the Azure Security World imports the encrypted tenant key and unwraps it. Recovery has been turned off. The ACLs were set to eliminate Microsoft’s ability to recover the keys in their plaintext forms. The ACLs permit the keys to be loaded into Azure based HSMs, used on behalf of the customer within Azure, but not exported or recovered.

This achieves the goal of preventing Microsoft from intentionally or accidentally “losing” a plaintext key copy. The key is only exposed within the HSM. Microsoft cannot recover key plaintext; it is prevented from doing so by the new ACL. Only the customer, the key owner, can do so. Microsoft cannot “expand the ACL” (a process by which ACLs can be changed); it is not the “owner” of the key plaintext. No plaintext copy of the key is ever presented on any Microsoft servers so accidental “loss” of a copy is a 0% probability risk.

ACHIEVING MULTI-TENANCY

A standard feature of Security World is its extensibility and the separation of keys from devices. In a Security World, the owner can deploy as many HSMs as she needs to meet processing demand. Likewise, any Security World key may be loaded into any or all HSMs within the Security World. The crypto that secures this key sharing across an unknown and unlimited number of HSMs is built into Security World and is invisible to the administrator. Security World provides all of the commands and key management methods needed to securely load keys over all or some portion of the Azure HSM estate.

Once the imported tenant keys are encrypted, watermarked, and signed (rendering them confidential, integrity protected, and authenticated) by Microsoft’s Security World they can be loaded into any HSM in the Azure pool. The ACLs set by the import process that limit the ability of Microsoft to recover their plaintext apply to all HSMs in the Security World. Thus it does not matter which HSMs process any given transaction on behalf of the Azure cloud user. The limitations imposed by the new ACLs apply to all. Any of them will do. The promise of “HSM capacity on demand” has become reality.
PROVING KEY ENCRYPTION KEY AUTHENTICITY

Every Thales HSM contains its own key pair used to identify itself. In general, this feature is not customer visible; it is used by Thales and a few customers/third party vendors to prove that keys generated in an HSM were indeed generated in a specific HSM. Thales maintains a database of certificates and HSM electronic serial numbers (ESNs) that associate HSMs and their certificates.

These certificates are known internally within Thales as “warrants” as they are used to warrant that a specific object was created by a specific Thales HSM. Simply put, a Thales HSM can sign its application key tokens. Another Thales HSM or a software process using nCore can validate these signatures much as digital signatures are used to sign contracts on line.

Validation of the signatures is accomplished via standard signature validation routines. Microsoft will host the certificates for each HSM that it deploys in Azure. Microsoft can present the HSM warrant when it ships its key encryption key by which the customer will wrap and secure their tenant key. Microsoft’s customer can validate by means of the warrant that the KEK came from Microsoft and that it was generated in a Thales HSM. This provides customers with cryptographic evidence that Microsoft’s assertions are indeed valid and true.
Summary

Until now, the promise of “crypto as a service” or “crypto on demand” via the cloud has remained elusive. Some have placed Web services in front of crypto enabled back end systems (sometimes called “single purpose private clouds”) in order to share the dedicated processing of hardware enabled crypto systems. These are variations of the on premises theme and, while useful and creative, cannot be characterized as “public clouds”. The Microsoft Azure offering clearly breaks new ground in its use of hardware cryptography to secure a public cloud application.

Microsoft began development of the Azure cloud convinced that hardware key management was a baseline service that customers would demand, and deserved. Over the course of the planning process for the Azure cloud offering, Microsoft and Thales traded architectures and concepts in a joint effort to design a deployment that met the core requirements of a proper security model:

- Tenant keys must be owned by the Azure customer, not Microsoft. This philosophy underpins the Azure BYOK program. While there is nothing inherently wrong with trusting your cloud vendor to generate and manage keys on your behalf, you sacrifice essential control over your data and make it virtually impossible to switch vendors.
- Keys must be impervious to loss during the process of copying them from customer premises to the Azure cloud. Microsoft designed its BYOK process so that it performs HSM to HSM key importation. This eliminates the risk that intermediate file types such as PEM files may be lost, misused, or otherwise compromised.
- Plaintext keys must not be accessible to Microsoft data center personnel. The use of restrictive ACLs limit the ability of Microsoft to expand (change) the ACL, to recover plaintext keys, or to export keys in plaintext intermediate formats. Microsoft can use keys on behalf of their customers but cannot read them, back them up in plaintext, or share them in plaintext with anyone including government agencies.
- Azure Key Vault customers must be able to verify that the key encryption key provided by Microsoft to secure the upload of their tenant key did indeed originate from Microsoft and was indeed generated in a Thales HSM. The warrants (digital certificates) that identify every Thales nShield HSM can be deployed to enable customers to validate this chain of trust using a utility supplied by Microsoft and access to the Thales root supplied by Thales.
- HSMs should be deployed as shareable resources. Customers should not have to purchase entire HSMs or rigid partitions, in effect solidifying a hardware resource before it is needed. The combination of custom ACLs, warrants, and standard Security World utilities make it simple to load customer keys onto any Azure Key Vault HSM or combination of HSMs in order to meet customer processing needs. The keys are each signed by their owners and restricted in their use by the custom ACLs.
When the German armed forces chose the Enigma from a population of competing electrical cipher machines they chose an example of the best that technology could offer at the time. The Enigma offered modern architectural features such as separation of keys and machinery. The combination of rotors and plug settings enabled a mathematically vast set of combinations that could defy even the most dedicated teams of cryptanalysts. The primary weakness of the Enigma systems lay not in the equipment but in its application. The habits and methods employed by the German military led inexorably to the final crack of the Enigma naval cipher in February, 1943. This loss of secrecy was one of the crucial elements in the loss of the battle of the Atlantic by the German navy.

In creating the Azure Key Vault offering, Microsoft chose Thales HSMs for their power, flexibility, and expressive API set. The twin tools of the Security World and nCore API allowed Microsoft to develop methods that epitomize cryptographic best practices. Upon starting with a sound product that meets the basic requirements for hardware key management, Microsoft partnered with Thales to develop software and process methods that utilize the strength of the underlying hardware platform in secure ways, methods the epitomize best practices.

Thales and Microsoft jointly explored the boundaries of Azure use cases seeking out threats that could threaten Azure users’ interests and means of mitigation. In all cases, trust in the cloud must start with trust in the integrity and intentions of the cloud provider. Microsoft has for a generation embodied trust and caution in technology deployment. Once established, that trust must permeate the systems deployed echo from the end user interface back though the cloud vendor’s key management practices.

During the run up to peace talks President Ronald Reagan was taught an old Russian proverb; Doveryai no Proveryai (trust but verify). The President recited his lines during the talks like the professional that he was and relations between the United States and the newly formed Russian Federation were cemented. Microsoft offers Azure users a set of tools, processes, and capabilities that will enable them to use best cryptographic practices to generate their own keys, export them securely to Azure Key Vault, achieve a high degree of confidence that the keys will be used on their behalf without the ability on the part of Microsoft to render them in plaintext no matter who asks them to, and to verify for themselves that Microsoft has indeed done what it claims. This represents a sound basis for confidence in the Azure RMS offering, among the best on offer in today’s market.

You can contact Thales at: msbyok-hsm@thalesesec.com
About Thales e-Security

Thales e-Security + Vormetric have combined to form the leading global data protection and digital trust management company. Together, we enable companies to compete confidently and quickly by securing data at-rest, in-motion, and in-use to effectively deliver secure and compliant solutions with the highest levels of management, speed and trust across physical, virtual, and cloud environments. By deploying our leading solutions and services, targeted attacks are thwarted and sensitive data risk exposure is reduced with the least business disruption and at the lowest life cycle cost. Thales e-Security and Vormetric are part of Thales Group. www.thales-esecurity.com.