Protecting Data in Microsoft Azure

Microsoft

Abstract

Microsoft® is committed to ensuring that your data remains your data, without exception. When stored in Microsoft Azure, data benefits from multiple layers of security and governance technologies, operational practices, and compliance policies in order to enforce data privacy and integrity at a very granular level. This white paper describes such capabilities in Microsoft Azure, including mechanisms for encryption, secrets administration, and access control that you can leverage for managing sensitive data. The sections that follow provide detailed guidance on how to use features in the Microsoft Azure platform to protect critical enterprise data in the cloud, whether structured or unstructured, in-transit, or at-rest.

Audience

This document focuses on data protection in Microsoft Azure, and is intended for Information Technology (IT) Professionals and IT Implementers who deal with information asset management on a daily basis, either as their main duties or as part of a broader cloud IT management role. This document will be most useful to individuals who are already familiar with Microsoft Azure, and are looking to increase their knowledge of tools and technologies for encryption, access control, and other aspects of data security in the platform and related services. Sections 2.1, 2.2, and 2.3 provide a brief overview of Azure and can be skipped depending on your existing knowledge of Azure services.

NOTE: Certain recommendations contained herein may result in increased data, network, or compute resource usage, and increase your license or subscription costs.

Published August 2014

Table of Contents

1 Overview

There are multiple tools within Microsoft Azure to safeguard data according to your company's security and compliance needs. One of the keys to data protection in the cloud is accounting for the possible states in which your data may occur, and what controls are available for that state. Specifically:

- **At-rest**: This includes all information storage objects, containers, and types that exist statically on physical media, be it magnetic or optical disk.
- **In-Transit**: When data is being transferred between components, locations or programs, such as over the network, across a service bus (from on-premises to cloud and vice-versa, including hybrid connections such as ExpressRoute), or during an input/output process, it is thought of as being in-motion. Being in-transit does not necessarily mean a communications process with a component outside of your cloud service; it moves internally, also, such as between two virtual networks.
- **In-use**: (or in-process) Dynamic data usage could be a table kept in virtual memory, transactions in a message queue, or even encryption keys in the CPU cache. Information being acted upon in some way by the host or guest during a process, such as real-time database queries running in active memory (as opposed to a page file sent out to disk), could be in different security states depending on whether it is encrypted or decrypted, and the security context of the operator.

Further, there are two (2) fundamental types of data at rest:

1. **Data in production.** There is data in some form of storage, e.g. Azure SQL Database, and compute processes that need to access that storage during production operations. In this case, encryption at rest is aimed at protecting the data in that storage (whereas the compute aspect deals with data in use).
2. **Data not in production.** There is data in some form of storage, e.g. a Virtual Hard Disk (VHD), but that VHD is not in production use. For example, it may be part of an upgrade operation, but the VHD has not yet been loaded or mounted. Data encryption at rest is applicable here, but the compute aspect is not relevant for this scenario.

As a result, you need to weigh the cost of protection in terms of compute cycles for cryptography, application performance, resource latency, management overhead, content classification and filtering, and rights management. If your approach includes redundancy to prevent data loss, then extra storage capacity, region-specific presence or geo-redundancy may also be necessary, further impacting costs.

In the sections that follow, we will discuss how and where information can be most effectively protected through encryption, access control, and other methods. In many cases, requirements are dictated by your organization's need for data governance and compliance efforts. Industry and government regulations such as HIPAA and FedRAMP, and international standards such as ISO 27001, lay out specific safeguards through processes and policies. It is a shared responsibility between Microsoft Azure and its customers to implement sufficient mechanisms to meet those obligations. Specifically, Microsoft provides a compliant platform for services, applications, and data, while Azure customers must design and configure their cloud environment to ensure the confidentiality and integrity of their information assets.

2 Data Storage in Microsoft Azure

When it comes to protecting data storage (that is, held in a container other than temporary storage or active process memory), it is possible to keep data in three (3) major areas within Microsoft Azure, as shown in Figure 3. Each technology has a unique security model designed to handle its specific data type(s):

- Azure Storage—persistent structured and unstructured cloud storage
- SQL Database—fully managed relational database service
- Azure Active Directory—identity and access management

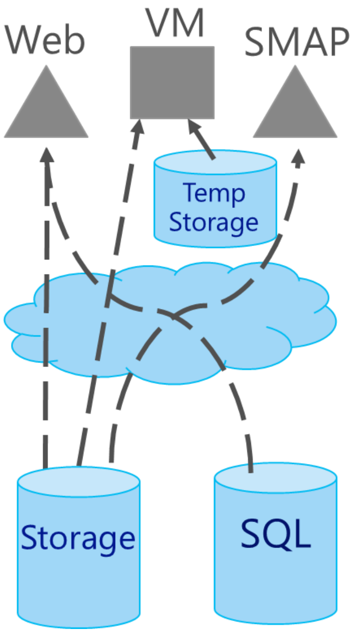

Figure 3: Azure data storage.

Both Virtual Machines (a.k.a. Infrastructure as a Service, or IaaS) and Cloud Services (a.k.a., Platform as a Service, or PaaS) have the concept of temporary disk space. In the case of a Cloud Service, the entire role instance should be treated as non-persistent. Any data which isn't written explicitly to Azure Storage could be lost at any time.

The Virtual Machine (VM) temporary disk is a physical disk on the node that can be used for scratch space (such as an OS pagefile). Data on this disk is not persistent, and will be lost any time the VM is moved to another physical machine, during patches, or when Azure detects a problem with your host node.

Virtual Machines has a similar concept, but unlike Cloud Services, the primary disk is persistent (as it is used to host the operating system and whatever else the customer deploys). Temporary drives are also provided to IaaS customers. These drives will often perform faster since they reside on the local host. The temporary drive is usually exposed as the D: drive in Windows or /dev/sdb1 in Linux.

Virtual Hard Drives (VHDs) are a category of container for disks and disk images (templates). VMs use VHDs which are mounted off of Azure Blob Storage. Cloud Service role instances (a virtual machine on which the application code and role configuration run) use VHDs that only exist on the local host on which they are running, and each time a role is de-provisioned from a host, that VHD is deleted. When a new role is started, it gets a fresh copy of the base image (thus, Cloud Service roles are not persistent).

Lastly, be mindful of the numerous other services and containers that can contain sensitive information:

- Key repositories (in the cloud, on-premises, with partners, in a Key Management Service (KMS));
- Logs and reports (Azure Storage, SQL Database, core services such as Azure Active Directory);
- Custom cloud applications and services (offerings such as HDInsight and BizTalk Services);
- Customer-selected third-party cloud services or data processors (e.g., EDI partners);
- Customer's business partner or customer cloud environments (or on-premises private clouds), such as third-party Security Information and Event Management (SIEM) systems that store and analyze security logs;
- Data that you offload or export, including Virtual Hard Drives, databases and backups.

2.1 Microsoft Azure Storage

Azure Storage is the repository for running VMs, VHDs, configurations, and customer data. It is built on a "log-based file system", meaning anytime that anything is written to Azure Storage (whether in a blob, table, or queue), it never overwrites an existing value on a disk. Instead, all writes (regardless of what object is being written) are written into a circular queue which is flushed to physical disks. This provides extra assurance against corruption as no transaction is ever finalized until the new data is in place. All data destined for Azure Storage is broken into chunks and copied across multiple physical disks (and at different geographic locations, if geo-replication is enabled).

- **Azure Table Storage** is a distributed (auto-partitioning and load-balancing) solution for storing loosely structured data. It does not contain a data engine like a typical relational database. So, while it is simple to persist data, if sophisticated queries, complex indexing, and manipulation are required, then it is more appropriate to use Microsoft Azure SQL Database (or SQL Server running in a VM).
- **Azure Blob Storage** (Binary Large Objects) can be used to store any type of text or binary file data such as a document, media file, application installer, or drive (e.g., a VHD). Blob Storage doesn't enable the same sort of structure or querying ability that Table Storage does, so it is best used for storing large chunks of unstructured data.
- **Azure Queues** provide reliable first-in-first-out (FIFO) messaging. This is often used for workflow processing and for communication between components of cloud services. Complicated pipelines can make use of Azure Service Bus which enables more flexible and configurable queue-like functionality.

Permanent data in Azure Storage and SQL Databases is replicated and stored redundantly to enhance availability, while temporary data stored in Azure Virtual Machines may be lost if there is a hardware, system or application failure.

Azure Storage nodes are all independent from Azure Compute machines. They are deployed on separate hardware, each with its own management and security model. Specifically, Azure Storage executes access control policies, and all storage requests must be authenticated. Authentication relies on a Bearer Token model. Any user or system possessing the correct token (key) is granted access to the data. Storage Keys should be considered highly privileged as they grant unrestricted access. One exception is that blobs can be configured to support anonymous authentication.

Another type of token supported by Azure Storage is known as a Shared Access Signature (SAS) which can be appended to a URL that enables delegated and scoped access to a container, blob, table, or queue. Access can be locked down to a selected activity: read, write, append, en-queue, de-queue, etc. (depending on the storage type) and scoped to specific time windows.

2.1.1 Data Structures

Microsoft Azure Storage organizes every blob into a container, which enables assigning security policies to groups of objects. Every Azure subscription can have multiple storage accounts: a storage account can hold any number of containers, and a container can hold any number of blobs, up to the capacity limit of the storage account. Azure Storage offers two (2) types of blobs:

- **Block blobs**—optimized for streaming and storing cloud objects, such as documents, media files, and backups
- **Page blobs**—optimized for representing Virtual Machine VHDs or Cloud Services drives and supporting random writes

2.1.2 Data Transfers

Moving data from your on-premises datacenter into Azure Storage over an Internet connection may not always be feasible due to data volume, bandwidth availability, or other considerations. The Azure Storage Import/Export Service provides a hardware-based option for placing/retrieving large volumes of data in Blob storage. It allows you to send BitLocker-encrypted hard disk drives directly to an Azure datacenter where cloud operators will upload the contents to your storage account, or they can download your Azure data to your drives to return to you. Only encrypted disks are accepted for this process (using a BitLocker key generated by the service itself during the job setup). The BitLocker key is provided to Azure separately, thus providing out of band key sharing.

2.2 Microsoft Azure SQL Database

For structured or transactional data, you have the options of either deploying instances of full SQL Server in an Azure VM (IaaS) or using Azure SQL Database (PaaS). A SQL Server VM in Azure is identical to what you would run virtualized in your own on-premises datacenter. It can even be joined to your domain to make it easier to develop hybrid applications that can span both on-premises and the cloud under a single corporate trust boundary.

Azure SQL Database, while very similar to SQL Server, has different design goals which result in functional differences when compared to SQL Server in a VM:

- SQL Server running in a VM is optimized for the best compatibility with existing applications and for hybrid applications. It provides full SQL Server box product features and gives the administrator full control over a dedicated SQL Server instance and cloud-based VM.
- SQL Database is optimized to provide a very quick and easy way to build a scale-out data tier in the cloud, while lowering ongoing administration costs since customers do not have to provision or maintain any Virtual Machines or database software.

Thus, beyond the operational characteristics of the database, such as the level of scale and availability you require, there are differences in your level of configuration and management between SQL Server in a VM versus SQL Database. When you deploy a VM, you are responsible for every aspect of its configuration and security (refer to the SQL Server and Azure SQL Database TDE section later in this paper). However, with Azure SQL Database, Microsoft provides configuration, upgrades, and patching of the platform.

Hybrid applications, where functionality operates on both sides of the Internet (i.e., a multi-tier application could have components running on-premises and in Azure simultaneously, with sensitive data stored locally by the customer with replication to the cloud), will benefit from protection implemented at the transport, protocol, and authentication/authorization layers.

2.2.1 Azure SQL Database Structure

Azure SQL Database is a PaaS relational database offering. Customers are granted access to their databases through standard interfaces while administration of the underlying system is managed by the Azure platform and Microsoft. Each physical machine can have hundreds (or more) of user databases. Customer databases exist in unique logical server instances implemented in individual containers. The effect of this is that customers and their data are strictly segmented from each other while machine capacity is optimized (thus becoming a more economical solution).

2.3 Microsoft Azure Active Directory

Functionally, Azure AD is the extension of Windows Server Active Directory into the cloud, designed to meet the needs of multi-tenant cloud services and applications. Like Windows Server Active Directory Domain Services (AD DS), Azure AD is an identity repository and engine that provides authentication, authorization, and access control for an organizations' users, groups, and objects. As a cloud service, Azure AD benefits from high scalability and availability, supporting more than thirty million active users and more than eight-hundred thousand active tenants simultaneously.

2.3.1 Azure Active Directory Architecture

Internally, the structure of Azure AD is much like that of AD DS—a hierarchical and highly-relational data store of objects and rules defined by schemas. Tenants are assigned to containers within replicated directory store partitions and each partition is assigned to a region. This can be seen in Figure 5 below. To safeguard against data loss, writes are then committed locally to a replica in a separate datacenter before returning to the caller. Replication between regions is controlled based on regional compliance requirements.

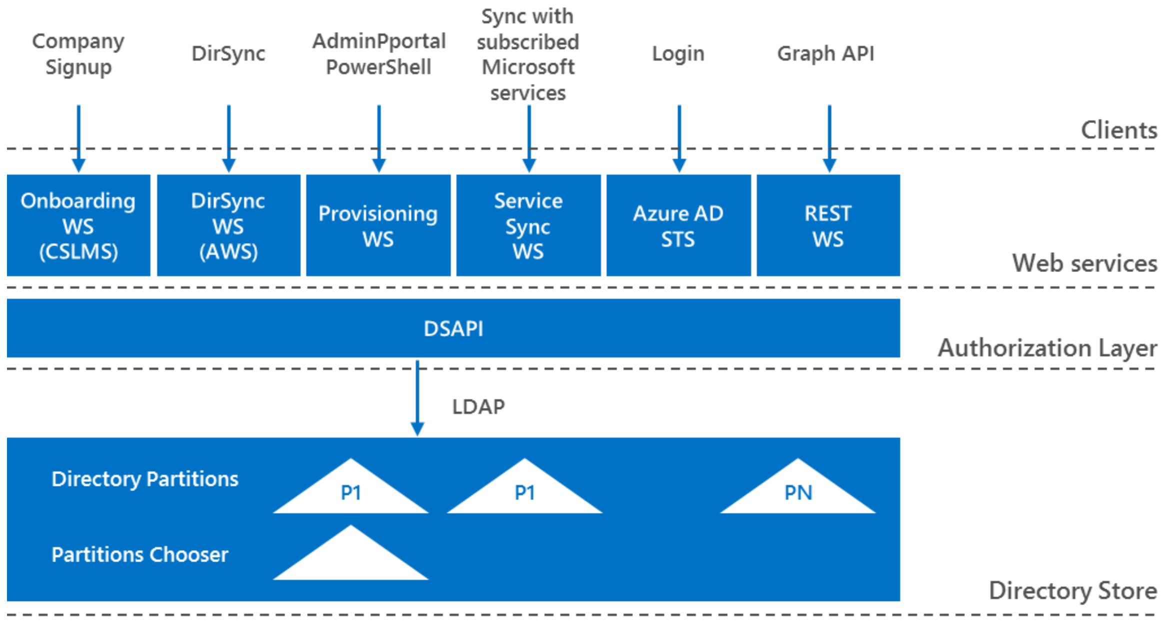

Figure 5: Architectural overview of Azure Active Directory.

Each organization (Azure AD tenant) is logically isolated using security boundaries so that no customer can access or compromise co-tenants, either maliciously or accidentally. Azure AD runs on “bare metal” servers isolated on a segregated network segment, where host-level packet filtering and Windows Firewall block unwanted connections and traffic.

2.4 Who Can Access Your Data?

All Microsoft Azure Storage types enforce data access procedures, yet each has its own data access and control model. This provides the ability to maintain control over the use and storage of data using several different available mechanisms detailed below.

Each Windows Azure subscription can create one or more Storage Accounts. Each Storage Account has a primary key (Storage Account Key, or SAK) and secondary secret key (the Shared Access Signature, or SAS) that is used to control access to all data in that Storage Account. The rights of the subscription owner allows a user to retrieve or change the access key. Within a subscription, administrators can also create and destroy storage accounts (along with other resources). This supports the typical scenario where storage is associated with applications (which are in possession of the storage key) and those applications have full control over their associated data. Storage accounts can be accessed by Azure services or by on-premises applications with the appropriate credentials. Table 3 below outlines typical access patterns to different classes of requestors.

Customers	Subscription	Authentication Type
Developers and Operators	Microsoft Azure Portal / SMAPI	Federated Identity / Managed Identity (Microsoft Azure Portal) Self-signed certificate (SMAPI) Azure AD with customer-supported two-factor authentication [e.g., Azure Multi-Factor Authentication (MFA)]
Role Instances	Storage	SAK
External Applications	Storage	SAK
Azure SQL Databases	Storage	Username/password, Connection Strings

Table 3: Supported authentication types for customer data and applications in Azure.

2.4.1 Single Sign-On (SSO)

Administrators can federate on-premises AD or other directory stores with Azure AD. Microsoft recommends that organizations synch their on-premises directory information with SSO to ensure that users removed from on-premises directories are also removed from the Azure AD to maintain sufficient access controls. Deployment of a highly available Security Token Service (STS) (Active Directory Federation Services 2.0, or AD FS, for example) on-premises is required as this STS will become part of the authentication flow for every user login. Once federation is configured, all Azure AD users whose identities are based on the federated domain can use their existing corporate logon to authenticate to Azure AD services. Federation enables secure, token-based authentication and SSO across Azure applications.

2.4.2 Two-Factor Authentication (2FA)

Microsoft provides Multi-Factor Authentication for Azure administrators with a phone as the second factor; it also supports integration with third-party authentication solutions via on-premises STS integration.

2.4.3 Access Controls: Subscriptions

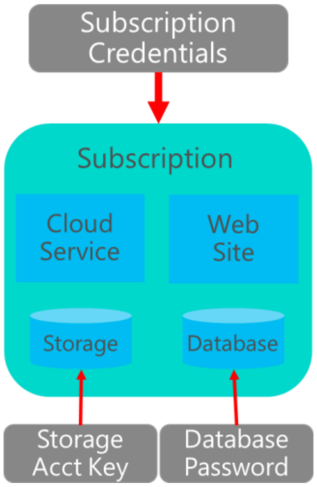

Figure 6: Subscription access.

All data in Azure, no matter the type or storage location, is associated with a subscription, as shown in Figure 6. Customers may have multiple subscriptions, and multiple deployments/tenants within each one, but the account used to create and manage the subscription has full rights over any data stored in it.

Authentication to the Azure Portal is performed via Microsoft account (formerly LiveID) or through Azure AD Authentication using an identity created either in Azure AD or federated with an on-premises directory.

Access to VMs by default occurs using credentials locally stored on that VM (usually the user name and password, although Linux allows the option for mutual certificate authentication through SSH). Tenants may elect to follow a different authentication scheme such as joining VMs to an on-premises Active Directory domain.

2.4.4 Access Controls: Storage

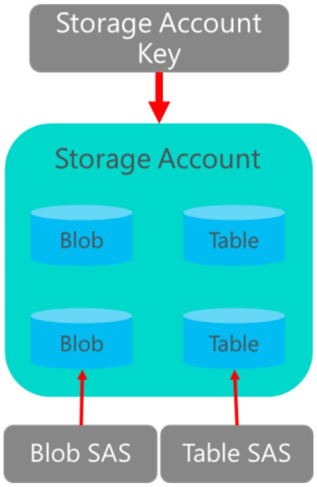

Figure 7: Storage account access.

Access to Azure Storage data (including Tables) can be controlled through a SAS token, which grants scoped access. The SAS is created through a query template (URL), signed with the SAK. That signed URL can be given to another process (i.e., delegated), which can then fill in the details of the query and make the request of the storage service. A SAS enables you to grant time-based access to clients without revealing the storage account's secret key. This key flow is shown in Figure 7.

2.4.5 Access Controls: Azure Tables vs. SQL Databases

While Azure Tables support URL-based access with a Storage Access Signature, SQL Database requires a connection string with user name and password to establish a connection with the database, and continues to use the same access control model used in traditional SQL Server databases. (With a SQL Server VMs, you can also authenticate via Kerberos tokens if the VM is domain-joined.)

You must ensure that security credentials (for example, the Hash-based Message Authentication Code (HMAC) key or SQL user name and password) are covered by appropriate protection measures such as encryption (discussed later in this document). Table 2 below outlines SQL DB and Azure Table access.

COMPARISON CRITERIA	AZURE TABLE STORAGE	SQL DATABASE
AUTHENTICATION	**Shared Access Signature, Storage Account Key** 512-bit HMAC key is used to authenticate users	**SQL Authentication** Standard SQL Authentication is used to authenticate users
ROLE-BASED ACCESS	**Yes** Read, Read/Write, Append	**Yes** Standard SQL database and application roles

Table 2: Supported access and authentication mechanisms in Azure SQL Database and Table Storage.

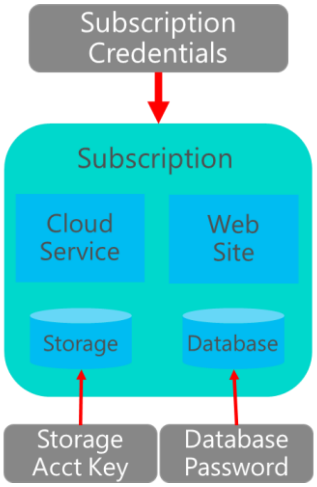

Figure 8: Azure SQL Database access.

An Azure SQL Database instance uses an access security model very similar to that of SQL Server, as shown in Figure 8. Based on subscription access, you can create usernames and accounts with full access rights, and with administrator access, you can create other usernames and passwords with granular access to data (read, write, read-write). Within a subscription, administrators can also create and destroy SQL Database instances.

Azure SQL Database supports the Tabular Data Stream (TDS) protocol, which means that the same data access technologies (e.g. ADO.Net, Entity Frameworks) used in traditional databases can be used to query data. Attention should be paid to Internet-facing input ports, as this can be a security concern when opening 1433 to the Internet. The Azure SQL Database Firewall blocks all access to your SQL Database instance at the network level until you specify which computers have permission, and grants access based on the originating IP address of each request.

3 Understanding Data Security

Many IT professionals immediately think of encryption when data security is raised. Although important, there's more to data security than just encryption—in fact, "safety" is perhaps a better word, as it entails the broader perspective of securing against loss, damage, and misuse, and encourages business process improvements to maintain data integrity.

3.1 Risk and Risk Management

Data should be considered "at risk" no matter where it is or how it is being used. In the cloud, this is largely because rather than data being sequestered in your datacenter on your physical server racks, it is in shared storage using (public) Internet endpoints. But the degree of that risk is dependent on the type of workload and the measures you employ to protect its data. The impact of the risk depends on the data's value.

Thus, it is important to classify your data, its sensitivity / risk horizon, the damage it could do if compromised, and categorize it relative to an overall information security management policy. You should also understand and document data flow requirements and processes to identify risks and necessary points of (protection) enforcement. Such activities are also core to standards compliance practices. For more information on how Azure helps ensure internal integrity, refer to the Compliance section of the Microsoft Azure Trust Center.

3.1.1 Understanding Data Risk

Just as there are different security considerations depending on the cloud services model you choose (public, private, or hybrid), there are also different considerations based on the computing model (IaaS, PaaS, and SaaS). Within Azure Cloud Services, there are roles (not to be confused with role-based access control, which can be implemented in Azure applications through WIF, WCF, and Azure AD) that define the capabilities of the VMs they run, and thus the security characteristics for the applications and data that they contain. Figure 9 illustrates the basic architecture of these roles.

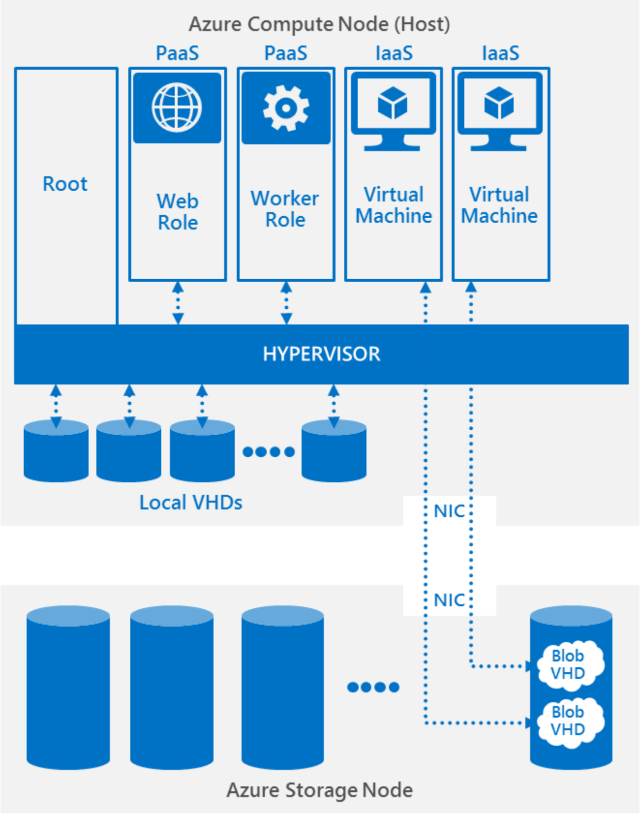

Figure 9: Compute types in Azure that also define security requirements.

A Cloud Services Role (PaaS) is comprised of application files and a configuration, of which there are two (2) types:

- A **Web role** provides a dedicated Internet Information Services (IIS) Web-server used for hosting front-end Web applications.
- Applications hosted within **worker roles** can run asynchronous, long-running or perpetual tasks independent of user interaction or input. (IIS is present, but disabled.)

Virtual Machines (IaaS) are Windows or Linux images uploaded (or mastered in) Azure as a VHD, and offer more comprehensive control over the server instance. This leads to a concept which helps secure Azure infrastructure: specialization. The infrastructure systems within Azure each perform a very narrow set of duties, as shown above with Web and worker roles. The task set of a single system can be defined at development time, and a Virtual Machine or Cloud Services Role Instance is built around that specific task.

3.2 Threats to Your Data

There are many different forms of data in use by Azure workloads and applications, where even the information about that information (i.e., metadata) could contain sensitive details. Even the information about the ways your applications communicate with other parties can reveal things you'd rather outsiders not know about your operations. Table 4 provides an overview of these data structures.

SERVICE	DATA TYPE	DATA ELEMENT
CLOUD SERVICES, STORAGE, & NETWORKING	Customer Data	Customer packages (CSPKG files)
	Application Configuration and Design Data	Customer service configuration (CSCFG) files Customer Fully Qualified Domain Names FQDN
	Customer Data	Customer data in storage
	Access Control Data	Subscription administrator and user account passwords Customer certificates Storage keys of customer accounts Shared Access Signatures
VIRTUAL MACHINES	Customer Data	Custom VM image
	Application Configuration and Design Data	Endpoint configuration
	Access Control Data	Administrator and user account passwords
VIRTUAL NETWORK	Meta Data	IP address/range of VPN gateway
	Access Control Data	Pre-shared key

Table 4: Examples of data in Azure workloads.

Many threats to cloud data are no different than in your on-premises datacenter; it is simply that their form in the cloud transitions to a virtual environment:

- Loss (“data breach”)—through error, disaster, or data theft
- Alteration (“integrity breach”)—by tampering or data corruption
- Misuse—accidental and purposeful disclosure (including eavesdropping)
- Repudiation – events occurring without sufficient audit logs

3.2.1 Data Attack Taxonomy

Attacks can arise through both online and offline mechanisms, as detailed in Table 5 below. An online attack means that the attacker has connected—or is attempting to connect—to a target resource that is actively running or available on a network. An offline attack happens when the attacker has possession of the target resource and can manipulate it however he wishes.

ATTACK TYPE	PROFILE
ONLINE	• Can occur when the VM is in a running state • Typically experienced when there is a compromise of the authentication and authorization of system administrators • Typically, logical attacks such as if an attacker steals VHDs by gaining access to blob storage using a URI and stolen secret
OFFLINE	• Experienced when unauthorized person(s) remove files or physical containers from their intended location(s) • Physical in nature (such as stealing a laptop, removing a physical hard drive from the datacenter, and stealing backup media) • Attacker modifies VHD and plants malware

Table 5: Profiles of the two common data attack scenarios.

3.3 Microsoft Azure Default Protection

Microsoft Azure provides a number of information security measures by default which help mitigate security issues at platform layers outside of a customer’s control. These include both physical and logical security controls, as well as automated security processes, comprehensive information security and privacy policies, and Microsoft Services administrators’ security and privacy training.

The following are native protections in the Azure platform that do not require a customer action. These are some of the ways that Azure protects the platform from threats to the infrastructure and data (including metadata, logs, and more).

Capability	Description
Development	Microsoft Azure is built using many of the same technologies as Windows Server, and is developed, tested and deployed following Microsoft Security Development Lifecycle (SDL) and Operational Security Assurance (OSA) requirements and procedures (including cloud security and privacy considerations). The SDL and OSA methodologies address security threats throughout the development process and operations of Azure services by means that include implementing threat modeling during the design process; following development best practices and code security standards during coding; and, requiring various tools for testing and verification before deployment and during operations of services.
Platform encryption	Please refer to the Platform Encryption section that follows.
Network security and isolation	The distributed and virtual networks in Azure help ensure that each customer's private network traffic is logically isolated from traffic belonging to other customers. A customer subscription can contain multiple isolated private networks (and include firewall, load-balancing, and network address translation): • **Deployment network**: Each deployment is isolated from other deployments at the network level. Multiple VMs within a deployment are allowed to communicate with each other through private IP addresses. • **Virtual network**: Each virtual network is isolated from other virtual networks. Multiple deployments (inside the same subscription) can be placed on the same VNET, and allowed to communicate through private IP addresses. By default, Virtual Machines inside the private network do not receive inbound traffic from outside of the deployment. The administrator defines an input endpoint that specifies which ports on which VMs should receive inbound traffic initiated from outside a deployment's isolated network—enabling traffic from the Internet and other deployments or customers inside Azure.
Secrets and secret storage	Customer secrets, including SSL certificates, private keys, RDP passwords and SAKs can be uploaded via SMAPI, or via the Azure Portal. Both occur over a channel secured by TLS/SSL. Customer secrets uploaded to Azure are stored in an encrypted form.
Log security	Azure provides authenticated and trustworthy logging of security-relevant events that generate an audit trail, and is engineered to be resistant to tampering. This includes system information, such as security event logs in Azure infrastructure VMs and Azure AD.
Penetration testing	Microsoft conducts regular penetration testing of Microsoft Azure infrastructure using a technique known as "Red Teaming". This activity tests the systems using the same techniques and mechanisms as real malicious attackers, against live operational infrastructure. The goal is to identify production environment (i.e., real-world) vulnerabilities, configuration errors, or other security gaps in a controlled process

	and to test security detection, investigation and response processes and technologies.
Hardened workstations	The Azure Operations Team uses hardened administrator workstations for connecting to and managing Microsoft Azure infrastructure components and customer environments (for support incidents). These machines run signed code and a minimal number of applications, operating in a logically segmented environment. Specific Microsoft credentials, using two-factor authentication, are required, and access is monitored and securely logged. Applications such as Outlook and Office which are often the targets of spear-phishing and other types of attacks are run in separate VMs within the hardened workstation.
Personnel	Microsoft Azure staff are required to undergo annual security and privacy training as part of their employment with Microsoft. This includes areas of secure operations, safe data handling practices, and standards of conduct.
Audits	Microsoft follows strict protocols for operating, managing, and monitoring Azure. Comprehensive audits for frameworks such as ISO, SOC, FISMA and FedRAMP are conducted by accredited third-party firms that provide attestations to how data protection requirements are met.
Just-in-Time access	Further protecting customer information, policy dictates that Microsoft personnel should not have persistent access to any customer data, including VMs, files, keys, databases, AD tenants, logs, or other types unless the customer explicitly grants access. If needed to resolve an urgent issue, Microsoft Azure administrators or support staff are provided with "just in time" access to customer data, which is revoked as soon as the issue is closed or requested.
Media destruction	Physical controls are in place to prevent customer data from leaving Azure datacenter premises. In particular, disk drives that are used for customer storage but must be removed (i.e., hardware failure) are securely erased prior to their being returned to the manufacturer for replacement/repair. In the event that a defective disk cannot be fully erased, it is destroyed according to NIST 800-88 guidelines. The same is true for drives purposefully decommissioned.
Data deletion	Where appropriate, confidentiality should persist beyond the useful lifecycle of data. The Azure Storage subsystem makes customer data unavailable once *delete* operations are performed. All storage operations including *delete* are designed to be instantly consistent. Successful execution of a *delete* operation removes all references to the associated data item and it cannot be accessed via the Azure storage APIs. Also, Azure Storage interfaces do not permit the reading of uninitialized data, thus mitigating the same or another customer from reading deleted data before it is overwritten. All copies of deleted data are then garbage-collected. The physical bits are overwritten when the associated storage block is reused for storing other data, as is typical with standard computer hard drives.
Datacenter security	Azure datacenters deploy ISO-compliant safeguards such as locked server cages and racks, Smartcard readers, 24x7 monitoring by security staff, and other mechanisms

designed to prevent data compromise by physical means. More information can be found in the Microsoft Azure Trust Center.

3.3.1 Data Security in Azure AD

Sensitive identity information stored in Azure AD is protected through the following means:

- **Data in transit**: All customer facing Web services are secured with SSL/TLS. All LDAP and partition / replication traffic to and within the directory store (within and between datacenters) is signed.
- **Data at rest**: When at rest, secrets stored in the directory (symmetric keys, private asymmetric keys, passwords) are encrypted using the Distributed Key Manager (DKM).

By default, Windows Azure AD disallows all operations issued by identities in other tenants. A tenant administrator may explicitly grant directory access to identities from other tenants, if desired.

The concept of tenant containers is deeply engrained in the directory service at all layers, from portals to persistent storage. These boundaries ensure a query scoped to a given tenant never returns directory data for another tenant, for instance. Front ends (Azure AD Sync, PowerShell, Graph) all store and retrieve data through an internal directory services API (DSAPI) which calls an authorization layer to ensure the data requested is allowed for the user requesting the data. Three (3) checks are performed at the authorization layer:

1. Is the user enabled for access to Windows Azure AD?
2. Is the user enabled for access to data in this tenant?
3. Is the user's role in this tenant authorized for the type of data access requested?

Access to data in Azure AD requires user authentication via a security token service (STS). Once authenticated, the user principal name (UPN) is read from the authentication token and the replicated partition and container corresponding to the user's domain is determined. Information on the user's existence, enabled state, and role is used by the authorization system to determine whether the requested access to the target tenant is authorized for this user in this session. Certain authorized actions (i.e., create user, password reset) create an audit trail that can be used by a tenant administrator to manage compliance efforts or investigations.

3.4 Platform Encryption

Among Microsoft Azure's data protection capabilities are built-in services, components and configurations that apply encryption to internal data and traffic. These serve to enable enhanced security for customer information, and also help enforce data governance and compliance with industry regulations (and are mandated as such).

Many of these mechanisms are enabled by default in the platform while others need to be configured by a customer administrator (such as IPsec VPN). Some can be optionally invoked at VM boot-time through service configuration files, or called by application components directly.

Azure implements encryption using both symmetric and asymmetric keys for encrypting and protecting confidentiality of data:

- Software-based AES-256 for symmetric encryption/decryption
- 2048-bit or better for asymmetric keys
- SHA-256 or better for secure hashing

3.4.1 Encryption in Transit

Microsoft Azure uses virtual networking to isolate tenants' traffic from one another, employing measures such as host- and guest-level firewalls, IP packet filtering, port blocking, and HTTPS endpoints. However, most of Azure's internal communications, including infrastructure-to-infrastructure and infrastructure-to-customer (on-premises), are also encrypted.

For communications within an Azure datacenter, Microsoft manages networks to assure that no VM can impersonate or eavesdrop on the IP address of another. TLS/SSL is used when accessing Azure Storage or SQL Databases, or when connecting to Cloud Services. In this case, the customer administrator is responsible for obtaining a TLS/SSL certificate and deploying it to their tenant infrastructure.

3.4.1.1 VM to VM

Data traffic moving between Virtual Machines in the same deployment or between tenants in a single deployment via Microsoft Azure Virtual Network can be protected through encrypted communication protocols such as HTTPS, SSL/TLS, or others.

Data leaving a customer's Cloud Service should be considered Internet-facing, and so appropriate safeguards such as HTTPS or VPN are recommended.

3.4.1.2 Customer to Cloud

Moving data into and out of your cloud environment is protected through the options available in Azure. This includes management operations, data transfers, and key provisioning. Customers can optionally configure TLS/SSL for defense-in-depth on their Virtual Networks; TLS/SSL is mandatory when accessing the Azure Portal or System Management API (SMAPI).

For small amounts of data, connections directly to your Azure Virtual Network can be made over encrypted connections, such as by an IPsec VPN into your tenant environment; larger data sets can be moved over an isolated high-speed channel such as the new ExpressRoute feature. If ExpressRoute is being used, you can encrypt the data at the application-level using TLS/SSL or other protocols for added protection.

In addition, when interacting with Azure Storage through the Azure Portal, all transactions occur via HTTPS. Storage REST API over HTTPS can also be used to interact with Azure Storage and Azure SQL Database. When populating data into Azure SQL Database, you can encrypt information before it is copied over. Note that data only remains encrypted until it is used and placed in memory on the Azure SQL Database compute node, at which point it exists in an unencrypted state.

3.5 Data Deletion

Data destruction techniques vary depending on the type of data object being destroyed, whether it be whole subscriptions themselves, storage, Virtual Machines, or databases. In a multi-tenant environment such as Microsoft Azure, careful attention is taken to ensure that one customer's data is not allowed to either "leak" into another customer's data, or when a customer deletes data, no other customer (including, in most cases, the customer who once owned the data) can gain access to that deleted data.

3.5.1 Subscriptions

If a subscription is cancelled or terminated, Microsoft will retain Customer Data for 90 days to permit the customer to extract its data. Microsoft will then delete all Customer Data within another 90 days after the retention period (i.e., by day 180 after cancelation or termination).

If a storage account is deleted within an existing subscription (or when a subscription deletion has reached its timeout), the storage account is not actually deleted for two weeks to allow recovery from mistakes. But once a storage account is finally deleted, or when blob or table data is deleted outside the context of a storage account deletion, the data is no longer available.

> ***NOTE:*** *If you want to make storage data unrecoverable faster, you should delete tables and blobs individually before deleting the storage account or subscription.*

3.5.2 Microsoft Azure Storage

In Azure Storage, all disk writes are sequential. This minimizes the number of disk "seeks", but requires updating the pointers to objects every time they are written (new versions of pointers are also written sequentially). A side effect of this design is that if there is a secret on disk, you can't ensure it is gone by overwriting with other data. The original data will remain on the disk and the new value will be written sequentially. Pointers will be updated such that there is no way to find the deleted value anymore.

Once the disk is full, however, the system has to write new logs onto disk space that has been freed up by the deletion of old data. Instead of allocating log files directly from disk sectors, log files are created in a file system running New Technology File System (NTFS). A background thread running on Azure Storage nodes frees up space by going through the oldest log file, copying blocks that are still referenced from that oldest log file to the current log file (and updating all pointers as it goes). It then deletes the oldest log file. So there are two categories of free space on the disk: space that NTFS knows is free, where it allocates new log files from this pool; and, space within those log files that Azure Storage knows is free since there are no current pointers to it.

3.5.3 Microsoft Azure Virtual Machines

Virtual Machines are stored in Microsoft Azure Storage as blobs, so the deletion rules apply as explained above. The virtualization mechanism, however, is designed to ensure that those spots on the disk cannot be read by another customer (or even the same customer) until data is written again, thus mitigating the threat of data leakage. When a new virtual disk is created for a VM, it will appear to the VM to be zeroed, however, the explicit zeroing of the data buffers occur when a portion of the virtual disk is read before it is written. If a VM instance is reinitialized in place, it’s the same as if it had been moved to new hardware.

3.5.4 Azure SQL Database

With Azure SQL Database, deleted data is marked for deletion. If an entire database is deleted, it is the equivalent of deleting the database’s entire contents. The SQL Database implementation is designed to ensure user data is never leaked by disallowing all access to the underlying storage except via the SQL Database API. That API allows users to read, write, and delete data, but does not have a way to express the reading of data that the user has not previously written.

3.6 Compute Security for Data in Use

One of the fundamental design methodologies used by Azure to help protect data is to “assume breach”, an extension to the traditional concept of defense-in-depth. In short, Microsoft’s procedures and systems are designed under the assumption that one or more systems or procedures may fail. Azure was built from the ground up as a multi-tenant service on shared infrastructure, with the goal of protecting customer information at every layer of the stack, across both customer-facing and internal management services, with checks and balances in the design and implementation at each layer.

Azure widely employs the concept of compartmentalization to enforce this principal. For example, every host node in the Azure datacenter (the machines which are used to host customer VMs) are bastions—customer-to-host isolation boundaries are maintained by Azure Hyper-V. Should an operator need to log into a system to troubleshoot, they get credentials for that specific host and that host alone. After a short time, those credentials expire.

Azure considers secrets management one of the platform's engineering fundamentals. When a role is built by the infrastructure, it is provisioned with just the secrets it needs to do its job. An extension of this functionality is provided to customers for using Azure to manage private keys (in particular, customers' keys are stored and transmitted in encrypted form, and kept in the Azure Secret Store). The Azure infrastructure has the ability to push secrets into machines, and keys are regularly rotated inside Azure.

3.7 Physical Data Security

A common question from customers using hosted services is, "Can somebody physically steal my data?" The answer is complex, because numerous factors stand in the way of anybody attempting such a feat.

- **Entering the Datacenter**: attested by multiple security and compliance audits, Microsoft employs rigorous operations and processes to prevent unauthorized access, including 24x7 video monitoring, trained security personnel, key-locked server racks (that house compute, storage, and networking hardware), smart cards and biometrics controls. Any access that is granted is logged.
- **Stealing a Disk Drive**: To target your business specifically, a trespasser would have to know which datacenter, building, floor, room, and server rack on which your data resides. In addition, Azure Storage data is written to disk in small chunks using striping. Thus, a customer's data likely spans across multiple disks (and large files, such as databases, may span multiple drives). At this point, the person would need to know which disk enclosure and drive(s) to pull out. A thief (even randomly grabbing disks) would also need your secret Azure Storage keys to read the media.
- **Copying Data onto Removable USB Media**: As with disk theft noted above, using removable media means discovering which storage device has the desired data. Azure nodes are physically protected and include headless operation, hardware passwords, and hardening techniques to prevent local code execution. Similarly, server clusters do not support optical media, and physical ports are blocked from access; any attempts to connect in this way generates security alerts.
- **Network Sniffing** (physical connection via wired, wireless, or remote tap): Azure's internal routers do not connect to any Internet-facing endpoints and run in a highly restricted mode to block any non-authenticated connections. There is no wireless access to any Azure production network systems or infrastructure, effectively eliminating the threat of mobile device exploits. And, Microsoft's Global Foundation Services (GFS) group operates Azure in ISO 27001-compliant datacenters, with security controls to lock down physical network access ports.

For additional information on physical security, visit the Microsoft GFS web site.

4 Customer-Configurable Protection

In this section, we will discuss the main data protection concepts within Azure that are controlled by you, the customer:

- Cryptography for storing data, in applications, and on the network (encryption and decryption).
- Key management (provisioning, lifecycle management, security/protection).
- Authentication, authorization, and access control (discussed previously).

These capabilities combine to provide a compliant foundation to help ensure control over the integrity, privacy, and security of your critical data. They can help you establish protective measures that automatically defend against hostile acts, and also provide assurance that your tactics are effective.

The data protection mechanisms (as shown in Figure 10 below) you use should be appropriate for the workload and data containers in question. In particular:

- Data security consists of encryption/decryption, key management (key lifecycle), and key protection. Does the encryption solution address all three of these?
- Does the encryption solution protect against online and/or offline attacks?
- Is the encryption provided at the most optimal data/container layer?
- Is the encryption transparent to the workload types that occur?
- Are authorized personnel able to recover the encrypted data when required?

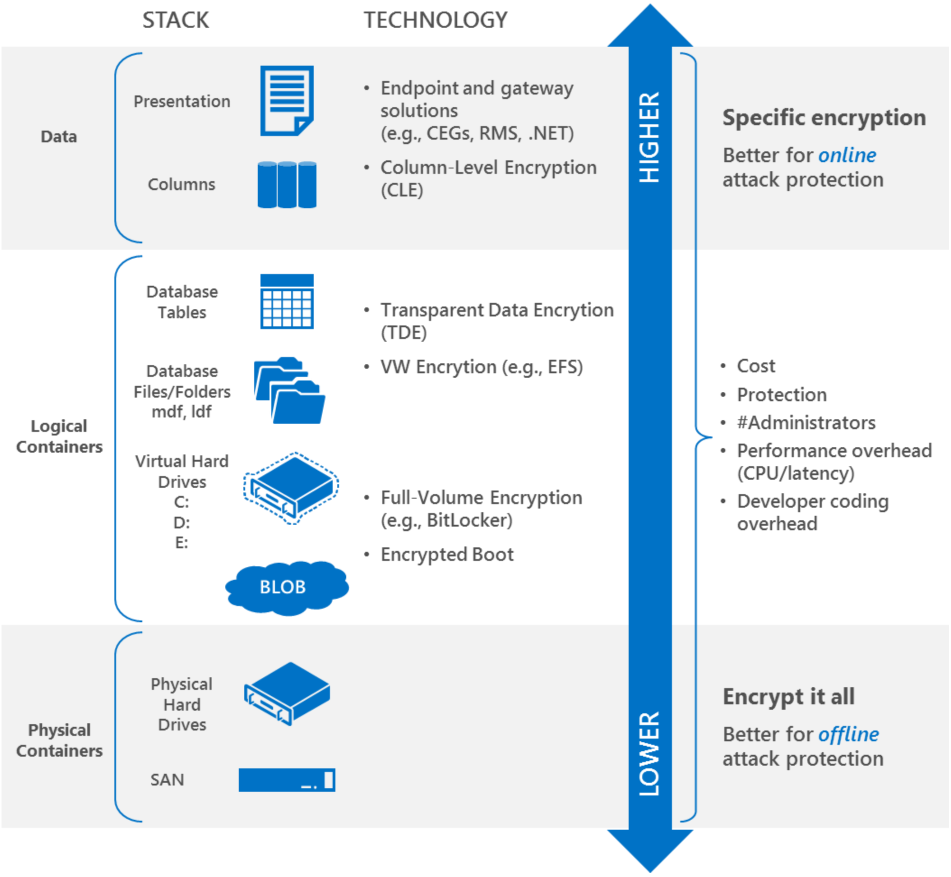
STACK
TECHNOLOGY
Data
Presentation
Columns
• Endpoint and gateway solutions (e.g., CEGs, RMS, .NET)
• Column-Level Encryption (CLE)
HIGHER
Specific encryption
Better for *online* attack protection
Logical Containers
Database Tables
Database Files/Folders mdf, ldf
Virtual Hard Drives C: D: E:
BLOB
• Transparent Data Encrytion (TDE)
• VW Encrytion (e.g., EFS)
• Full-Volume Encryption (e.g., BitLocker)
• Encrypted Boot
• Cost
• Protection
• #Administrators
• Performance overhead (CPU/latency)
• Developer coding overhead
Physical Containers
Physical Hard Drives
SAN
LOWER
Encrypt it all
Better for *offline* attack protection

Figure 10: Common data encryption mechanisms.

4.1 Volume Level Encryption

In general, cryptography consists of encryption/decryption, key management (e.g. key lifecycle), and key security. Windows operating systems provide encryption routines (.NET CAPI, CNG) that enable customers to encrypt data before storing it in Azure, and these same mechanisms can also be used within Azure VMs.

The ultimate choice of where you should do your encryption / decryption (in the cloud, on-premises, in-application, on the client, etc.) will depend on the level of control you need to maintain, the cost you are willing to incur (e.g., performance, administration), the confidentiality that must be kept, and the potential for incurring risk. Table 6 below provides an overview of common options.

	LAYER	ENCRYPTION SUPPORT	KEY MANAGEMENT	DETAILS
	Application	.NET Cryptography API	Managed by customer	.NET Cryptography documentation
		Encrypt data using RMS SDK	Managed by customer via on-premises ADRMS service or Azure RMS	RMS SDK documentation
TRANSPARENT DATA	**Platform**	SQL TDE/CLE on SQL Server on Azure IAAS VMs	Managed by customers	SQL TDE/CLE documentation
		StorSimple provides primary, backup, archival	Managed by customers	More Information
	System	EFS, BitLocker support for data and boot volumes	Managed by customers	BitLocker command-line tool
	Others	Import/Export of data onto drives can be protected by BitLocker	Managed by customers	Import/export step by step blog

Table 6. Cryptographic mechanisms available across main Microsoft Azure services and containers.

4.1.1 BitLocker Drive Encryption

Azure Virtual Machines are typically associated with storage disks (VHDs) which are in turn stored in Azure Storage. In Azure Storage, data is broken into small chunks and each small chunk is striped across multiple physical disks, providing safeguards against loss of that disk. Additional protection such as drive encryption can be used to mitigate threats such as a compromise of a SAK (used by a VHD). With encrypted disks, even when an unauthorized user obtains the key and in turn uses the key to fetch the VHD from Azure Storage, the VHD is encrypted and thus makes the data unreadable.

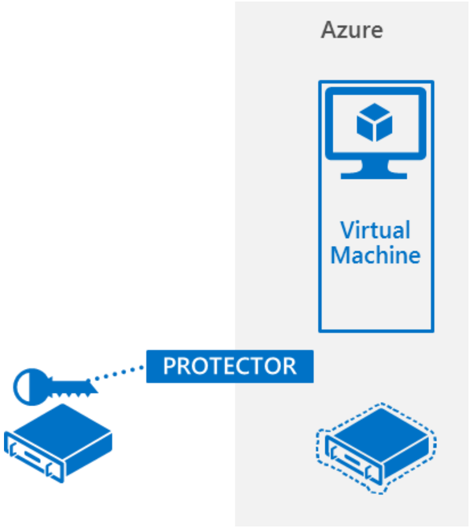

Figure 11: BitLocker drive encryption.

Windows offers Full Disk Transparent Encryption through BitLocker for Data Volumes and Boot Volumes, which is transparent to the application. The same BitLocker Drive Encryption (BDE) can be implemented for Azure VMs and VHDs using command line tools such as 'manage-bde'. BitLocker enables volume encryption through several different protectors, such as passwords and certificates. As shown in Figure 11. Azure PowerShell will allow you to remotely execute encryption commands using 'manage-bde', or encryption can be controlled by startup scripts. An auto-unlock feature in BitLocker allows you to unlock the volume automatically without an interactive session.

Keys can be protected using on-premises key management services including Hardware Security Modules (HSMs). In Azure VMs, boot volumes can also be encrypted using BitLocker. For additional information, refer to BitLocker documentation including this MSDN article on the 'manage-bde' BitLocker command-line tool.

4.1.2 Drive Encryption - Partners

Partners such as Trend Micro and others offer volume-level encryption, and manage policies surrounding encryption. These partner solutions also integrate with third-party HSMs and offer solutions for both Windows and Linux VMs. Encryption is transparent to the OS and the applications, thus applications do not need to be changed.

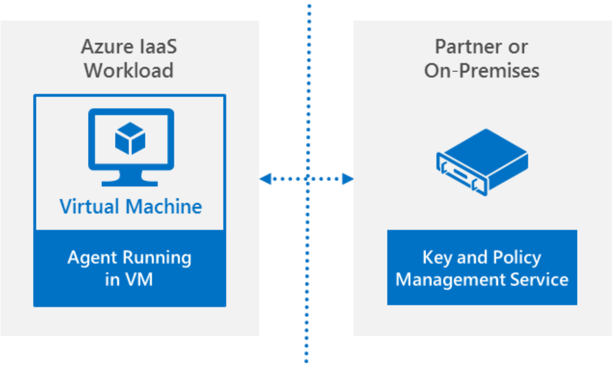

Figure 12: Using third-party HSMs.

While the implementation of the solutions can vary, an agent typically 'sits' in the OS stack between the disk driver and the file system driver, encrypting the data. Encryption persists even after the instance is stopped, as shown in Figure 12.

Keys can also be managed by an off-cloud system such as an HSM or other Key Management Service.

4.1.3 Key Management and Security

Encryption and authentication do not improve security unless the keys themselves are well protected. It is generally considered a critical IT security task to manage key lifecycles, as proper key management is important to maintaining high security, high reliability, and low overhead.

Encryption key management is left to the implementation of the end-customer. Customers can develop a secure architecture that works for their particular solutions, and have full control over data encryption.

4.1.4 Subscription and Service Certificates

The Azure platform builds on the straightforward key management methods incorporated into the Windows security model, providing the ability to use certificates to secure data (both Virtual Machines and Cloud Services can use any cryptographic facilities in Windows, including those in .NET, CAPI, and CNG).

The main type of certificate that plays a role in securing customer applications or services are called Service Certificates. These are traditional SSL certificates (uploaded by the customer) used to secure endpoint communications. Service certificates can also be used for other purposes, such as Public-Key Cryptography Standards (PKCS)-encrypting data, or to encrypt secret configuration information such as Storage Access Keys.

Azure provides each subscription with an associated logical certificate store that enables automatic deployment of service-specific certificates, and to which customers can upload their own. The certificate store is independent of any hosted service, so it can store certificates regardless of whether they are currently being used by any of those services. These certificates and other credentials uploaded to Azure are stored in encrypted form.

Azure also provides an administrative path to upload certificates and private keys, but not to retrieve private keys. Customer secrets, including certificates, private keys, RDP passwords and SAKs are communicated through the SMAPI via Representational State Transfer (REST)-based protocols or the Azure Portal using SSL. Certificates and private keys are stored in an encrypted form in Azure on the Fabric Controller. Customer certificates can be directed to customer VMs where they are automatically installed in non-exportable form.

Certificates can be managed separately from services, and may even be managed by different individuals. For example, a developer may upload a service package that refers to certificates that an IT manager has previously uploaded to the Azure secret store. The IT manager can manage and renew those certificates without stopping the service or uploading a new service package.

4.2 Encryption for SQL Server in Azure Virtual Machines

SQL Server Transparent Data Encryption (TDE) is a proven mechanism for providing storage encryption for on-premises SQL Server 2008 and above installations. TDE is set up through SQL Server configuration and requires no application changes, providing protection from physical storage device theft as well as logical breaches where access to the file system is gained and database files are exposed. More details on SQL Server TDE can be read here.

SQL Server Column-Level Encryption (CLE) offers a more granular level of encryption where data is not decrypted until it is used (conversely, TDE encrypts the entire database in storage, and then decrypts each page in the database fully when it is accessed). Therefore, even if a page is loaded in memory, sensitive data is not in the clear until SQL Server processes it. CLE does require the calling application to be modified to encrypt and decrypt data written to tables. Also, there are performance implications associated with it that customers should consider, as encryption does affect query optimization on the encrypted columns. For this reason, CLE is usually used when the data to be encrypted is small or there are other custom design requirements. More details on SQL Server CLE can be read here.

Encryption, however, is only effective when the keys used to encrypt data are kept secret. Under default scenarios, SQL Server keeps all of the keys required to decrypt the database in the master database. If this master database is kept on the same storage device as the user database, the exposure of that single storage device could lead to data exposure. SQL Server provides an extensible key management (EKM) provider architecture to redirect key security and storage to a key management service (KMS) external to the server itself.

Hardware security modules (HSMs) are hardware devices specifically designed to assist in keeping encryption keys secret. A number of commercial vendors that provide HSMs for key management also have EKM providers for SQL Server.

4.2.1 Cloud Implementation

As you migrate applications to the cloud, you can leverage the same TDE/CLE functionality in VMs running SQL Server to encrypt databases as you do for on-premises deployments. This creates a security barrier between the at-rest data persisted in cloud storage and a potential attacker. Even if an attacker were to gain access to the VHDs that back a database server, or manage to gain physical access to a storage device, the data is encrypted and therefore further protected from breach.

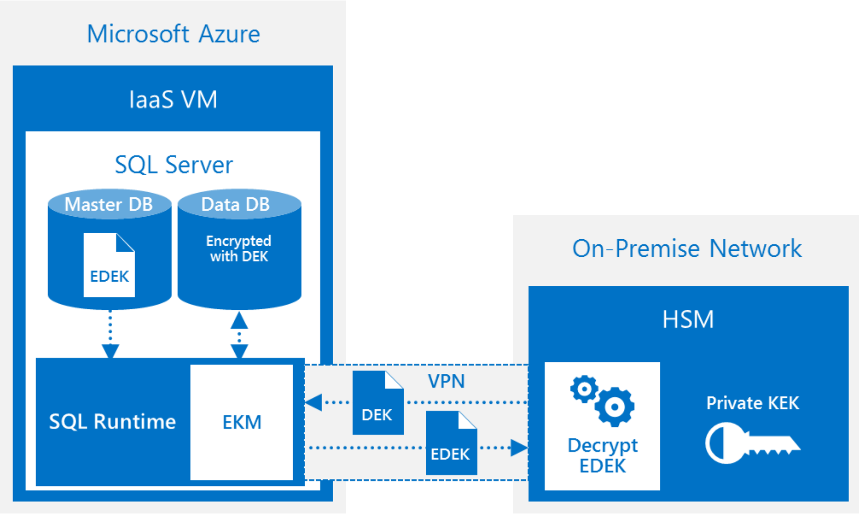

Figure 13: Key flow in SQL Server encryption.

Shown in the Figure 13, when a HSM cryptographic provider is specified, rather than using a certificate derived from the Database Master Key, SQL Server will use an asymmetric key from the HSM to encrypt the Database Encryption Key (DEK).

When a customer has a hybrid network (the Azure Virtual Network (VNET) is connected to their on-premises network via site to site VPN or Azure ExpressRoute), they can redirect the key management tasks to an on-premises HSM or Key Management Service (KMS), effectively retaining control of keys in off-cloud infrastructure. The Extensible Key Management (EKM) providers available from the HSM vendors for SQL Server can be leveraged for this purpose.

When a SQL Server instance starts and attempts to mount a database that is encrypted in this fashion, it will request that the EKM provider, and subsequently the HSM, decrypt the Encrypted Data Encryption Key (EDEK). The returned DEK is then stored in memory and used for any subsequent decryption of the database.

This method prevents anyone without direct access to the HSM from decrypting the database, even if both the master database and user database files were compromised.

This scenario has been tested by Microsoft using SQL Server 2008 and 2012 Enterprise instances in Azure VMs. The VMs used were in a VNET that was joined to the Microsoft on-premises network using ExpressRoute and were joined to the corporate domain. The SQL Server instances leveraged an EKM module from Thales to redirect key management to a Thales HSM on the Microsoft corporate network. Customers should contact their HSM vendors directly to see if they have tested their network accessible HSMs in this scenario.

SQL Server CLE with key management by an on-premises HSM does have some additional performance implications since key requests are usually more frequent than for SQL Server TDE (every time a SQL column is encrypted or decrypted). Each time this happens the key has to be fetched remotely from the on-premises HSM. Customers should consider this in their design. Customers should contact their HSM vendor for the best way to secure HSM access from Azure VMs.

4.3 Azure Rights Management Services

Microsoft Rights Management Services (RMS) is a comprehensive toolkit that includes an SDK for developers, ready-to-use applications for information workers, and management tools for IT administrators. RMS enables an organization to:

- Encrypt and decrypt data
- Manage, distribute, and track the distribution of encryption keys
- Enforce who can receive keys, and what they can do with the decrypted data

Data must often be shared across applications, computers, devices, users, and organizations, and often flows from sender to receiver in multiple hops. In the RMS model, encryption and access policies travel with the data, up to the last mile. (The RMS client SDK is available on Windows, iOS, Android, and OS X.)

4.3.1 RMS Basics: How the RMS Components Work Together

Every RMS flow includes an *RMS server*, and one or more *RMS-aware applications*, which in turn use the *RMS SDK*. Figure 14 below illustrates this, showing a user in blue (*publisher*) sharing data with a user in green (*consumer* or *recipient*). The 'user' is anybody with an identity, such as a real human or a service.

First, the publisher sets up an *RMS server*. During setup, the RMS server generates (or lets the publisher import) a unique RSA key pair per organization, called the Server Licensor Certificate (*SLC) key*.

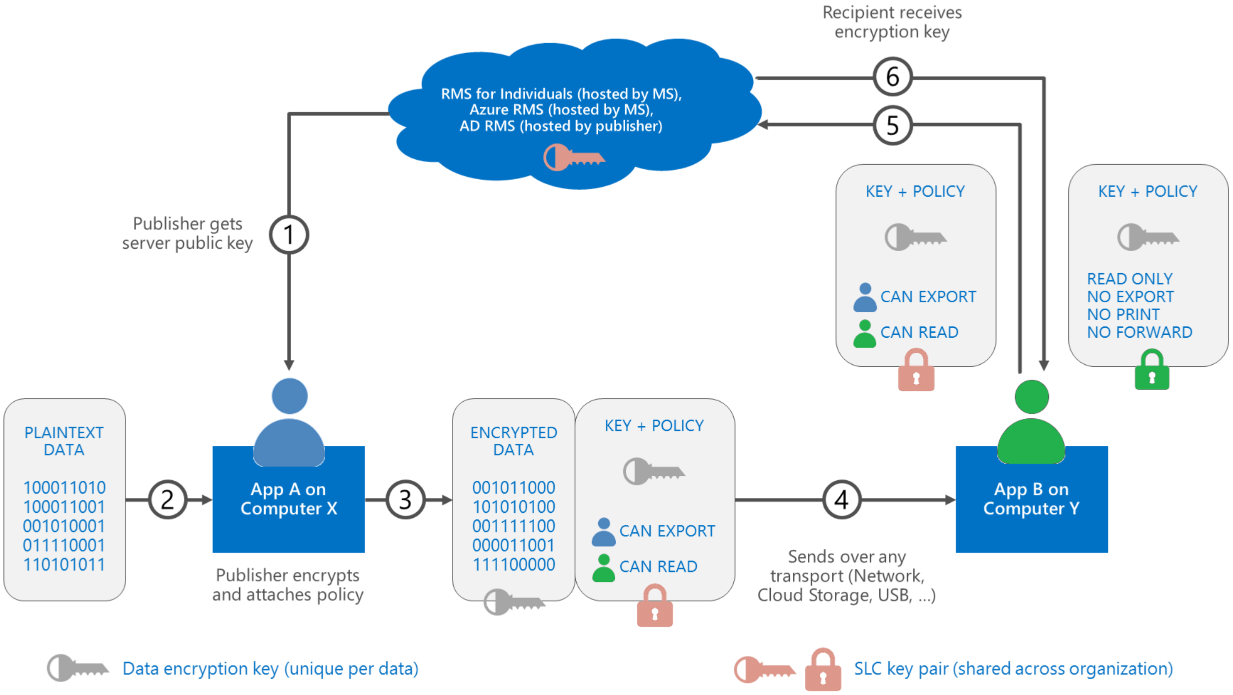
Recipient receives encryption key
RMS for Individuals (hosted by MS), Azure RMS (hosted by MS), AD RMS (hosted by publisher)
6
5
Publisher gets server public key
1
KEY + POLICY
CAN EXPORT
CAN READ
KEY + POLICY
READ ONLY
NO EXPORT
NO PRINT
NO FORWARD
PLAINTEXT DATA
100011010
100011001
001010001
011110001
110101011
2
App A on Computer X
3
Publisher encrypts and attaches policy
ENCRYPTED DATA
001011000
101010100
001111100
000011001
111100000
KEY + POLICY
CAN EXPORT
CAN READ
4
Sends over any transport (Network, Cloud Storage, USB, ...)
App B on Computer Y
Data encryption key (unique per data)
SLC key pair (shared across organization)

Figure 14: Key and data flow in RMS.

Second, the publisher needs an *RMS-aware application* (application A in the diagram below).

- Application A uses the *RMS SDK* (not shown) to generate a new *data encryption key* (shown in gray), and to encrypt the data with that key (steps 2 and 3).
- Application A specifies a *usage policy* (who can decrypt, what they can do with the data).
- The RMS SDK retrieves the SLC public key from the RMS server (shown as step 1). It uses this to encrypt the data encryption key as well as the usage policy, together called the *publishing license*.

Third, the publisher sends the encrypted data and publishing license to the intended recipient/consumer (step 4). It does not matter what transport they use for this, or how many "hops" it takes, since the data travels encrypted.

Lastly, the recipient uses application B to decrypt the data:

- Application B uses the RMS SDK to locate the RMS server to authenticate to it, and to present the encrypted publishing license (step 5).
- If the publishing license authorizes the recipient to decrypt the data, then the RMS server returns the data encryption key, and the specific rights that the user (in green) has (step 6).
- Application B uses the RMS SDK to decrypt the data. The application and RMS SDK uphold the usage policy. (This step assumes the user is trusted.)

One notable aspect is that *the RMS server does not see the data*. It only brokers the key exchange. Data and keys flow from publisher to consumer through separate routes, so nobody other than the publisher and consumer can assemble both the data and key.

For more details on how RMS works, please visit http://blogs.technet.com/b/rms/archive/2012/04/16/licenses-and-certificates-and-how-ad-rms-protects-and-consumes-documents.aspx.

4.3.2 RMS Server Choices

Microsoft offers three implementations of an RMS server.

- A free hosted service, called *Microsoft Rights Management for Individuals*. This is the easiest option for beginners. Sign up at https://portal.aadrm.com.
- A premium hosted service with additional controls for organizations, called *Microsoft Rights Management Service* or *Azure RMS*. See http://technet.microsoft.com/library/dn655136.aspx for the pre-requisites and the options for how to purchase or trial.
- An on-premises implementation, called *Active Directory Rights Management Services*. See http://technet.microsoft.com/en-us/windowsserver/dd448611.aspx.

4.3.3 RMS SDK for developers

As a developer, you can use the RMS SDK to encrypt or decrypt data in your application. The RMS SDK allows you to encrypt or decrypt bytes in memory, or an entire file at a time.

When encrypting an entire file, the RMS SDK uses pre-defined formats for the popular file types (PDF, Microsoft Office formats, JPG, TXT) and a generic container format (called PFILE) for all other file types. This is done so existing RMS applications can open those files.

When encrypting byte by byte, you decide where you store your encrypted data, the format in which it is stored, and where you store the publishing license—as long as the applications that consume (decrypt) the data are aware of your configuration.

Two code samples in particular are relevant to Azure application developers. One sample shows a Cloud Service using RMS to encrypt data before storing it in Azure Storage. The other sample shows a Web application that uses RMS to encrypt data that it distributes to users. These samples are available at http://go.microsoft.com/fwlink/?LinkId=398638.

You, or whoever uses your application, will need an accompanying RMS server to complete the solution.

For more information about using the RMS SDK, please visit http://msdn.microsoft.com/en-us/library/hh552972(v=vs.85).aspx.

4.3.4 RMS-Aware Applications

You may be able to leverage an existing application to accomplish your task, instead of writing a new application from scratch. The following applications are RMS-aware:

- For email: Outlook on Windows, NitroDesk Touchdown on devices. Also, Outlook Web Access in Exchange Server (if your Exchange administrator has enabled IRM)
- For Office documents: Word, PowerPoint, and Excel
- For PDF: FoxIt PDF reader and NitroPDF reader
- For others: Use the RMS Application, available at https://portal.aadrm.com/Home/Download

4.3.5 RMS in Organizations

If you are deploying the RMS server for use in an organization, or if you want to customize your RMS server configuration, consider Azure RMS or AD RMS on-premises. These versions of the RMS server give an IT administrator features beyond the free RMS for Individuals service. Examples of such features are:

- Customize how you manage the SLC keys for your organization.
- Monitor who received data encryption keys.
- Require your employees to assign specific usage policies to their data via templates.
- Enable Exchange and SharePoint Servers to publish and consume RMS-protected data.

For more information please visit http://technet.microsoft.com/en-us/dn175751.

4.3.6 Key Management with RMS

RMS provides options to protect SLC keys and the distribution of data encryption keys. Specifically, AD RMS allows you to protect your SLC key with a compatible HSM.

Azure RMS also allows you to *bring your own key* (BYOK). In this mode, you upload an SLC key from your on-premises HSM to Microsoft's HSMs. The key stays protected by the HSMs at all times, and you have access to a usage log. Combined, you have high assurance of proper key usage. For more information on these features, please visit http://technet.microsoft.com/en-us/library/dn440580.aspx.

Authentication in RMS is tied to Active Directory or Azure Active Directory. When you delete a user (or application principal) from the linked AD or Azure AD, that person (or application) can no longer receive data encryption keys from the RMS server, even if they were previously authorized to receive such keys.

4.3.7 Tracking Key Distribution

Both AD RMS and Azure RMS offer near-real time logs, enabling the administrator to monitor who has received access to the data encryption keys for a given piece of data.

For ADRMS, see http://technet.microsoft.com/en-us/library/dd772686(v=WS.10).aspx. For Azure RMS, see http://blogs.technet.com/b/rms/archive/2014/01/07/enabling-and-using-logging-in-azure-rms.aspx.

5 Protecting Data through Redundancy and Backup

Azure offers a number of mechanisms for ensuring data availability. Azure Storage replicates data across multiple physical drives, you can create availability sets for applications such as SQL Server in Virtual Machines, and use auto-scaling and Traffic Manager to improve load balancing and failover—which can help your infrastructure survive a DoS attack (although it isn't part of this white paper's scope).

5.1 Azure Storage

The Windows Azure Storage service geo-replicates customer's Blob and Table, and Queue data. Data is replicated between two (2) locations hundreds of miles apart within the same region (i.e., between North and South United States, between North and West Europe, and between East and Southeast Asia). Geo-replication is provided for additional data durability in case of a major datacenter disaster and even transient hardware failures. You have three (3) options for replicating the data in your storage account:

- **Locally redundant storage** (LRS) is replicated three times within a single datacenter. When you write data to a blob, queue, or table, the write operation is performed synchronously across all three replicas. LRS protects your data from normal hardware failures.
- **Geo-redundant storage** (GRS) is replicated three (3) times within a single region, and is also replicated asynchronously to a second region hundreds of miles away from the primary region. GRS keeps an equivalent of six (6) copies (replicas) of your data (three in each region). GRS enable Microsoft to failover to a second region if we can't restore the first region due to a major outage or disaster. GRS is recommended over locally redundant storage.
- **Read-access geo-redundant storage** (RA-GRS) provides all of the benefits of geo-redundant storage noted above, and also allows read access to data at the secondary region in the event that the primary region becomes unavailable. Read-access geo-redundant storage is recommended for maximum availability in addition to durability.

5.2 Azure Backup

Azure Backup works with the System Center 2012 Data Protection Manager (DPM) disk-based protection feature. When you enable online protection, the disk-based replicas are backed up to an online location. Backups of your on-premises datacenter servers (or cloud services) are encrypted before transmission, and stored encrypted in Azure using AES-256 (as shown in Figure 14 below). Backups by the Windows file system and through DPM are similarly encrypted automatically. You can optionally use Windows Server File Classification Infrastructure (FCI) for an additional level of protection by identifying sensitive files for a rights management process, such as Azure RMS.

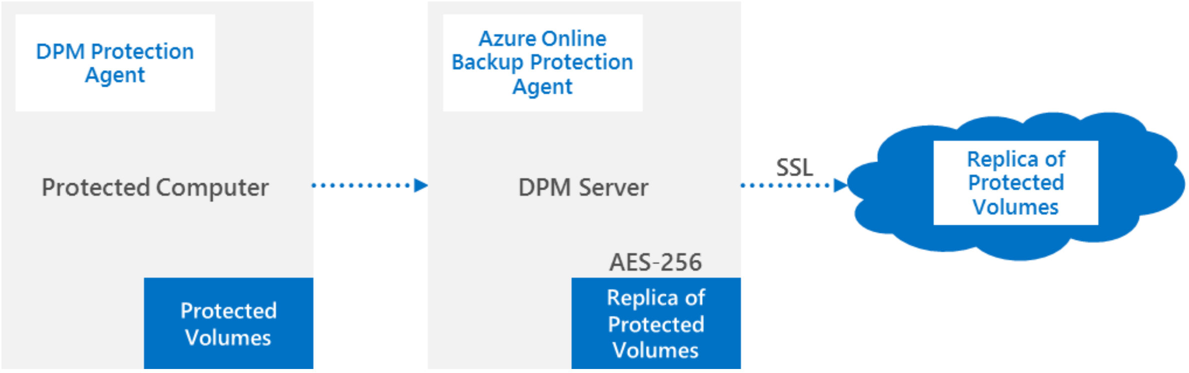
DPM Protection Agent
Protected Computer
Protected Volumes
Azure Online Backup Protection Agent
DPM Server
AES-256
Replica of Protected Volumes
SSL
Replica of Protected Volumes

Figure 14: Data flow for encrypted backups in Azure.

5.3 StorSimple Cloud Integrated Storage (CiS)

The process of data protection by StorSimple is a two-part effort providing both local and cloud snapshots. Local and cloud snapshots each serve a specific purpose when it comes to backup and recovery of data. This method of backup addresses concerns by IT departments over security and management of data by providing continuous availability of backup data and the associated desired recovery points, regardless of its location within the cloud integrated storage model.

Data is protected at multiple levels. Each file is broken into blocks; deduplication occurs at the block so that only changed blocks are stored. Each block that is sent to the cloud is encrypted with AES 256-bit encryption and compressed (the private key is stored at the client premises, not in the cloud) before being stored.

Encryption is applied to all Microsoft Volume Shadow Copy Services (VSS) data transmitted and stored in the cloud by CiS systems to ensure its security. SHA-256 hashing is applied to all data transmitted and stored in the cloud as a means to provide data integrity.

6 Privacy and Accountability

Microsoft is committed to protecting the security of personal information. The online service delivery teams use a variety of security technologies and procedures to help protect information from unauthorized access, use, or disclosure. Microsoft software development teams apply the PD3+C principles, defined in the Security Development Lifecycle (SDL) and Operational Security Assurance (OSA), throughout the company's development and operational practices:

- **Privacy by Design** – Microsoft uses this principle in multiple ways during the development, release, and maintenance of applications to ensure the data collected from customers is for a particular purpose and that the customer is given appropriate notice in order to enable informed decision-making. When data to be collected is classified as highly sensitive, additional security measures such as encrypting while in transit, at rest, or both, may be taken.
- **Privacy by Default** – Microsoft offerings ask customers for permission before collecting or transferring sensitive data. Once authorized, such data is protected by means such as access control lists (ACLs) in combination with identity authentication mechanisms.
- **Privacy in Deployment** – Microsoft discloses privacy mechanisms to organizational customers as appropriate to allow them to establish appropriate privacy and security policies for their users.
- **Communications** – Microsoft actively engages the public through publication of privacy policies, white papers, and other documentations pertaining to privacy.

7 Summary

Maintaining information security and privacy is a continuous process that spans both your on-premises datacenter and your Azure environment. The following table highlights the main capabilities that you should consider implementing:

Scenario	***Threats Mitigated***	***Encryption Technology***	***More Information***
Running Virtual Machines (IaaS) with sensitive data on VHDs stored in Azure Storage and attached to the compute instances (Windows workloads)	• Loss of Disks • Loss of Storage Account Keys where VHDs are stored	• BitLocker Drive Encryption • Volume Level Encryption by ISV partners	http://technet.microsoft.com/en-us/library/cc732774.aspx
Running Virtual Machines with sensitive data on VHDs stored in Azure Storage and attached to the compute instances (Linux workloads)		• Volume Level Encryption by ISV partners	http://azure.microsoft.com/en-us/gallery/store/#all
Running a Virtual Machine with SQL Server database		• SQL Transparent Data Encryption • SQL Column Level Encryption	http://technet.microsoft.com/en-us/library/bb934049.aspx http://technet.microsoft.com/en-us/library/cc278098(v=SQL.100).aspx#_Toc189384682
Running an Virtual Machines or Cloud Services with data in Azure Storage and using Storage Client Libraries / REST API	• Loss of Disks	• Application Level Encryption with .NET crypto API or other languages	http://msdn.microsoft.com/en-us/library/dn720893(v=vs.85).aspx
Running an on-premises workload with data in Azure Storage and using Storage Client Libraries / REST API	• Loss of Disks • Cloud Service Provider Administrators • Attacks originating from the internet against the cloud	• Application Level Encryption with .NET crypto API or other languages	http://blogs.msdn.com/b/windowsazurestorage/archive/2012/11/06/windows-azure-storage-client-library-2-0-tables-deep-dive.aspx
Workloads extending Azure as a backup / archiving / extension of on-premises storage		• Azure Backup • StorSimple	http://blogs.msdn.com/b/windowsazure/archive/2011/04/22/storing-encrypted-data-in-windows-azure.aspx

8 References and Further Reading

The following resources are available to provide more general information about Microsoft Azure and related Microsoft services, as well as specific items referenced in the main text:

- Microsoft Azure Home – general information and links about Microsoft Azure
  - http://www.microsoft.com/windowsazure/
- Microsoft Azure Documentation Center – developer guidance and information
  - http://msdn.microsoft.com/en-us/windowsazure/default.aspx
- Microsoft Azure Trust Center
  - http://azure.microsoft.com/en-us/support/trust-center/
- Microsoft Security Response Center [where Microsoft security vulnerabilities, including issues with Microsoft Azure, can be reported]
  - http://www.microsoft.com/security/msrc/default.aspx
  - Or via email to secure@microsoft.com.

8.1 Sources

- Azure Security Overview Training Module
  - http://www.microsoftvirtualacademy.com/tracks/windows-azure-security-overview
- Azure Security Review
  - http://blogs.msdn.com/b/buckwoody/archive/2011/08/02/windows-azure-security-review.aspx
- Crypto Primer: Understanding encryption, public/private key, signatures and certificates
  - http://blogs.msdn.com/b/plankytronixx/archive/2010/10/23/crypto-primer-understanding-encryption-public-private-key-signatures-and-certificates.aspx
- Field Note: Using Certificate-Based Encryption in Windows Azure Applications
  - http://blogs.msdn.com/b/windowsazure/archive/2011/09/07/field-note-using-certificate-based-encryption-in-windows-azure-applications.aspx
- Azure Security Guidance
  - http://www.windowsazure.com/en-us/documentation/articles/best-practices-security/
- Windows Azure Security Best Practices – Part 7: Tips, Tools, Coding Best Practices
  - http://blogs.msdn.com/b/usisvde/archive/2012/03/15/windows-azure-security-best-practices-part-7-tips-tools-coding-best-practices.aspx
- Should All Data Be Encrypted By Default?
  - http://blogs.msdn.com/b/buckwoody/archive/2011/08/09/should-all-data-be-encrypted-by-default.aspx
- Crypto Primer: How does SSL work?
  - http://blogs.msdn.com/b/plankytronixx/archive/2010/10/28/crypto-primer-how-does-ssl-work.aspx
- Windows Azure Data Security (Cleansing and Leakage)
  - http://blogs.msdn.com/b/walterm/archive/2012/02/01/windows-azure-data-cleansing-and-leakage.aspx
- Crypto Services and Data Security in Microsoft Azure
  - http://msdn.microsoft.com/en-us/magazine/ee291586.aspx
- Microsoft Rights Management Services (RMS)

    - http://www.microsoft.com/rms
- Deploying Highly Available and Secure Cloud Solutions
    - http://Aka.ms/avail
- 10 Things to know about Azure Security
    - http://technet.microsoft.com/en-us/cloud/gg663906.aspx
- Windows Azure Security Essentials
    - http://technet.microsoft.com/en-us/gg621084.aspx
- Azure SQL Database and SQL Server - Performance and Scalability Compared and Contrasted
    - http://msdn.microsoft.com/en-us/library/windowsazure/jj879332.aspx
- Azure Table Storage and Windows Azure SQL Database - Compared and Contrasted
    - http://msdn.microsoft.com/en-us/library/windowsazure/jj553018.aspx
- Data Series: Exploring Windows Azure Drives, Disks, and Images
    - http://azure.microsoft.com/blog/2012/06/27/data-series-exploring-windows-azure-drives-disks-and-images/
- Authenticating Access to Your Azure Storage Account
    - http://msdn.microsoft.com/en-us/library/hh225339.aspx
- How to: Implement Role Based Access Control (RBAC) in a Claims-Aware ASP.NET Application Using WIF and ACS
    - http://msdn.microsoft.com/en-us/library/gg185914.aspx
- Understanding the Temporary Drive on Azure Virtual Machines
    - http://blogs.msdn.com/b/wats/archive/2013/12/07/understanding-the-temporary-drive-on-windows-azure-virtual-machines.aspx
- RMS Boot Camp
    - http://curah.microsoft.com/56313/boot-camp-for-windows-azure-rights-management-rms