

Adding KMS to Existing NetBackup Deduplication Disk Pools

Key Management Server Overview

An important advantage of Key Management Servers (KMS) is that the encryption keys are not stored with the data. NetBackup Deduplication, also known as Media Server Deduplication Pools (MSDPs), added support for KMS starting at NetBackup 8.0, with an option in the Storage Server configuration wizard. For best results, plan deployments to include KMS at the time of initial configuration.

This document will review how to update the data at rest encryption for existing unencrypted data stored on the NetBackup deduplication disk pool. Veritas cryptographic modules, including those used in NetBackup deduplication, use AES-256 ciphers and SHA-2 hashing, and are registered with the National Institute of Standards and Technology (NIST) as Federal Information Processing Standards (FIPS) 140-2 compliant when combined with KMS.

KMS Considerations

When adding cloud tiers to NetBackup deduplication, a best practice is to set encryption and integrate with KMS at the time of creation according to the configuration guidelines of the cloud service provider (CSP). If encryption is added to the cloud logical storage unit (LSU) after initial configuration and after data has been written to the MSDP cloud tier, there may be inconsistent results. Follow the CSP guidance on encryption configuration.

AdvancedDisk encryption using KMS configuration is only possible during creation and further only from the command line interface (CLI). The Key Group for AdvancedDisk_crypt has specific naming requirements, while MSDP with KMS does not have any Key Group naming requirements. AdvancedDisk with KMS encryption configuration procedures can be referenced in the AdvancedDisk Storage Solutions Guide.

Choosing a KMS Solution

NetBackup KMS (NBKMS) is an additional role that can be added to the NetBackup primary server. For NBKMS, the keys are stored on the primary server and encrypted with the host master key and key protection key. NetBackup can also integrate with one or more external KMS using supported versions of key management interoperability protocol (KMIP). When linked to the external KMS server, the NetBackup primary server will need to query the encryption keys every 10 minutes and respond to the storage server when they are requested, requiring a reliable network connection to the KMS. This interval means that new keys will not be used immediately. All the known keys from its key group are cached in memory on the storage server while deduplication services are running, enabling all read and write operations to proceed, even if the connection to the KMS is broken. Protecting the KMS data source is also critical, and customers should follow all vendor-recommended procedures to ensure resiliency for the encryption solution. When using NBKMS, the data protection strategy can be referenced in the NetBackup Security and Encryption Guide, or later in this document.

NetBackup KMS and external KMS solutions can be used together. The key group is a logical grouping of keys and can be spread across multiple KMS servers, requiring the use of prioritization among the KMS server configurations within NetBackup. The highest priority KMS server with an active key for the key group will be used for encryption. It is possible that during the lifecycle of the data, the keys may need to be rotated, or external KMS may change or require migration. In the case of a configuration change, follow the supported procedures to add new key groups and keys. In the case of migration, add the new KMS to the configuration, and set the proper priority. KMS vendors may or may not support migrating keys between servers, and this is outside of the scope of NetBackup, so be sure to check before making significant changes. Each key has a unique ID, which is captured during the encryption process and must stay the same for the lifecycle of the data.

NetBackup Deduplication Data and Encryption

To enable encryption for MSDP cloud LSUs (those volumes other than the PureDiskVolume), KMS is required. For best results, configure KMS prior to adding an MSDP cloud LSU, and specify the correct key group and KMS name. If encryption with KMS is added after initial creation, only new data will be encrypted.

The segment objects (SO) are data fingerprints that hold the NetBackup deduplication encryption key. The SO encryption key is further encrypted by KMS. During a replication between two NetBackup deduplication storage servers, there are several encryption scenarios to consider when enabling encryption with KMS.

First, consider that not all data is transmitted during replication or optimized duplication, limiting the transmitted data to only those unique pieces of data that do not already exist in a deduplicated manner on the target storage.

In one scenario, the source has only MSDP encryption enabled, with no KMS. Data in transit stays encrypted regardless of the target MSDP encryption setting. If the target storage server does not have MSDP encryption enabled, the data is still sent encrypted to the target MSDP.

In a second scenario, the source has MSDP encryption with KMS enabled. In this modified scenario, KMS is used to encrypt the source SO encryption key. During transmission, the SO encryption key is encrypted with a session-based symmetric key shared by the source and target, and the data itself stays encrypted when transmitted to another storage server. Additionally, the KMS keys do not leave the environment. If KMS is desired at the source and target, a KMS solution is needed for each storage server.

To check encryption on specific data, reference MSDP encryption.

The following procedures are the same as preparing KMS prior to deploying NetBackup Deduplication. A KMS database can be set up at any time without disrupting NetBackup operations, allowing plenty of lead time before integration with NetBackup deduplication.

NetBackup KMS

First, authenticate via the CLI as a user with NetBackup administrator privileges:

Linux/UNIX/Appliance CLI User	Windows
<pre># /usr/opensv/netbackup/bin/</pre>	<pre># [install path]\Veritas\NetBackup\bin\</pre>
<pre>bpnbat -login -logintype WEB</pre>	

To initialize NetBackup KMS, create an empty KMS database on the NetBackup primary server and set the ID and passphrases when prompted for the host master key (HMK) and key protection key (KPK):

Linux/UNIX/Appliance CLI User	Windows
<pre># /usr/opensv/netbackup/bin/</pre>	<pre># [install path]\Veritas\NetBackup\bin\</pre>
<pre>nbkms -createemptydb</pre>	

Next, start the `nbkms` service from the CLI:

```
nbkms
```

NetBackup KMS Keys and Key Groups for MSDP

Keys are associated with key groups. For most disk encryption procedures, each disk pool will have a single key group with an active key for the PureDisk storage server. For KMS with any disk solutions, do not deprecate or terminate keys. New keys can be added, which will inactivate the current key, but the older keys must be retained for restores in an inactive state.

Create Key group:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/admincmd/	# [install path]\Veritas\NetBackup\bin\admincmd\ nbkmsutil -createkg -kgname {key group name}

Create one active key and provide a passphrase; please note, all supplied values are case-sensitive.

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/admincmd/	# [install path]\Veritas\NetBackup\bin\admincmd\ nbkmsutil -createkey -keyname {key name} -kgname {specifiy the same key group name} -activate [-desc "free text to describe this key"]

Verify Key group and key creation; please note, all supplied values are case-sensitive:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/admincmd/	# [install path]\Veritas\NetBackup\bin\admincmd\ nbkmsutil -listkeys -all

It is important to save this output to a secure location for the potential of key recovery, but this act of prevention is secondary to ensuring that there is a non-encrypted backup of the KMS database itself, which will be discussed later.

Ensure KMS service is re-discovered, to use the newly created KMS objects:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/	# [install path]\Veritas\NetBackup\bin\ nbkmscmd -discoverNBKMS -autodiscover

External KMS

Review the NetBackup compatibility guides for external KMS vendors. Interoperability is dependent on the vendor using key management interoperability protocol (KMIP). NetBackup uses a security certificate that the KMS server trusts for all KMIP communication. This certificate requires the common name (CN) of the NetBackup primary server to be specified, and the KMS server must have a user entity to enable permissions for the NetBackup primary server.

First, authenticate via the CLI as a user with NetBackup administrator privileges:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/	# [install path]\Veritas\NetBackup\bin\ bpnbat -login -logintype WEB

Validate and check compatibility for the external KMS server with the two commands below:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/goodies/	# [install path]\Veritas\NetBackup\bin\goodies\ nbkmsutil -listkeys -all

```
nbkmiputil -kmsServer {kms_server_name} -port {port} -certPath {cert_path} -privateKeyPath {private_key_path} -trustStorePath {trust_store_path} -validate
```

```
nbkmiputil -kmsServer {kms_server_name} -port {port} -certPath {cert_path} -privateKeyPath {private_key_path} -trustStorePath {trust_store_path} -ekmsCheckCompat
```

When the validation and compatibility are verified, continue to add the external KMS server to the NetBackup primary server by creating credentials to access the external KMS server.

Linux/UNIX/Appliance CLI User

Windows

```
# /usr/opensv/netbackup/bin/
```

```
# [install path]\Veritas\NetBackup\bin\
```

```
nbkmscmd - -configureCredential -credName {credential_name} -certPath {certificate_file_path} -privateKeyPath {private_key_file_path} -trustStorePath {CA_certificate_file_path} [-passphrasePath {private_key_passphrase_file_path}] [-crlCheckLevel LEAF | CHAIN | DISABLE]
```

Validate these credentials by querying them within NetBackup:

```
nbkmscmd -listCredential
```

Next, add the external KMS server:

```
nbkmscmd -configureKMS -name {configuration_name} -type KMIP -port {port_to_connect_to_external_KMS_server} -kmsServerName {network_name_of_external_KMS_server} -credId {credential_ID} | -credName {credential_name} -enabledForBackup 1
```

Lastly, validate the KMS configuration and securely store this with other essential information about the environment:

```
nbkmscmd -listKMSConfig
```

External KMS (EKMS) Keys and Key Groups for MSDP

Using KMIP, NetBackup can discover existing keys on the EKMS, or else create new keys. When associating with pre-existing keys, be sure to assign the `x-application` and `x-keygroup` attributes to the desired keys. Set `x-application` with a value of `NetBackup` and set `x-keygroup` with a value according to the desired key group name that will be referenced later. NetBackup can only use keys with both of these custom attributes, and the values are case-sensitive. For KMS with any disk solutions, do not deprecate or terminate keys. New keys can be added, which will inactivate the current key, but the older keys must be retained for restores in an inactive state.

Not all KMS vendors support changing these attributes from a graphical user interface (GUI), but the KMIP allows NetBackup to interface and perform these actions:

Linux/UNIX/Appliance CLI User

Windows

```
# /usr/opensv/netbackup/bin/goodies/
```

```
# [install path]\Veritas\NetBackup\bin\goodies\
```

```
nbkmiputil -kmsServer {kms_server_name} -port 5696 -certPath {cert_path} -privateKeyPath {private_key_path} -trustStorePath {caCertificatePath} -setAttribute -attributeName {attributeName} -attributeValue {attributeVal}
```

Create a key associated with the same x-keygroup value. This command automatically creates a key in the active state:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/	# [install_path]\Veritas\NetBackup\bin\
nbkmscmd -createkey -name {configuration_name} -keyGroupName {keygroup_name} -keyName {key_name} [-comment comments]	

Changes to external keys can take approximately 10 minutes to update due to caching behavior. Manually clear the cache with the following command:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/	# [install_path]\Veritas\NetBackup\bin\
bpclntcmd -clear_host_cache	

Updating the NetBackup Deduplication Storage Server

Now that a valid key group and key exist, the NetBackup deduplication configuration can be updated to use these new entities. On the NetBackup deduplication storage server, create a local file containing the following lines of text. Edit the values noted below in green with the values from your environment. For NBKMS, use `kmsservername` as the primary server name and specify the key group that you wish to use. Otherwise, specify the KMS host name. Configure and add KMS encryption before adding any cloud tiers to MSDP. The values must be enclosed within quotes and are case-sensitive:

```
v7.5 "operation" "set-local-lsu-kms-property" string
v7.5 "encryption" "1" string
v7.5 "kmsenabled" "1" string
v7.5 "kmsservertype" "0" string
v7.5 "kmsservername" "{KMS Host Name}" string
v7.5 "keygroupname" "{KMS Key Group Name}" string
```

To push the new KMS configuration for an MSDP cloud LSU, populate a separate text file with the appropriate key group and KMS server information, as follows:

```
v7.5 "operation" "update-lsu-cloud" string
v7.5 "lsuName" "the name of the cloud LSU" string
v7.5 "lsuKmsEnable" "YES" string
v7.5 "lsuKmsServerName" "{KMS Host Name}" string
v7.5 "lsuKmsKeyGroupName" "{KMS Key Group Name}" string
```

All encrypted LSUs in one storage server must use the same `keygroupname` and `kmsservername`. Confirm with the CSP for any additional encryption considerations.

Next, apply the changes from the above file to the NetBackup catalog using the `nbdevconfig` command:

Linux/UNIX/Appliance CLI User	Windows
# /usr/opensv/netbackup/bin/admincmd/	# [install_path]\Veritas\NetBackup\bin\admincmd\

```
nbdevconfig -setconfig -storage_server {storage server name} -stype PureDisk -configlist {path and file name from above steps}
```

To ensure that KMS will be active for all related components, such as universal shares or instant ccess mount points, it is important to restart all of the NetBackup services on the NetBackup deduplication storage server, specifically the vpfs_mounts processes.

Validating NetBackup Deduplication Encryption Settings

To remove any delays between configuring encryption and implementing encryption on new jobs, be sure to restart NetBackup services on the storage server after completing the configuration steps.

To confirm the desired encryption settings are in place, execute the following commands on the storage server:

Linux/UNIX/Appliance CLI User

Windows

```
# /usr/opensv/pdde/pdcr/bin/
```

```
# [install path]\Veritas\pdde\
```

```
crcontrol -getmode
```

Expected output:

```
Mode : GET=Yes PUT=Yes Deref=Yes SYSTEM=Yes STORAGED=Yes REROUTE=No COMPACTD=Yes FIPS=No KMS=Yes
```

```
keydictutil -list
```

Expected output:

```
KMS type: (0)
```

```
KMS Server Name [{KMS host name}].
```

```
KMS Key Group Name [{storage server name}:{MSDP key name}].
```

```
Active key index 2
```

```
Key 1: {hash1}
```

```
//additional keys may be present in some situations, such as manual key rotation//
```

```
dsid: 0 encryption: Yes kms: Yes
```

```
key group name: {storage server name}:{MSDP key name}
```

```
kms server name: {KMS host name}
```

```
readonly: No
```

In both outputs, observe that KMS is affirmed.

The MSDP encryption crawler will only encrypt previously unencrypted data on the local PureDiskVolume, and has no impact on deduplicated data tiered to cloud-based LSUs. The encryption crawler for NetBackup deduplication is included with NetBackup 10 and later, and is not enabled by default. The default Graceful mode will run in the background, and pauses if there are any active jobs; compared with Aggressive mode, where the process runs regardless of the storage server load. Both modes have tunable options, and start by encrypting the newest data first.

Before enabling the encryption crawler, be sure that the encryption configuration validation is complete.

To enable the encryption crawler:

Linux/UNIX/Appliance CLI User

Windows

```
# /usr/opensv/pdde/pdcr/bin/
```

```
# [install path]\Veritas\pdde\
```

```
crcontrol -enconverton
```

To check the progress:

```
crcontrol -enconvertstate 2
```

Example output:

```
**** Data Encryption Conversion ****
```

```
Status      : ON
Level       : Graceful (1)
Busy        : Yes
Max Group ID : 10
Current Group ID : 4
Current Container ID : 4331
Containers Estimated : 3211
Containers Scanned : 985
Containers Converted : 984
Containers Skipped : 0
Data Size Scanned : 61.2 GiB
Data Size Converted : 61.1 GiB
Progress     : 30%
Conversion Ratio : 99%
```

```
***** Mount Points Information *****
```

```
===== Mount point 1 =====
```

```
Path        : /msdp/data
Status      : ON
Level       : Graceful (1)
Busy        : Yes
Max Group ID : 10
Current Group ID : 4
Current Container ID : 4331
Containers Estimated : 3211
Containers Scanned : 985
Containers Converted : 984
Containers Skipped : 0
Data Size Scanned : 61.2 GiB
Data Size Converted : 61.1 GiB
Progress     : 30%
Conversion Ratio : 99%
```

The conversion processing time will vary depending on the mode, the amount of data to encrypt, storage server load, and system resources.

Protecting the KMS Database

For external KMS hosts, follow the vendor's documentation, but do not assume that the keys are protected automatically. In the case of a lost KMS host, restoration is critical for restores. If NetBackup is used to protect the external KMS data, do not store it on the KMS-enabled storage units.

The NBKMS database is made up of three files, KMS_DATA.dat, KMS_HMKF.dat, and KMS_KPKF.dat, found on the NetBackup primary server. These files are not protected by default in a NetBackup catalog backup and must be protected by a policy that is not using a KMS encrypted storage unit. Do not store the critical KMS information in a storage unit only accessible with an active KMS server, since this will result in an unrecoverable situation.

There are several methods to protect NBKMS, but the most direct method is a file-level backup of the KMS directory. NBKMS backups should occur on the same schedule as full catalog backups; otherwise, protect these files monthly and after any significant changes to the keys.

Path to NBKMS files to protect:

Linux/UNIX/NB Appliance	Windows
# /usr/opensv/kms/*	# [install path]\Veritas\kms*

Protection can be accomplished with a file-level backup when the KMS configuration is marked read-only with the command:

Linux/UNIX/NB Appliance	Windows
# /usr/opensv/netbackup/bin/admincmd/ nbkmsutil -quiescedb	# [install path]\Veritas\NetBackup\bin\admincmd\ nbkmsutil -quiescedb

When the backup is complete and NBKMS is ready to return to read and write, execute:

```
nbkmsutil -unquiescedb
```

Alternatively, alter the configuration to include KMS in the NetBackup catalog backup. Set the flag below on the primary server to include KMS data in the encrypted DRPKG file:

Linux/UNIX/NB Appliance	Windows
# /usr/opensv/netbackup/bin/admincmd/ echo KMS_CONFIG_IN_CATALOG_BKUP=1 bpsetconfig	# [install path]\Veritas\NetBackup\bin\admincmd\ echo KMS_CONFIG_IN_CATALOG_BKUP=1 bpsetconfig

Always save the passphrases and IDs from the initial NBKMS setup to a secure location for reference during recovery of the KMS database. Always save the key tag and salt when creating new KMS keys. With this information, more recovery options are available.

For other methods of protecting and recovering NBKMS, see the NetBackup Encryption and Security Guide.

Conclusion

While it is best practice to select encryption options at the time of NetBackup deduplication (MSDP) storage server creation, this is the method to add KMS to existing pools. MSDP encryption is enhanced by integrating KMS, separating the encryption key from storage. Data encrypted with KMS is additionally noted in the deduplication database, and that key will be used for encryption during write requests and decryption during read requests. For consistent encryption results for optimized duplication and replication workflows, enable encryption on all NetBackup deduplication disk pools, with a unique KMS key for each disk pool.

For more information about checking the encryption status of your deduplicated data, refer to the Veritas Support knowledge base article about MSDP encryption.

About Veritas

Veritas Technologies is a leader in multi-cloud data management. Over 80,000 customers—including 95 percent of the Fortune 100—rely on Veritas to help ensure the protection, recoverability, and compliance of their data. Veritas has a reputation for reliability at scale, which delivers the resilience its customers need against the disruptions threatened by cyberattacks, like ransomware. No other vendor is able to match the ability of Veritas to execute, with support for 800+ data sources, 100+ operating systems, 1,400+ storage targets, and 60+ clouds through a single, unified approach. Powered by Cloud Scale Technology, Veritas is delivering today on its strategy for Autonomous Data Management that reduces operational overhead while delivering greater value. Learn more at www.veritas.com. Follow us on Twitter at [@veritastechllc](https://twitter.com/veritastechllc).

VERITAS™

2625 Augustine Drive
Santa Clara, CA 95054
+1 (866) 837 4827
veritas.com

For global contact
information visit:
veritas.com/company/contact