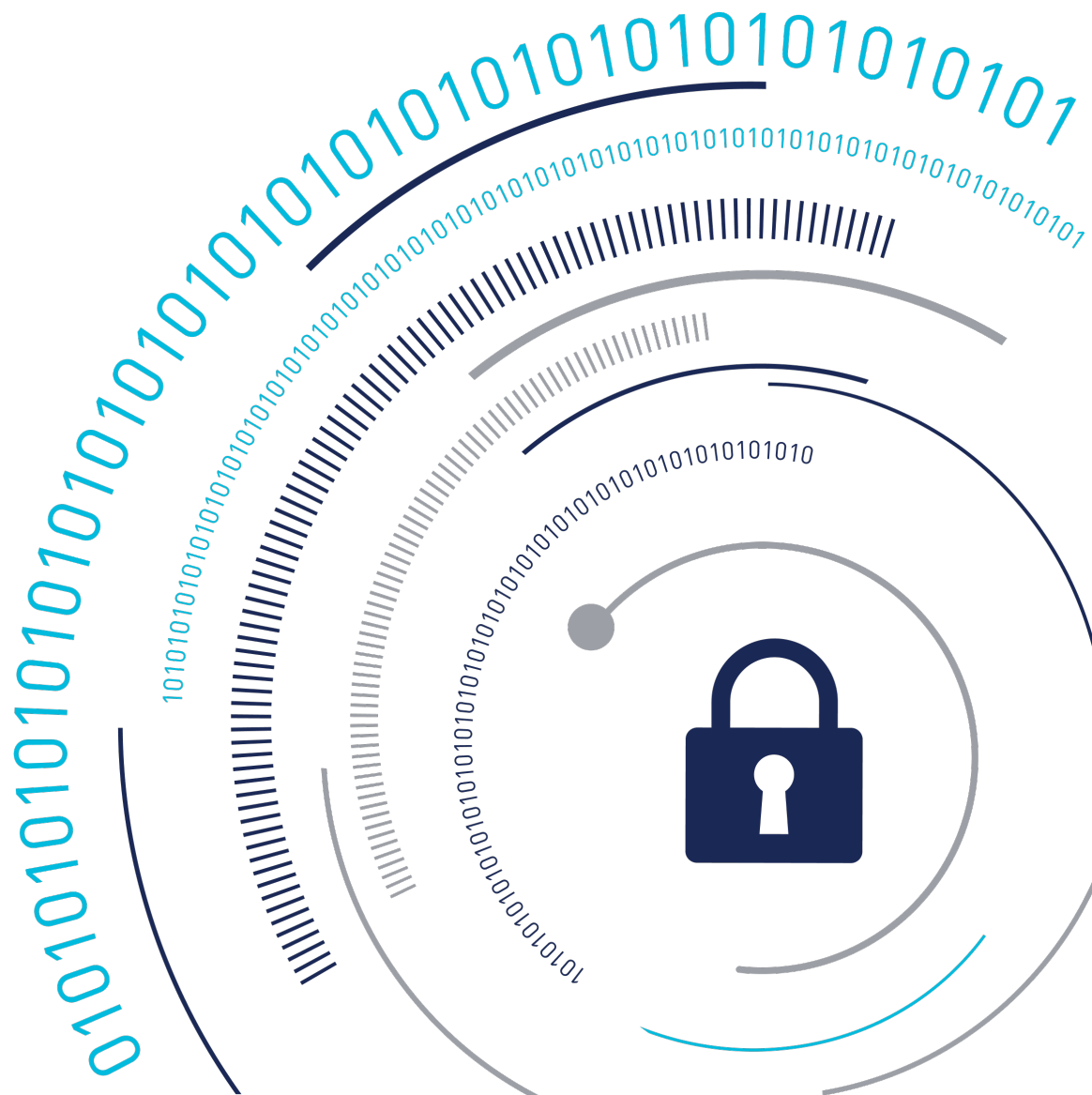


THALES

Live Data Transformation User Guide

FOR CTE V7.6.0



Live Data Transformation User Guide

- [Introduction to CTE-LDT](#)
- [Getting Started](#)
- [Installing CTE-LDT](#)
- [Using CTE-LDT](#)
- [CTE-LDT Administration](#)
- [Using CTE-LDT with CIFS/NFS shares](#)
- [Troubleshooting CTE-LDT](#)

Note

File Name Change

The metadata storage (MDS) file name has changed from `::vorm_mds::` to `__vorm_mds__`. GuardPoints will automatically rename the MDS file when guarding with a CTE 7.6.0 agent. LDT NFS GuardPoints will rename once all of the hosts, with the same share guarded, have been upgraded.

Introduction to CTE-LDT

This section contains the following topics:

- [Overview of CTE-LDT](#)
- [Keys in CTE-LDT \(Versioned Keys\)](#)
- [CTE-LDT Policies](#)
- [CTE-LDT Runtime Flow](#)
- [CTE-LDT Administrator Roles](#)
- [Resiliency](#)

Overview of CTE-LDT

CipherTrust Transparent Encryption - Live Data Transformation (CTE-LDT) is an optional, separately licensed feature of CipherTrust Transparent Encryption (CTE). With CTE-LDT, after enabling a GuardPoint, a Administrator can encrypt, or rekey, GuardPoint data without blocking user or application access to the data. In CTE-LDT, *rekey* means decrypting data with the current cryptographic key and then encrypting it with a new cryptographic key. The concept of rekey, and how CTE-LDT rekeys data, is described in this document.

After enabling GuardPoints, CTE-LDT performs *initial* encryption or rekeying in the background, unnoticed by users. The data stays live and available. This accelerates CTE deployments and eliminates the need to block application and user access to data during encryption or rekey operations, which can seriously inconvenience users and affect operational efficiency.

With CTE-LDT, the Administrator can create a single CTE-LDT policy for both initial encryption and subsequent rekeying. The same policy applies to production access and security rules without restricting user or application access to data. Applications have continuity of access to GuardPoint data during CTE-LDT.

Warning

To prevent data loss or corruption, you must stop all applications and users that are accessing files inside a GuardPoint before enabling a Live Data Transformation encryption policy for that GuardPoint. Terminating the applications closes all files that are currently being accessed inside the GuardPoint.

Unlike non-Live Data Transformation policies, however, you do not need to keep the GuardPoint offline while data transformation takes place. Instead, you can restart all applications as soon as the GuardPoint has been applied to the host, and CTE will perform the data encryption in the background. This is the only application service downtime required when using CTE-LDT.

Note

While CTE-LDT is supported on platforms with SELinux enabled in enforcing mode, you may run into interoperability issues with certain SELinux policies that may be enforcing rules against CTE access. If you experience issues running CTE with SELinux enabled, contact your system administrator for assistance. Thales Technical Support will recommend that you disable or change the mode to permissive mode to rule out SELinux when investigating reported issues.

Use Cases

This section provides a summary of typical uses for CTE-LDT. The concepts mentioned in this section are described in more detail throughout the rest of this guide.

1. Encrypt unprotected data.

When protecting files in a directory, you must encrypt them. This process is called *initial* data encryption.

2. Convert non-CTE-LDT GuardPoints to CTE-LDT GuardPoints.

Use when you have existing GuardPoints that are protected with policies created before you started using CTE-LDT.

3. Rekey process.

Changing the key from one version to another version of the same key provides more security. Using CTE-LDT, you can change the encryption keys to more secure keys.

4. Transform the encrypted data to clear data.

Keys in CTE-LDT (Versioned Keys)

CTE-LDT uses *versioned* keys. Each version of a particular versioned key has the same key name and encryption algorithm, but its own unique cryptographic material. That means that data encrypted with version 3 of a key named `LDT-Key` *cannot* be decrypted by any other version of `LDT-Key`, even though the key name remains the same.

Versioning allows you to add one key to your CTE-LDT encryption policy and then use different versions of that same key to periodically re-encrypt your data over time. CTE-LDT uses the new key material to transform data to the new key version, as part of the same Live Data Transformation policy that also protects the data. The process of re-encrypting data with a new version of the existing key is called *Key Rotation*.

In CipherTrust Manager, you can create a versioned key and then add that key to one or more Live Data Transformation policies. When you use CipherTrust Manager to create a new version of the key, CipherTrust Manager automatically pushes the new key version to any CTE clients associated with the Live Data Transformation policies that contain the key. As soon as the CTE Agent receives the new key, it begins transforming the data to the new key version in the background.

You can also create a Key Rotation Schedule in CipherTrust Manager that automatically rotates your keys periodically. When the scheduled rekey date is reached, CipherTrust Manager automatically creates a new key version and pushes it to any CTE clients that are associated with any policies that include the key. When CTE receives the new key version, it automatically starts the rekey process on the affected CTE-LDT GuardPoints.

Rekey | Key Rotation

In CTE-LDT, *rekeying* or *key rotation* means decrypting the data with a previous version of the key and re-encrypting it with a new version of the key. CTE-LDT allows users and applications to access data while CTE-LDT is rekeying the data. Rotating the key and re-encrypting the GuardPoint data with the new version of the key helps to maintain a high level of data security.

Most often, the rekey happens automatically based the Key Schedule defied in CipherTrust Manager, but you can also generate a new version of the key whenever you want to rekey the GuardPoints associated with that key.

CTE-LDT Policies

In CTE-LDT, you define a single policy for initial data encryption and subsequent rekeying. The policy specifies:

- **Current key**

Associated with data that you want to protect using CTE-LDT. This is either a non-versioned key from an earlier policy, or `clear_key`, which means that the data is not currently encrypted.

- **Transformation key**

The versioned key that CTE-LDT applies to transform the data from the key used for initial data transformation. When the transformation key rotates, it transforms the data from a previous version of the transformation key to a new version.

Note

Transformation key and versioned key are used interchangeably throughout this document.

As soon as CTE-LDT applies the policy to a GuardPoint and enables protection for it, CTE-LDT triggers an initial transformation from the current key to the transformation key.

When the transformation key expires, it generates the next version of the versioned key with new cryptographic material. The CipherTrust Manager then pushes the policy to the hosts. The policy now contains the new version of the key. This initiates a rekey process on the GuardPoint to transform data to the new version of the transformation key specified in the policy.

Users and applications can continue accessing data without any interruption during initial encryption and subsequent key transformations.

Note

During CTE-LDT policy creation, you must use the Apply Key effect in your policy. If you do not, then end users can see the clear text data until the file is transformed.

CTE-LDT Runtime Flow

This section presents an overview of how CTE-LDT works and what to expect when CTE-LDT is enabled and running in your environment. All of the tasks mentioned here are described in more detail later in this chapter.

First, the administrator completes CTE-LDT setup:

1. Upload the CTE-LDT license on the CipherTrust Manager.
2. Register CTE-LDT hosts with the CipherTrust Manager and be sure that the hosts are licensed for CTE-LDT.
3. Create one or more versioned keys.
4. Optionally create a Key Rotation Schedule or add your new keys to an existing schedule.
5. Define Live Data Transformation policies which use the versioned key(s) and contain rules governing CTE-LDT operations.
6. Optionally provide Quality of Service (QoS) settings for the CTE-LDT hosts. The QoS settings control the:
 - Window of time in which CTE-LDT operations are allowed to run.
 - Percentage of CPU resources that CTE-LDT can use, or the amount of data to transform according to the QoS setting per the Administrator.

Note

Configuring the QoS settings is highly recommended as a best practice.

When these items are set up, CTE-LDT is ready to transform and encrypt data by applying policies to GuardPoints for live initial transformation and subsequent rekeys, as well as enforcement of security rules.

CTE-LDT goes through the following phases after the keys and Live Data Transformation policies are defined:

1. Initial data transformation starts

CTE-LDT begins when an Live Data Transformation policy is first applied to a GuardPoint or an LDT key is automatically rotated through a key rotation schedule in CipherTrust Manager. When a client that uses the associated policy contacts CipherTrust Manager, CipherTrust Manager sends the new policy, or the the key rotation notification, to the client. (If the same versioned key is used in multiple policies, *all* of the clients associated with the policies that contain the key are notified when the key changes.)

2. New key version triggers a rekey on the affected GuardPoints

On each host/client , CTE determines which GuardPoints are using the key that has just rotated to a new version. CTE starts an CTE-LDT rekey on each of those GuardPoints.

On Windows, you must wait for the current key rotation process to finish before you can launch another rekey request. On Linux, if another rekey is already underway on that GuardPoint, the new rekey is queued for later execution. For details, see [Rotating Encryption Keys While a Rekey is in Progress \(Relaunch\)](#).

3. Scan for files

On each GuardPoint where CTE has started a rekey, CTE-LDT determines which files to transform. CTE-LDT takes inventory of files encrypted with earlier versions of the rotated key and makes a persistent list of the files for transformation.

The scan phase might be interrupted, such as by a host reboot. In this case, when the host reboots and the GuardPoint is enabled again, the scan operation starts over from the beginning.

4. Rekey/Key Rotation

- a.** Each file, from the persistent list of files, is decrypted using the old version of the key. The old key is applied to each file and then re-encrypted using the new version of the key. Note that new files created during the CTE-LDT process do not need to be rekeyed, as they inherit the new version of the key. Multiple files and multiple regions of files are rekeyed simultaneously.
- b.** The CTE-LDT extended attribute of each file is updated. (For more about extended attributes, see [CTE-LDT Metadata in Extended Attributes](#).)

- c. You can suspend and resume the CTE-LDT rekey operation manually, or through the QoS schedule. This manages the impact CTE-LDT has on other applications and processes.

If system errors occur during rekeying, such as IO errors or crashes, CTE-LDT can manage and recover from them after the system error is fixed.

5. Finish

When all of the required files in the GuardPoint have been rekeyed, the system and storage resources used by CTE-LDT are released, except for the storage required for the extended attributes.

CTE-LDT creates a rekey report, listing all of the files that were rekeyed. For more information, see [Obtaining a Rekey Report](#).

Upon completion of rekey, the Rekey Status in the GuardPoint Status window of the CipherTrust Manager Console shows Rekeyed.

CTE-LDT Administrator Roles

Using CTE-LDT requires coordination and collaboration between several different administrators. Security, system, application performance and backup schedules are critical factors that affect planning. It is very important for the **CipherTrust Manager Administrator** to coordinate with the **system administrator** and the **backup administrators**. The following table describes the roles and responsibilities for these administrators.

CTE-LDT Administrators

Role	Responsibility	Actions
CipherTrust Manager Administrator	Administers CipherTrust Manager	Coordinates with system and backup administrators. Creates security policies, key rotation schedules, and rules for CTE-LDT. Defines Quality of Service settings for CTE-LDT, applications, and backups.
System administrator	Administers servers on which CipherTrust Transparent Encryption (CTE) is deployed	Coordinates with the CipherTrust Manager Administrator to create Quality of Service schedule, taking

Role	Responsibility	Actions
		into consideration the backup schedules.
Backup administrator	Manages data backups for data encrypted by CTE	Coordinates with the CipherTrust Manager Administrator to create Quality of Service schedule, taking into consideration the backup schedules.

Resiliency

CipherTrust Transparent Encryption - Live Data Transformation is resilient to many user actions and system occurrences. Since it is designed to run periodically without intervention, it includes various features to provide this resilience.

Before CTE-LDT enables a GuardPoint, it checks for any inconsistencies in files that were undergoing rekey at the time when system operations were interrupted. If it finds any inconsistencies, CTE-LDT corrects them before it enables the GuardPoint. It should only take a few seconds to identify and correct any inconsistency. On Windows, this process is delayed until applications access the affected files.

During rekey, if an issue such as an I/O problem or system crash occurs, the resiliency features of CTE-LDT ensures the consistency of user data. Such issues can cause an interruption in the middle of a rekey operation. When system operations resume, CTE-LDT corrects the problems and then resumes rekeying.

Getting Started

This section contains the following topics:

- [Using CTE-LDT](#)
- [Backup/Restore](#)
- [Restrictions](#)

Using CTE-LDT

Note

- If you are new to CTE and CTE-LDT, read the CipherTrust Manager documentation first to familiarize yourself with the concepts of GuardPoints and Policies.
- Before installing CTE-LDT, see the [Linux Kernel Support](#) to verify that your version of Linux is supported.

The following list contains the steps for successfully setting up and using CTE-LDT.

1. Install the CTE-LDT License on the CipherTrust Manager.

CipherTrust Transparent Encryption - Live Data Transformation is a separately-licensed feature of CTE. Before you can use it, you have to install the license to activate it. CTE-LDT is licensed for a specific number of hosts.

For details about licensing, see the CipherTrust Manager documentation.

2. Install the CTE Agent and select the CTE-LDT feature during the install. For more information, see [Installing and Registering the CTE Agent Software on Linux](#) or [Installing and Registering the CTE Agent Software on Windows](#).

If the client is already registered with CipherTrust Manager, you can enable CTE-LDT on the client through the CipherTrust Manager Console. For more information, see [Enabling CTE-LDT on a Protected Host](#).

3. Create Versioned Keys.

CTE-LDT uses *versioned keys*. A versioned key rotates to the next version of the key generating new key material automatically without policy change. CTE-LDT encrypts data with keys that use encryption standards like AES-256. This allows data to be re-encrypted without users having to edit the policy.

For more information, see [Keys in CTE-LDT \(Versioned Keys\)](#) and [Creating and Viewing Versioned Keys](#).

4. Optionally create a Key Rotation Schedule that will automatically rotate the versioned keys periodically. For more information, see [Creating a Key Rotation Schedule](#).

5. Create CTE-LDT Policies.

CTE-LDT uses a single Live Data Transformation policy to address both initial encryption and subsequent rekeying. The same policy applies to production access and security rules without restricting user or application access to data. Applications have continuity of access to GuardPoint data during CTE-LDT. For more information, see [Creating CTE-LDT Policies](#).

6. Set QoS Settings.

QoS enables administrators to manage and control CTE-LDT impact to application workloads by monitoring and controlling the use of host system resources, such as memory or I/O utilization, during CipherTrust Transparent Encryption - Live Data Transformation. Administrators can also choose schedules for data transformation, or manually pause or resume transformation operations. For more information, see [Quality of Service](#).

7. Create GuardPoints and apply CTE-LDT policies to the GuardPoints.

A GuardPoint is a directory in the file system hierarchy, where its contents have a CipherTrust data protection policy applied to it. The CTE Agent intercepts any attempt to access anything in the GuardPoint and uses the policies obtained from the CipherTrust Manager to grant or deny the access attempt. Typically data copied into a GuardPoint is encrypted, and only authorized users can decrypt and use that GuardPoint data.

For more information, see [Creating a CTE-LDT GuardPoint](#).

Backup/Restore

Before you enable a GuardPoint with an CTE-LDT Policy, make sure that you back up your data.

For more information, see [Backup Requirements](#) and [Backing Up and Restoring CTE-LDT GuardPoints](#).

Restrictions

Remember the following restrictions when using CTE-LDT:

- CTE-LDT does not support nested GuardPoints, where a GuardPoint is contained inside another GuardPoint.
- For HA clusters, CTE-LDT only supports the Asymmetric (active/passive) configuration. CTE-LDT does not support Symmetric (active/active) configuration.

- If you want to create CTE-LDT GuardPoints on Linux NFS shares or Windows CIFS shares, you must register the host with CipherTrust Manager and add the host to an LDT communication group.

Windows Only Limitations

CTE-LDT supports GuardPoints on CIFS network shared directories with the following restrictions:

- You cannot guard both CIFS shares with CTE-LDT and local directories on the same host (even if the local directories use a Standard CTE policy). Instead, you must choose one or the other when you install the CTE Agent. In addition, if you want to upgrade an existing CTE Agent to support CTE-LDT CIFS share GuardPoints, you must first remove any existing GuardPoints, uninstall the CTE Agent, and then install the latest CTE Agent in its place.

For details, see [Installing and Registering the CTE Agent Software on Windows](#) and [Upgrading an Existing CTE Agent to Support NFS/CIFS Shares](#).

- Only unstructured data can be encrypted.
- If any files are opened exclusively by another application, CTE-LDT cannot rekey those files until the other applications have released the lock.
- If a backup is taken at the snapshot level, CTE metadata is also backed up. If a restore operation tries to restore CTE metadata, CTE agent does not allow this operation and the metadata restore fails. Do not restore the CTE metadata, or ignore the error from the restore utility, if the CTE metadata restore fails.
- CTE-LDT on a ReFS file system runs slowly because of limited support from the Extended Attributes on the ReFS file system.

Note

Customers running older versions of ReFS.sys on Windows Server 2012 R2 should be aware of the memory growth issue encountered by the Thales engineering team. This issue seems to occur only when CTE-LDT is running on a large number of files. As the system memory consumption by REFS file system increases, it can eventually make the system unresponsive. This issue does not occur with the recent versions of ReFS file system drivers available on Windows Server 2016. After consulting with Microsoft, they suggest that all customers migrate to Windows Server 2016 if they are using ReFS file.

Linux Only Limitations

- CTE-LDT does not support Linux auto-mounted file systems.
- CTE-LDT support is limited to `ext3`, `ext4`, and `XFS` file systems when `user_xattr` mount option is enabled.
- CTE-LDT does not support system hibernation (`pm-hibernate`) on Linux hosts where CTE-LDT is in use.
- You cannot use CTE-LDT and Docker container on the same host.
- You cannot use CTE-LDT and OpenShift container on the same host.

Installing CTE-LDT

This section contains the following topics:

- [System Requirements](#)
- [Installing the CTE-LDT License](#)
- [Installing and Registering CTE Agent Software on Linux](#)
- [Installing and Registering CTE Agent Software on Windows](#)
- [Setting the Linux Kernel Time Zone](#)
- [Enabling CTE-LDT on a Protected Host](#)

System Requirements

CipherTrust Transparent Encryption - Live Data Transformation (CTE-LDT) requires the following environment.

Note

See the latest [Compatibility Matrix for CTE Agent](#) for a list of CTE versions and supported operating systems.

CipherTrust Software Requirements

- CTE Agent version 7.0.0 or higher must be installed on every client that you want to protect.

- The CipherTrust Manager with which these clients are registered must be version 2.0.0 or higher.

CTE-LDT License Requirements

- One CTE-LDT license is required for *each* protected host.

Host System Requirements

Memory requirements: Both Linux and Windows require a minimum of 8 GB memory on each protected host.

Disk space requirements: CTE-LDT requires a specific amount of disk space in the file system or NFS/CIFS share for each GuardPoint, over and above the space required for the guarded files themselves. CTE-LDT uses the additional space to store CTE-LDT metadata.

- For Linux, to estimate the amount of free file system disk space required by CTE-LDT on a Linux host, use the `voradmin ldt space` command.
- For Windows, the typical minimum space requirement for a GuardPoint is the number of files in the GuardPoint multiplied by 4K, plus 256MB. To estimate the amount of free space required for a GuardPoint, use the `voradmin ldt space <guardpoint-name>` command.

Backup Requirements

- For **Windows**, GuardPoints in local file systems or CIFS shares, the application can use Volume Shadow Copy service (VSS) or roboCopy for backup and restore.
- For **Linux**, GuardPoints in local file systems, the backup application must have the capability to back up user-extended attributes. For example, Thales tested CTE-LDT with NetBackup. You can also use other applications that can back up user-extended attributes.

Backup Requirements for Linux GuardPoint backup on NFS

- You can only perform a backup or restore on the NFS client designated as the primary client for the GuardPoint. Use backup clients or tools such as tar, rsync, etc. on this client. You can backup or restore individual files, or subdirectories, within a GuardPoint namespace without an Apply Key Effect. However, this excludes MDS files and all CTE-LDT private files under `vorm_ldtprivspace`.

- Performing a backup or restore on a non-primary client is not supported and may lead to data and/or LDT metadata inconsistencies, in target files, in the backup image or in the restored files. Attempts to restore files on a non-primary client will be detected and rejected. Refer to section [Encrypted Backup and Restore](#) for additional information.
- File backup/restore can only be executed when the GuardPoint is rekeyed, or in a subsequent rekeying suspended state. File backup should **not** be executed during initial rekeying state.
- For **Linux**, GuardPoints on NFS shares, the normal procedure is to back up the NAS server that serves the NFS clients rather than backing up the individual NFS clients. Because backups on NAS servers are remote from a CTE perspective, it is extremely important that CTE is *not* performing any data transformation in any GuardPoint while the NAS server is being backed up. This includes both initial data transformation and automatic or manual data rekey.

Warning

Make sure that all live data transformation has completed on all NFS GuardPoints before you back up the NAS servers associated with the NFS shares on which those GuardPoints reside. You must also make sure that no automatic rekey tasks will start while the NAS backup is in progress.

Linux-Specific Requirements

- To use CTE-LDT on an ext3 or ext4 file system, the block size **must** be 4K. Run `dumpfs` to determine the block size of the ext3 or ext4 file systems before using CTE-LDT. This limitation does not apply to XFS file systems or NFS shares. You can use CTE-LDT on XFS with a block size of 1K or 2K.

Local File System Requirements

- For CTE-LDT to work properly on local file systems, the underlying file system must support and enable user-extended attributes. All of the file systems supported by CTE-LDT support these attributes. If you are using CTE-LDT with ext3 or ext4 mount points, you must explicitly enable the extended attribute mount option by editing `/etc/fstab` and adding the `user_xattr` mount option. In the other file systems supported by CTE-LDT, user extended attributes are enabled by default, so you do not have to explicitly enable them.

Example `/etc/fstab` entry for ext3 on Red Hat 6 or SLES:


```
/dev/sdb1 /disk2 ext3 defaults,user_xattr 0 0
```

Example `/etc/fstab` entry for ext4 on Red Hat 6 or SLES 12 (ext4 is not supported on SLES 11):

```
/dev/sdc1 /disk3 ext4 auto,users,user_xattr,exec 0 0
```

For more information about extended attributes, see [CTE-LDT Metadata in Extended Attributes](#).

NFS Share Requirements

- For GuardPoints on NFS shares, CTE-LDT embeds file specific LDT metadata in the beginning of each file during the initial data transformation phase. The size of each file becomes larger by 4096 bytes to accommodate this required metadata. Both the metadata and the file size increase are hidden from users and applications as long as the GuardPoint remains enabled.
- If a GuardPoint is disabled, the metadata remains in each file and both the metadata and the file size increase become visible to users and applications.

Note

Thales highly recommends that NFS shared directory be mounted with `sync` option.

Supported Applications in Linux

- For all of the supported operating systems for database applications, see the [Compatibility Matrix for CTE Agent](#).

Replication

- `rsync`
- Hardware/software based replication system

SAP HANA Fibre Channel Systems

- SAP HANA is compatible with CTE-LDT. See the [Compatibility Matrix for CTE Agent](#).

Windows-Specific Requirements

CIFS Share Requirements

- For GuardPoints on CIFS shares, CTE-LDT embeds file specific LDT metadata in beginning of each file during the initial data transformation phase. This metadata remains hidden to users and other applications as long as the GuardPoint is enabled.
- If a GuardPoint is disabled, the metadata remains in each file and becomes visible to users and other applications.
- The size of each file becomes larger by 4096 bytes to accommodate the required CTE-LDT metadata.

Installing the CTE-LDT License

CTE-LDT installs with the CTE Agent software, but it is a separately-licensed CTE feature. Before you can use it, you have to install the license on the CipherTrust Manager to activate CTE-LDT on any CTE clients. CTE-LDT is licensed for a specific number of clients.

1. Obtain an CTE-LDT license from Thales. You can purchase CTE-LDT along with the CipherTrust Manager software or you can add CTE-LDT later.
2. Install the license on the CipherTrust Manager.
For details, see the CipherTrust Manager documentation.

To confirm that the CTE-LDT license is in effect:

1. In the CipherTrust Manager Applications Page, open the **Admin Settings** application.
2. Click **Licensing** in the left-hand menu bar.
3. Expand the entry for **CTE - Live Data Transformation** in the License table. CipherTrust Manager shows the total number of client licenses in the **Total Clients** field and the total number of licenses used in the **Used Clients** field.

Installing and Registering the CTE Agent Software on Linux

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

Prerequisites

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

- CipherTrust Manager installed and configured. See [CipherTrust Manager Documentation](#) for more information.
- CipherTrust Manager must contain a Client Profile. See [Changing the Profile](#) for more information.
- CipherTrust Manager must contain a registration token. See [Creating a Registration Token](#).
- Optionally, the name of the host group you want this client to be a part of.
- CipherTrust Manager must contain an LDT Communication Group if you will use CTE to guard data over CIFS/NFS shares using LDT policies. See [Managing LDT Communication Groups](#) for more information.

Procedure

1. Log on to the host where you will install the CTE Agent as `root`. You cannot install the CTE Agent without `root` access.
2. Copy or mount the installation file to the host system. If necessary, make the file executable with the `chmod` command.
3. Install the CTE Agent. A typical installation uses the following syntax:

```
./vee-fs-<release>-<build>-<system>.bin
```

For example:

```
./vee-fs-7.3.0-135-rh8-x86_64.bin
```

To install the CTE Agent in a custom directory, use the `-d <custom-dir>` option. For example:

```
./vee-fs-7.3.0-135-rh8-x86_64.bin -d /home/my-cte-dir/
```

Note

If possible, Thales recommends that you use the default directory `/opt/vormetric.`

To view all installer options, use the `-h` parameter. For example:

```
./vee-fs-7.3.0-135-rh8-x86_64.bin -h
```

4. The Thales License Agreement displays. When prompted, type **Y** and press Enter to accept.

The install script installs the CTE Agent software in either `/opt/vormetric` or your custom installation directory and then prompts you about registering the CTE Agent with a key manager.

Welcome to the CipherTrust Transparent Encryption File System Agent Registration Program.

```
Agent Type: CipherTrust Transparent Encryption File System Agent
Agent Version: <Release.build-number>

In order to register with a CipherTrust Manager you need a valid
registration token from the CM.

Do you want to continue with agent registration? (Y/N) [Y]:
```

5. Type **N** and press Enter to end the installation procedure without registering the CTE Agent with either key manager.
6. Enter **Y** to continue with the registration process. The install script prompts you to enter the host name or IP address of the CipherTrust Manager with which you want to register CTE. For example:

Do you want to continue with agent registration? (Y/N) [Y]: Y

```
Please enter the primary key manager host name: 10.3.200.141
```

Note

The default communication port is 443. If you want to specify a different communication port, enter it with the primary key manager host name in the format: `<hostName>:<port#>`

You entered the host name 10.3.200.141

Is this host name correct? (Y/N) [Y]: Y

7. Enter the client host name when prompted.

```
Please enter the host name of this machine, or select from the following list.
```

```
[1] sys31186.qa.com
```

```
[2] 10.3.31.186
```

```
Enter a number, or type a different host name or IP address manually:
```

```
What is the name of this machine? [1]: 2
```

```
You selected "10.3.31.186".
```

8. Enter the CipherTrust Manager registration token, profile name, host group and host description. If you omit the profile name, CipherTrust Manager associates the default client profile with this client.

```
Please enter the registration token: 12345
```

```
Please enter the profile name for this host: My-Profile
```

```
Please enter the host group name for this host, if any:
```

```
Please enter a description for this host: RHEL7 system West Coast Datacenter
```

```
Token : 12345
```

```
Profile name : My-Profile
```

```
Host Group : (none)
```

```
Host description : RHEL7 system West Coast Datacenter
```

```
Are the above values correct? (Y/N) [Y]: Y
```

9. At the hardware association prompt, select whether you want to enable the hardware association feature to prevent cloning. The default is Y (enabled):

It is possible to associate this installation with the hardware of this machine. If selected, the agent will not contact the key manager or use any cryptographic keys if any of this machine's hardware is changed. This can be rectified by running this registration program again. Do you want to enable this functionality? (Y/N) [Y]: Y

10. At the LDT prompt, specify that you want this client to use CTE-LDT by typing **Y** and pressing Enter:

```
Do you want this host to have LDT support enabled on the server?  
(Y/N) [N]: Y
```

11. If you are planning to create GuardPoints on NFS shares, enter the name of the LDT Communication Group that this node will join.

```
Enter the LDT Communication Group name: LCG1
```

Warning

The registration token, profile name, client group name and LDT Communication Group name are case-sensitive. If any of these are entered incorrectly, the client registration will not succeed. If the registration fails, click Back in the installer and verify that the case is correct for all entries on this page.

12. At the Cloud Object Storage (COS) prompt, specify whether you want this client to use CTE COS.

```
Do you want to configure this host for Cloud Object Storage? (Y/N)  
[N]:
```

13. CTE finishes the installation and registration process.

```
Generating key pair for the kernel component...done.  
Extracting SECFS key  
Generating EC certificate signing request for the vmd...done.  
Signing certificate...done.  
Enrolling agent with service on 10.3.200.141...done.  
Successfully registered the CipherTrust Transparent Encryption CTE  
Agent with the CipherTrust Manager on 10.3.200.141.  
  
Installation success.
```

14. In CipherTrust Manager, change the client password using the manual password creation method. This password allows users to access encrypted data if the client is ever disconnected from the CipherTrust Manager. For details on changing the password, see the CipherTrust Manager documentation.

Installing and Registering the CTE Agent Software on Windows

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

Prerequisites

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

- CipherTrust Manager installed and configured. See [CipherTrust Manager Documentation](#) for more information.
- CipherTrust Manager must contain a Client Profile. See [Changing the Profile](#) for more information.
- CipherTrust Manager must contain a registration token. See [Creating a Registration Token](#).
- Optionally, the name of the host group you want this client to be a part of.

- CipherTrust Manager must contain an LDT Communication Group if you will use CTE to guard data over CIFS/NFS shares using LDT policies. See [Managing LDT Communication Groups](#) for more information.

Installation

1. Log on to the host as a Windows user with System Administrator privileges.
2. Copy the CTE installation file onto the Windows system.
3. Double-click the installation file. The InstallShield Wizard for CipherTrust Transparent Encryption opens.
4. Verify the version of CTE you are installing and click **Next**.
5. On the **License Agreement** page, accept the License Agreement and click **Next**.
6. On the **Destination Folder** page, click **Next** to accept the default folder or click **Change** to select a different folder. When you are done, click **Next**.

Note

- Thales recommends that you install CTE in the default installation directory, `c:\Program Files\Vormetric\DataSecurityExpert\agent\`
- You **must** install the CTE Agent on the same drive as Windows. For example, if Windows is installed on the `c:` drive, you must install the CTE Agent on the `c:` drive.

7. On the **Ready to Install** page, click **Install**. When the installation is finished, the **Install Shield Wizard Completed** window opens.

Registration

The following procedure describes how to register the CTE Agent after installation is complete.

Use the `register_host` utility to create certificate requests, exchange certificates between the CipherTrust Manager and the host, to enable ransomware detection, and to register CTE on the CipherTrust Manager. After the host is registered, if you selected to enable file system encryption, you can configure CTE, apply GuardPoints, or perform database backups. If you selected ransomware protection, you can apply the protection to volumes. Run the `register_host` utility in text mode on a terminal window.

Caution

The default host registration timeout is 10 minutes. If the host is unable to reach the CipherTrust Manager within the allotted period because of an extremely slow network connection, set the `REGISTER_HOST_TIMEOUT` environment variable to extend the registration timeout. The variable value is an integer expressed in seconds. You might also have to extend the default TCP timeout.

1. Log on to the host as a Windows user with administrative privileges.
2. Launch the CTE Registration Wizard using one of the following methods:
 - In the system tray, right click the CipherTrust Lock icon and select **Register Host**.
 - Run `...\register_host.exe -vmd -silent`.
3. Reboot the system. CTE automatically displays the registration wizard if CTE is not already registered.
4. In the Register Host dialog box, verify the host's machine name and click **Next**.
5. On the **Gathering agent information** page, select one or both of the following options and click **Next**.
 - **File System**: Allows you to protect and encrypt CTE files with policies
 - **Ransomware Protection**: Protects volumes from ransomware
6. On the **Gathering Key Manager information** page, enter the FQDN or IP address of the primary CipherTrust Manager.

The default communication port is 443. If you want to specify a different communication port, enter it with the primary key manager host name in the format: `:. For example:10.3.200.141:8445`

When you are done, click **Next**. CTE communicates with the selected CipherTrust Manager to validate what features have been licensed and are available to the CTE Agent.
7. On the **Gathering host information** page:

File System

- Specify the host name or IP address of the client. You can select the host name from the drop-down list or type it in the field.

- To prevent cloning, select **Enable Hardware Association**. For details, see [Hardware Association Feature](#).
- If you want to have CipherTrust Transparent Encryption - Live Data Transformation available on the client, select **Enable LDT Feature**. For details on CTE-LDT, see **CTE-Live Data Transformation with CipherTrust Manager**.
- If you want the node to be an [LDT AccessOnly](#) node, select **Enable LDT AccessOnly Feature**.

Ransomware

- Specify the host name or IP address of the client. You can select the host name from the drop-down list or type it in the field.
- To prevent cloning, select **Enable Hardware Association**. For details, see [Hardware Association Feature](#).

When you are done, click **Next**.

8. On the **Gathering registration information** page, enter the following:

File System

- **Registration token**: The registration token for the CipherTrust Manager with which you want to register this host.
- **Profile name**: The name of the profile that you want to associate with this host. This name must match exactly the name of the profile in the CipherTrust Manager. If you do not specify a profile name, the CipherTrust Manager associates the default client profile with this client.
- **Host group** (optional): The name of the client group to which the client will be added.
- **Host description** (optional): A user-defined description of the client. This description will be displayed in the CipherTrust Manager.
- **LDT Communication Group**: If you are planning on using LDT over CIFS/NFS on a CipherTrust Manager, enter the name of the LDT Communications Group that this node will join. See [Adding Clients to an LDT Communication Group](#) for more information.

Ransomware

- **Registration token:** The registration token for the CipherTrust Manager with which you want to register this host.
- **Profile name:** The name of the profile that you want to associate with this host. This name must match exactly the name of the profile in the CipherTrust Manager. If you do not specify a profile name, the CipherTrust Manager associates the default client profile with this client.
- **Host group** (optional): The name of the client group to which the client will be added.
- **Host description** (optional): A user-defined description of the client. This description will be displayed in the CipherTrust Manager.

Warning

The registration information is case-sensitive. If any of it is entered incorrectly, the client registration will not succeed. If the registration fails, click Back in the installer and verify that the case is correct for all entries on this page.

When you are done, click **Register**. CTE contacts the CipherTrust Manager and attempts to register the client with the specified options. The Register Host dialog box displays a message with the results of the registration request.

If the registration completed successfully, click **Finish**.

9. Restart the client to complete the installation process on the client.
10. After the host has rebooted, you can verify the installation by checking CTE processes:
 - a. In the system tray of the protected host, right-click the CipherTrust Lock icon.
 - b. Select **Status**. Review the information in the **Status** window to confirm that the correct CTE version is installed and registered.
 - If you are using CipherTrust Manager version 2.2 or later, you can now use CipherTrust Manager to administer CTE on the client.
If you are using CipherTrust Manager version 2.1 or earlier, change the client password using the manual password creation method. This password allows

users to access encrypted data if the client is ever disconnected from the CipherTrust Manager. For details on changing the password, see the CipherTrust Manager documentation.

Setting the Linux Kernel Time Zone

The Linux kernel contains an internal time structure that may or may not contain time zone information. On system configurations that do not contain time zone information, CTE-LDT stores and displays timestamps for rekey beginning and ending in UTC (Coordinated Universal Time) rather than the system's local time zone. If this occurs, the administrator can set the kernel's internal time zone to the local time zone if they desire timestamps in their local time zone.

To set the Linux Kernel time zone information, at boot time type:

```
hwclock --systz
```

The command sets the kernel's time zone to the local time zone and resets the System Time based on the current time zone.

Note

On systems that do not set the time zone by default, existing timestamps for completed rekeys remain in UTC, even if you run `hwclock --systz`. Only timestamps for new rekeys display the local time zone.

Enabling CTE-LDT on a Protected Host

When you install CTE on a host, you can enable CTE-LDT during the registration process as described in [Installing and Registering the CTE Agent Software on Linux](#). If you have already registered the host with CipherTrust Manager, you can enable CTE-LDT in the client entry in CipherTrust Manager.

Note

- The CTE-LDT license is valid for a certain number of clients. Once you reach this limit, you can either purchase additional CTE-LDT licenses, or reclaim a license by removing a CTE-LDT client. For details, see [Removing CTE-LDT and Security Encryption](#).
- You cannot disable CTE-LDT on a client once it has been enabled.

To enable CTE-LDT on an existing host:

1. In the CipherTrust Manager Applications Page, open the **CTE** application and click **Clients** in the left-hand menu bar.
2. Click on the name of the client on whom you want to enable CTE-LDT.
3. On the Client Details page, select the **Live Data Transformation** check box, then click **Apply** to enable CTE-LDT on the protected host.

Note

After CTE-LDT has been enabled for the host, the check box is greyed out because you cannot change this option once it has been set.

Using CTE-LDT

- [Creating and Viewing Versioned Keys](#)
- [Creating CTE-LDT Policies](#)
- [Quality of Service](#)
- [Creating a CTE-LDT GuardPoint](#)
- [Rotating Encryption Keys \(Rekey\)](#)
- [File System Operations](#)
- [Excluding Files or Directories from Rekey](#)
- [Using CTE-LDT with SAP HANA Fibre Channel Systems \(Linux Only\)](#)

Creating and Viewing Versioned Keys

For CTE-LDT, you create a versioned key that you can add to any number of Live Data Transformation policies. When you add a new key version, CipherTrust Manager pushes the new version to every CTE client that is associated with any of their Live Data Transformation policies that contain that key.

Use the **Keys & Access Management** application in the CipherTrust Manager Applications Page when creating or viewing versioned keys.

Caution

Make sure that all keys you create for CTE-LDT are Symmetric. CTE-LDT does not support asymmetric keys.

Creating a Versioned Key

1. Log into the CipherTrust Manager Console as an administrator.
2. From the Products page in the CipherTrust Manager Console, click **Keys** in the left hand pane.

Tip

To navigate to the Products page from anywhere in the CipherTrust Manager Console, click the App Switcher icon in the top left corner.

3. Above the Key table, click **Create a New Key**.
4. Enter a Key Name so that you will be able to find this key easily when you want to rotate it. For example, LDT-Key-1.

Note

- When CipherTrust Manager displays the list of available keys, you cannot filter the list by versioned vs. non-versioned. Therefore, it is especially important to name the key in such a way that you can easily search for it by name.
- Using a standard naming convention will also help if you want to create an automatic key rotation schedule for all your CTE-LDT keys.

5. In the **Key Metadata > Groups for Key sharing** section, do the following:

- In the **Search** box, type "cte".
- Add CTE Admins and CTE Clients to the key sharing groups by clicking the green **Add** button. The **Key Shared?** check box is automatically selected and the **Add** button changes to a **Remove** button.
- Below the Groups table, click the **CTE Key Properties** check box.

CipherTrust Manager displays the following options for CTE keys:

- **CTE Versioned:** Specifies that this is a versioned key. Make sure this option is selected.
- **Persistent on Client:** Specifies whether the key is stored in persistent memory on the client.

If this option is selected, the key is downloaded and stored (in an encrypted form) in persistent memory on the client.

If this option is *not* selected, the key is downloaded to non-persistent memory on the client. Every time the key is needed, the client retrieves it from the CipherTrust Manager. This is the default setting.
- **Encryption Mode:** Encryption mode of the key. Choose one of the following for CTE-LDT keys:
 - CBC
 - CBC-CS1

Encryption using CBC-CS1 keys is known as enhanced encryption. For details, see the *CTE Agent for Linux Advanced Configuration and Integration Guide* or the *CTE Agent for Windows Advanced Configuration and Integration Guide*.

Note

XTS encryption is not supported for protecting directory-based GuardPoints. Do not select XTS as your encryption mode.

Groups for Key sharing

Q cte

Group Names	Key Shared?
CTE Admins	<input checked="" type="checkbox"/>
CTE Clients	<input checked="" type="checkbox"/>

2 Groups 10 per page ▾

Set as "Versioned Key" for backwards compatibility

CTE Key Properties

CTE Versioned Persistent on Client

Encryption Mode
CBC CS1 ▾

- In the Key Behaviors section at the bottom of the page, clear the **Prevent this key from being exported** check box. If the key cannot be exported, the key will not appear in the keys list when you add the key rule to the policy. If you want the option to delete this key later on, clear the **Prevent this key from being deleted** check box. Use caution when deleting keys, as any data encrypted with that key will be inaccessible if you delete the encryption key.

Key Behaviors

Prevent this key from being deleted Prevent this key from being exported

Create Cancel

- Click **Create**. The new key appears in the Keys table.

Viewing Versioned Key Information

- Log into the CipherTrust Manager Console as an administrator.

2. Open the **Keys & Access Management** application.
3. Click the **Latest Version Only** check box to show only one entry for each versioned key.
4. Find the key in the Keys table.

Tip

You can filter the list by key name using wildcard searches. For example, if you know that the key name contains "LDT", you can search for `*ldt*`. The search is *not* case sensitive.

If the **Version** for a key is 0, then this is the first version of the key and no key rotation has taken place. If the **Version** is greater than 0, the key has been rotated.

5. To find all versions of a key, click the name of the key in the **Key Name** column. CipherTrust Manager displays the ID, version number, state, and date created information for each version of the key.

Modifying Key Rules

CTE v7.4 (Windows) and CTE v7.6 (Linux), and subsequent versions, allows users to modify the Key Rules while an LDT policy is active and enforced on a client. Users can add new rules and modify the existing rules. The following use case is addressed with this change:

1. Users can prioritize and plan the encryption of large files inside a GuardPoint.
2. User can transform the excluded files by adding new Key Rules.

Warning

While making Key Rule changes, you cannot change the order of the rules, or the encryption key rules applied to the file. If the data encryption keys are mismatched, it may result in data corruption.

Note

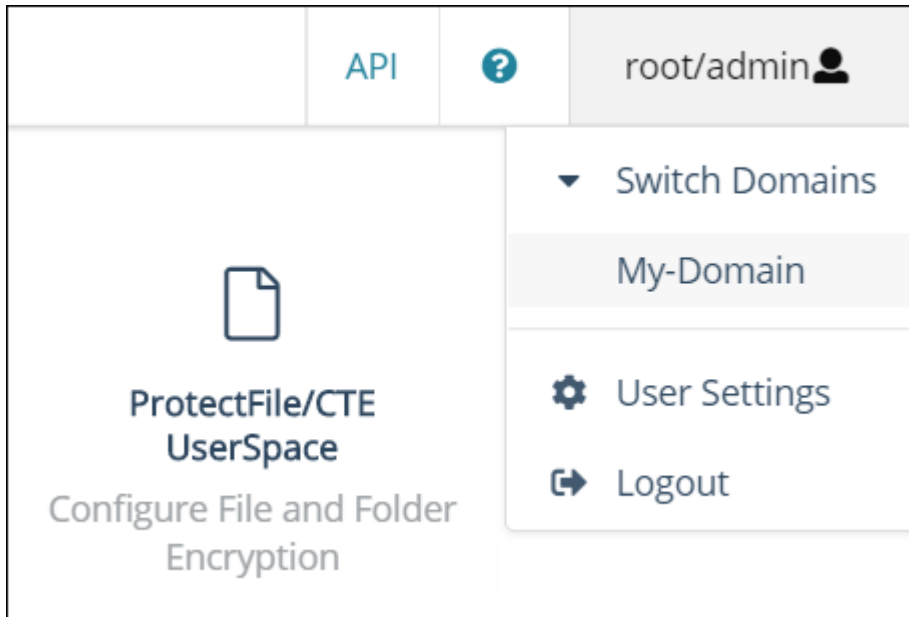
Dynamic resource sets for CTE Linux are compatible with CipherTrust Manager v2.14 and subsequent versions.

Creating CTE-LDT Policies

CTE-LDT uses a single Live Data Transformation encryption policy to address both data transformation and ongoing protection. In contrast, if you do not use CTE-LDT, you need a separate policy for initial data transformation, or rekey, and another policy to protect the data while it is in production use. For more information about CTE-LDT policies, see [CTE-LDT Policies](#).

To create an Live Data Transformation encryption policy:

1. In a web browser, navigate to the URL of the CipherTrust Manager you want to use and log in with CipherTrust Manager Administrator credentials.
2. If the client you want to protect is registered to the default domain (root), proceed to the next step. If you need to change to a different domain, do the following:
 - a. In the top menu bar, click the domain/user name (default: **root/admin**) on the right-hand side.
 - b. Select **Switch Domains**, then select the domain in which the client is registered.
 - c. The logged in user now shows the new domain name/user name.



3. In the CipherTrust Manager Applications Page, open the **CTE** application.
4. In the left-hand menu bar on the Clients page, click **Policies**.
5. Click **Create Policy**. CipherTrust Manager displays the Create Policy Wizard.
6. On the General Info page, set the following options:

Field	Description
Name	A unique name for the policy. Make sure you use a name that is descriptive and easy to remember so that you can find it quickly when you want to associate it with a GuardPoint. For example: LDT-Policy-West
Policy Type	Select Live Data Transformation.
Description	A user-defined description to help you identify the policy later. For example: LDT policy for the West Coast Datacenter.
Learn Mode	Learn Mode provides a temporary method for disabling the blocking behavior of CTE/CTE-LDT policies. While useful for quality assurance, troubleshooting, and mitigating deployment risk, Learn Mode is not intended to be enabled permanently for a policy in production. This prevents the policy Deny rules from functioning as designed in the policy rule set. Ensure that the policy is properly configured for use in Learn Mode. Any Security Rule that contains a Deny effect must have Apply Key applied as well. This is to prevent data from being written in mixed states, resulting in the loss of access or data corruption.

Field	Description
	Apply Key will have no effect when combined with a Deny rule unless the policy is in Learn Mode.

7. Click **Next**.

8. On the Security Rules page, define the security rules you want to use.

CipherTrust Manager automatically adds a default security access rule with an action of `key_op` and the effects `Permit` and `Apply Key`. This rule permits key operations on all resources, without denying user or application access to resources. This allows it to perform a rekey operation whenever the encryption key rotates to a new version. This rule is required by CTE-LDT, so you cannot edit it, move it, or delete it.

To add additional security rules, click **Create Security Rule** and enter the requested information. If you want to grant user and application access to files at all times, including during initial CTE-LDT and subsequent key rotations, add a security rule with the action `all_ops` and the effects `Permit` and `Audit`.

9. On the Create Key Rule page, click **Create Key Rule** and enter the following information:

Field	Description
Resource Set	If you want to select a resource set for this key rule, click Select and either choose an existing resource set or create a new one. Resource sets let you specify which directories or files will either be encrypted with the key or will be excluded from encryption with this key.
Current Key Name	Click Select to choose an existing key or create a new one. If the data has not yet been encrypted, select <code>clear_key</code> . Otherwise select the name of the non-versioned key that is currently being used to encrypt the data.
Transformation Key Name	Click Select to choose an existing versioned key or to create a new one. CTE uses the versioned key specified in this field to encrypt the data in the GuardPoint. If the data is currently encrypted, CTE decrypts it using the key specified in the

Field	Description
	Current Key Name field and re-encrypts it using the key specified in this field.

Note

LDT now supports dynamic resource sets. This means that you can alter the resource set, and the policy will pick up the changed resource set and apply it. However, make sure that you **do not** mix encryption keys in your edited resource set. This will cause corruption.

When you are done specifying key rules, click **Next**.

10. On the Confirmation page, review the information you entered and click **Save** when you are ready to save the policy.

The following example shows a simple CTE-LDT policy that encrypts clear-text files using a versioned key named LDT-Key-1. The `key_op` and `all_ops` actions in the Security Rules panel grant user and application access to files at all times, including during initial CTE-LDT and subsequent key rotations.

1 General Info

Name: LDT-Policy-West
Policy Type: Live Data Transformation
Description: LDT policy for the West Coast Datacenter

2 Security Rules

Resource Set	User Set	Process Set	Action	Effect	Browsing
			key_op	permit,applykey	No
			all_ops	permit,audit,applykey	Yes

3 Key Rules

Resource Set	Current Key Name	Transformation Key Name	Exclusion Rule
	clear_key	LDT-Key-1_xts	No

Security Rule Ordering for Policies

If you want to enforce restrictions when guarding NFS shares using an LDT or standard policy with a **CBC-CS1** key, note the following:

CipherTrust Transparent Encryption embeds and hides LDT and/or IV (initialization vector) attributes in the first 4K of files for NFS shares guarded with an LDT or standard policy with a **CBC-CS1** key. Embedding CipherTrust Transparent Encryption attributes

increases the actual file size by 4K, and CTE hides that extra 4K when reporting the file size. The exception to this is when a backup/restore process reads/writes such files. This requires embedded attributes to be read/restored by the backup/restore process. In such cases, CipherTrust Transparent Encryption does not hide the 4K attribute space in the file. The backup user/process views the actual file size. Non-backup users/applications view the file size as less than 4K.

If you want a security rule to enforce restricted access for reading file level attributes on such GuardPoints, you must specify the **Apply Key effect**. Alternatively, you can place the security rule that is enforcing the restricted access after the rule granting read/write access. This avoids application failure if the Apply key effect is not desired. For example, the order of the two rules in a policy that does not hide the user user-name would be:

- **Security Rule n**

Rule	Value
User	<user-name>
Action	Read-file-attribute and/or Read directory
Effect	permit

- **Security Rule n + 1**

Rule	Value
User	<user-name>
Action	all_ops
Effect	permit, apply-key

Assuming <user-name> is not affiliated with backup/restore operations, <user-name> would view the actual file size which is 4K larger than the size of the user data in the file. The returned file size can result in failure when user-name attempts to read/write files. By reordering rules n and n + 1, <user-name> will view the correct size hiding the 4K attribute space in the target file.

For more information, see [Adding Security Rules](#) in the CipherTrust Manager documentation.

Quality of Service

- Purpose of QoS
 - Manage CTE-LDT Impact
 - Monitor and Control CPU Usage
 - Monitor and Control Rekey/Scan I/O Rate
- QoS Scheduling During Backup/Restore

CTE-LDT runs in real time, while users actively interact with applications. This could impact performance. However, CTE-LDT is designed to not adversely affect application or system performance.

Purpose of QoS

Quality of Service (QoS) provides tools for an administrator to minimize the effect of CTE-LDT on system and application performance. It provides a set of parameters that administrators can set to control CTE-LDT use of system resources, primarily CPU and I/O bandwidth. When the QoS parameters are set appropriately, CTE-LDT stays within the defined boundaries to ensure that critical user applications are not adversely affected by CTE-LDT operations.

Manage CTE-LDT Impact

Administrators can pause or resume CTE-LDT operations to manage and control CTE-LDT impact to application workload. When data transformation occurs, either during initial or subsequent transformations, it requires substantial host CPU and I/O resources. This can cause contention for resources between the applications simultaneously running on the protected host. The administrator specifies QoS settings on each host, or at a host group level, that is using CTE-LDT. When CTE-LDT is running, QoS monitors CPU or rekey/scan rate on the host and enforces the QoS settings. QoS can also monitor and enforce an administrator imposed limit on the volume of data undergoing rekey per second. The QoS settings enable you to strike a balance between completing an CTE-LDT process and not interfering with host application performance.

Monitor and Control CPU Usage

QoS monitors and controls the use of host system resources during CTE-LDT, specifically, CPU usage and rekey/scan rate.

Note

You can control CPU usage or rekey/scan I/O rate, but not both. The CPU usage and rekey/scan I/O rate options are mutually exclusive.

Monitor and Control Rekey/Scan I/O Rate

You can choose Rekey I/O Rate as a threshold to control the CTE-LDT processing rate. When this threshold is entered, the Quality of Service continuously monitors CTE-LDT transformation and enforces the specified amount of data during:

- **Rekeying:** CTE-LDT is transforming the data on active GuardPoints based on the new key version.
- **Scanning:** CTE-LDT is analyzing files in GuardPoints. Scanning occurs:
 - Before initial transformation (Linux only)
 - Before a rekey (Linux only)
 - Following an interrupted rekey, such as a reboot on Linux or Windows, and also a directory rename or directory deletion on Windows

You can set the Rekey I/O Rate or CPU Threshold for multiple clients through the QoS Settings section in a CipherTrust Manager client profile. All clients associated with a given client profile will use the QoS thresholds set in that profile unless the thresholds are overridden locally on the individual client. (The default setting for Rekey I/O Rate in the client profile is 0 (zero), which means QoS will run full throttle.)

To set the QoS thresholds locally on a particular client, use the `voradmin ldt ior` and `<iorate>` command on that client. When you do so, the `voradmin` setting overrides the Rekey I/O Rate or CPU Threshold set in the CipherTrust Manager client profile. If QoS was already set locally on this client and you use `voradmin` to set the Rekey I/O Rate, CTE-LDT ignores any CPU threshold previously set.

To resume using the client profile QoS values for a client, use `voradmin ldt ior 0` to set the Rekey I/O Rate to 0 (zero). When you do so, CTE returns to using the Quality of Service settings in the client profile.

A tolerance level is associated with the Rekey I/O Rate. Together, the tolerance and Rekey I/O Rate specify a range for the CTE-LDT processing rate. The Quality of Service selects a proper tolerance for a Rekey I/O Rate provided through the `voradmin` command, and maintains the CTE-LDT processing rate at the specified Rekey I/O Rate plus or minus the tolerance. The tolerance is selected as follows:

- When the Rekey I/O Rate is less than or equal to 10MB/sec, the tolerance is 3MB/sec.
- When the Rekey I/O Rate is greater than 10MB/sec. and less than 50MB/sec, the tolerance is 4MB/sec.
- When the Rekey I/O Rate is at 50MB/sec or higher, the tolerance is 10% of the specified Rekey I/O Rate.

To set or reset Rekey I/O Rate on a single host, use the `voradmin` command as follows:

- To set the threshold of 50 MB/sec., use the following command:

```
voradmin ldt ior 50
```

- To reset the current threshold:

```
voradmin ldt ior 0
```

For more information about setting the Rekey I/O Rate using `voradmin`, see [Select and Set Rekey I/O Rate](#).

QoS Scheduling During Backup/ Restore

QoS scheduling plays an important role when backing up/restoring data without the *Apply Key* rule applied to the backup/restore process. During backup/restore, you **must** pause CTE-LDT operations before taking backups. QoS scheduling allows the administrator to enter the schedule for QoS aligned with the backup schedule, and pause the CTE-LDT processes for the duration of the backup. The schedule specifies which days of the week, and what times of day, CTE-LDT is permitted to run. CTE-LDT

cannot run at any time that is not permitted by the QoS schedule. QoS suspends CTE-LDT operations at all times outside of the schedule.

When setting a QoS schedule, consider your system and application peak demand periods during the day and week. Also consider your schedule for data backups. Schedule CTE-LDT to pause when you need all available system resources for other tasks, such as meeting peak user demand or performing data backups.

Note

On Windows, if your backup applications are using VSS, then you do not need to pause CTE-LDT on Windows.

How to Set QoS

To set QoS for a host to enforce a threshold on CTE-LDT based on CPU percentage:

1. Log into the CipherTrust Manager Console as an administrator.
2. Open the **CTE** application.
3. In the left-hand menu, select click **Profiles**.
4. If you want to add a new profile, click Create Profile and specify a profile name and description.
5. Click the profile name in the **Profiles** table.
6. Click the Quality of Service Configuration section heading to open the QoS options and select one of the following radio buttons:
 - **Rekey by Rate** — Specifies that you want to set a Rekey I/O Rate threshold. Enter a value between 0 and 100 in the associated **LDT QoS Rekey Rate** field. If you specify 0, CTE does not throttle the CTE-LDT rekey or scanning processes.
 - **Rekey by CPU**— Specifies that you want to set a CPU Rate threshold. Enter a value between 0 and 99 in the associated **CPU Percentage** field.

If you select this option, CTE-LDT applies a tolerance of +/- 2% on CPU threshold settings up to 7%. For threshold over 7%, CTE-LDT applies a tolerance of +/- 4%.

If CPU% is set to 0, QoS stops monitoring CPU usage and CTE-LDT operations run at maximum rekey rate within the available system resources. Setting CPU percentage to 0 does not affect CTE-LDT schedules, so CTE-LDT operations are suspended and resumed per QoS schedule.

If you do not enter CPU threshold percentage, CTE-LDT applies CPU threshold of 5% capped by default.

If CPU% is set to a very high value, such as 25%, the rekey process competes with other applications to use as many CPU cycles as it can. As a best practice, start with a setting of 10%, and increase or decrease it slowly by 5% until it reaches a reasonable level that does not adversely affect the performance of user applications. The higher the percentage, the more quickly CTE-LDT completes its processing. However, this speed causes increased competition for resources, which can significantly degrade the performance of other applications using this host.

Caution

Do not set CPU% to a very high value in an attempt to force faster data transformation. This can potentially exhaust other system resources.

When you select Rekey by CPU, the Cap CPU Allocation check box becomes available. Check this box to specify that the CPU allowance must never exceed the percentage set in CPU%. If Cap CPU Allocation is not checked, and additional CPU resources are available on the host, CTE-LDT consumes part of the available resources for rekey above the CPU threshold. Exceeding the threshold may impact your production workload as your production CPU resource consumption fluctuates over time.

For example, if CPU% is set to 10%, but Cap CPU Allocation is not set, the rekey process continues consuming available CPU cycles after reaching 10% CPU utilization, at which point the rekey process starts contending with applications for CPU cycles.

7. In the QoS Schedules section, select the schedule you want to use from the **LDT QoS Schedule** drop-down list. This list includes the following options:

- ANYTIME
- WEEKNIGHTS
- WEEKENDS
- CUSTOM

If you select anything other than CUSTOM, CipherTrust Manager displays the time ranges in which CTE-LDT can run in the **Time Ranges** column. If you select CUSTOM, you can then click the **Create New QoS Schedule** button to create a custom schedule. Add one custom schedule entry for each time range in which you want QoS to run.

By limiting CTE-LDT to periods of low application usage, you minimize the potential for resource contention between applications and CTE-LDT.

The following example shows a profile with the CPU threshold capped at 10%. The QoS schedule is set for 1 am to 4 am on Mondays, Wednesdays, and Fridays.

QUALITY OF SERVICE CONFIGURATION

Rekey by Rate
LDT QoS Rekey Rate (MB/s)

Rekey by CPU
 Cap CPU Allocation
CPU Percentage

QoS Schedules
LDT QoS Schedule: [Create New QoS Schedule](#)

Name	Time Ranges	
CUSTOM	Monday 1:00 AM - Monday 4:00 AM	...
CUSTOM	Wednesday 1:00 AM - Wednesday 4:00 AM	...
CUSTOM	Friday 1:00 AM - Friday 4:00 AM	...

QoS Best Practices

This section gives tips and examples to help you set QoS parameters for the best results.

Related Topics

- [General Best Practices for QoS](#)
- [Select and Set Rekey I/O Rate](#)

General Best Practices for QoS

- Use Rekey I/O Rate threshold to limit CTE-LDT impact, if any, to your production workloads. Rekey I/O Rate approach is a simpler method for a Administrator or system administrator to enforce a limit on the volume of data that CTE-LDT should rekey per second. You can choose a threshold, in units of MBs per second, which is a small percentage of peak IOPS from your production workload.

Note

When choosing a threshold on CTE protected hosts with GuardPoints over NFS/CIFS shares, you must consider network bandwidth between your host and NFS/CIFS servers. QoS does not monitor the impact of LDT operations on network connections between your hosts and NFS/CIFS servers. However, the rate selected as LDT Rekey IO rate directly correlates to the network bandwidth to the target NFS/CIFS servers. In the follow-up discussions for selecting optimal rekey IO rate, you must monitor and collect the network traffic instead of disk IO transfers.

- You will see the effects of QoS settings only if the number and/or types of files in the GuardPoints stress the rekey or scan processes. On hosts with a relatively small number of files, the rekey or scan process may complete quickly without hitting a threshold and causing throttling to occur.
- Use QoS CPU parameters as an alternate method for controlling the effect CTE-LDT has on application performance.

Set limits on CTE-LDT CPU usage whenever runtime monitoring shows that user applications are affected by CTE-LDT. Start by setting the CPU parameter to 10%, then increase or decrease in 5% intervals, as needed, to tune the CPU allocation. When an acceptable level is reached, and CTE-LDT is not noticeably affecting user applications, leave the QoS CPU parameters at a constant setting.

- Use monitoring tools.

Monitor host CPU utilization with tools like `vmstat`, `top`, and `iotop` on Linux and perform on Windows.

You can also monitor and obtain statistics with the `voradmin ldt stats` command.

For more information about `voradmin ldt stats`, see [Obtaining CTE-LDT Statistics at the Command Line](#).

- Select CPU resource allocation for CTE-LDT from 1% to the available limit minus 5%.

If the monitoring tools indicate system CPU usage, without CTE-LDT, it is at N%, available CPU resource is M%, where $M = 100 - N$. Select a percentage within 1 - (M - 5) to allocate to CTE-LDT CPU usage. However, remember that QoS tolerates 2% - 4% leeway in the actual CPU usage, so adjust your selection by 2 - 4%.

- Do not set CPU resources to 0% or 99% in an attempt to minimize or stop CTE-LDT

A CPU% value of 0 or 99 is reserved for disabling the QoS CPU monitoring function. This does not stop CTE-LDT or minimize its resource usage; rather the opposite. It enables CTE-LDT to run with its maximum rekey rate. Note that when CPU % is not set, CTE-LDT clients enforce a 5% CPU threshold by default.

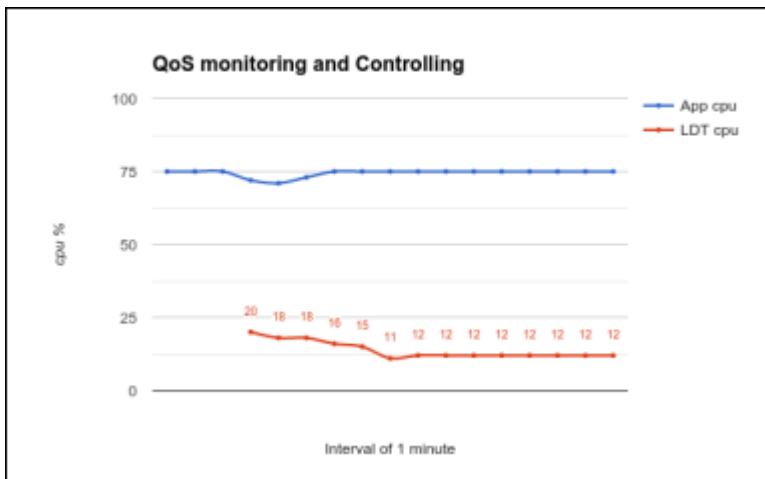
- Cap the CPU allocation.

QoS provides a **CAP CPU Allocation** parameter. Set this parameter to **True**. This ensures that CTE-LDT resource usage never exceeds the allocated percentage.

Example: Setting QoS before starting CTE-LDT

You can be proactive and set up QoS parameters before enabling GuardPoints that are protected with CTE-LDT policies. This ensures QoS starts monitoring and controlling CTE-LDT resource usage from the start. The following graph shows an example where 10% of the CPU is assigned to CTE-LDT. QoS makes sure that CTE-LDT is restricted to use only 10% of the CPU. There is a tolerance level of +/- 4%, so actual CTE-LDT usage can range between 5% and 15% of CPU. In the following example, applications use 75% of the CPU resources. As the graph shows, when CTE-LDT starts, application CPU utilization drops for a moment, because CTE-LDT exceeds the CPU threshold. QoS immediately reduces CTE-LDT's CPU usage to 12%, which is within tolerance levels for a 10% setting, and the application CPU share returns to normal.

QoS makes visible improvement immediately when CTE-LDT starts



The graph above was obtained on a Linux system running sysbench.

- To find the amount of CPU resources currently in use by applications, type:

```
top -n 1 -b | grep sysbench | awk 'BEGIN {cpu=0} {cpu += $9} END {print cpu}'
```

- To find the amount of CPU currently in use by the CTE-LDT-protected host, type:

```
top | grep Cpu
```

- To find the amount of CPU currently in use by CTE-LDT, type:

```
voradmin ldt stats | grep CPU
```

Example: Monitoring and controlling resource usage during CTE-LDT

Suppose that CTE-LDT has started with CPU set to 25%, and users realize their applications are affected. For example, there might be a higher than expected level of CTE-LDT I/O operations. To return application performance to normal, reduce the CPU allocation for CTE-LDT.

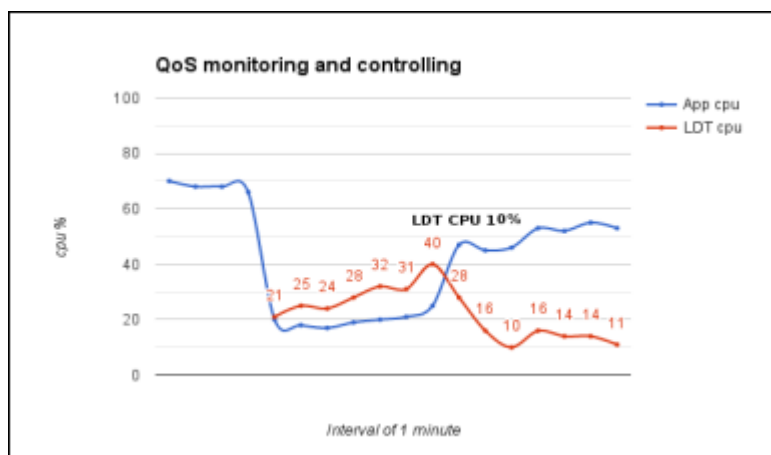
1. Set the **CPU parameter** to a lower value, such as 10%.
2. Select the **Cap CPU Allocation** option.

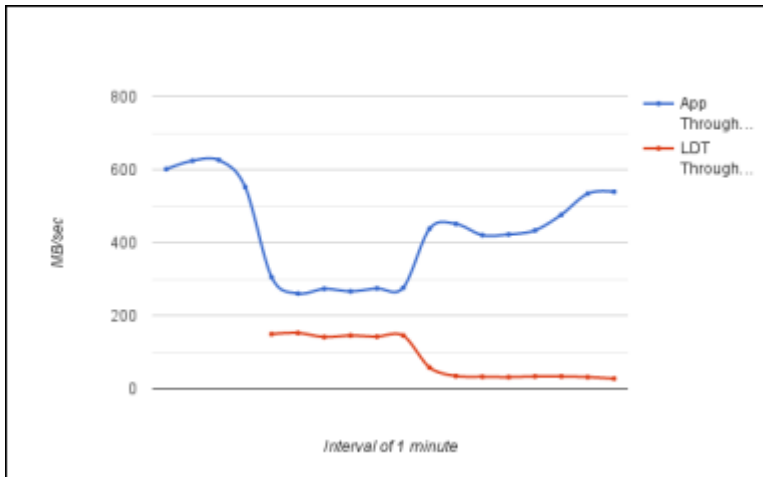
QoS restricts CTE-LDT CPU usage to 10%. The application user should monitor their application. If the application's performance is still affected, reduce the CPU parameter

further, such as to 5%. Repeat this procedure until application performance returns to a satisfactory level.

The following graph shows an example where QoS is not enabled to monitor and control CTE-LDT CPU usage from the start. When CTE-LDT starts, application CPU usage drops from 65% to 20%. By setting the QoS CPU parameter to 10%, application usage is greatly improved.

CPU usage allocation before and after QoS CPU parameter is set





To obtain the data for this graph:

- Use `iostat` and benchmarking tools on an RHEL system to obtain application I/O throughput.
- Use `voradmin ldt stats` to obtain the CTE-LDT I/O throughput and rekey rate.

Select and Set Rekey I/O Rate

You can choose to set the Rekey I/O Rate to control I/O operations from CTE-LDT to minimize CTE-LDT impact to your production workload. It's assumed that you already know the maximum IOPS on your host system during your production workload. With this information, you can choose a threshold for Rekey I/O Rate and enforce the selected threshold during CipherTrust Transparent Encryption - Live Data Transformation. The work flow is as follows:

1. Set **Rekey I/O Rate** threshold using `voradmin` or in the CipherTrust Manager Console.
2. QoS retrieves the threshold and starts monitoring and controlling CTE-LDT according to the specified threshold and the tolerance factor corresponding to the threshold.
3. The selected threshold will be in effect within 2 to 4 minutes after entering the threshold.

When Rekey I/O Rate and CPU or IOWAIT thresholds are set, QoS will monitor and control the CTE-LDT processing rate based on the Rekey I/O Rate threshold. The CPU threshold will be ignored.

Set Rekey I/O Rate Threshold

1. Set Rekey I/O Rate threshold by using `voradmin`:

```
voradmin ldt ior 10
```

You can also set the Rekey I/O Rate for one or more managed hosts using the Quality of Service section in a client profiles. For more information about using this method, see [How to Set QoS](#).

In the `voradmin` example above, QoS enforces the threshold of 10MB/sec with the tolerance of +/- 3MB/second. Effectively, CTE-LDT attempts to rekey the amount of data in the range of 7MBs/second to 13MB/second.

On Linux and Windows, you can use `voradmin ldt ior` to report the current threshold setting without specifying a value for threshold:

```
voradmin ldt ior
QoS Rekey I/O rate threshold: 10 MB/sec
QoS Rekey I/O tolerance: 3 MB/sec
```

2. Be sure the threshold you enter is appropriate for your production workload. To verify this:

- a. Observe the Rekey I/O Rate for a few minutes using `voradmin`.

On Linux, you can do this using:

```
voradmin ldt stats
Host level statistics:
File stats: rekeyed=202390, passed=0, created=0, removed=0
Data stats: rekeyed=6.2GB, truncated=0.0MB
QoS: IOR threshold=10MB/sec, tolerance=3MB/sec
current_rekey_rate=2MB/sec, current_iow=0ms
load_factor=50, delay_factor=0, delay_scan=0
```

On Windows, you can do this using:

```
voradmin ldt monitor
Host Stats:
Total number of Guard Points = 1
```

```

Rekey Status                = Rekey done (Finished rekey on
1 out of 1 GP's)
Total files to be transformed = 0
Total files transformed      = 0
Total files in progress     = 0
Total transformation threads = 0

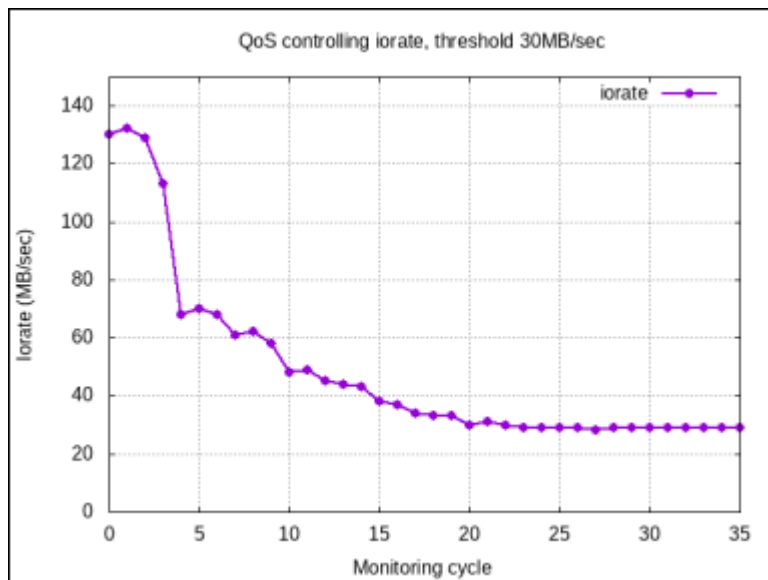
Current rekey rate          = 0 KB/s
Rekey IO rate threshold    = 1000 MB/s
Rekey IO rate tolerance    = 4 MB/s

```

b. Set an appropriate threshold. Do not set the threshold value too high, as QoS might not be able to achieve it because of other resource bottlenecks.

3. Check the QoS controlling rekey rate.

QoS will monitor and control CTE-LDT utilization using the specified threshold. The following figure shows an example of how QoS monitors and controls CTE-LDT utilization. In this example, the threshold is 30 MB/sec. Throughput of CTE-LDT was nearly 130 MB/sec. QoS brings it down to within the range of 30 MB/second.



4. Disable QoS.

QoS will not monitor and control resources when all the thresholds, CPU, Rekey I/O rate, and IOWAIT are set to 0. When Rekey I/O Rate and IOWAIT are not explicitly set, it is considered to be set to 90 MB/second.

QoS continues to apply its schedules for suspending CTE-LDT operations at certain days and times regardless of what values are set for CPU, Rekey I/O Rate, and IOWAIT thresholds.

Summary of QoS Resources

The following table summarizes the available thresholds and the actions of QoS module to enforce the set thresholds:

Scenario	QoS Action
Only Rekey I/O Rate threshold is set	Monitor and control the CTE-LDT processing rate based on Rekey I/O Rate
Rekey I/O Rate and CPU threshold are set	Monitor and control the CTE-LDT processing rate based on Rekey I/O Rate. CPU threshold is ignored.

Creating a CTE-LDT GuardPoint

After you have installed the license and registered the CTE-LDT host (see [Installing CTE-LDT](#)), you can create a CTE-LDT GuardPoint on the host. When you create the CTE-LDT GuardPoint, you select a Live Data Transformation policy and apply that policy with its transformation keys to that GuardPoint. CTE automatically gets the Quality of Service settings from the associated Client Profile in CipherTrust Manager.

This section describes two scenarios:

- [Creating an CTE-LDT GuardPoint for an Unguarded Directory](#)
- [Converting a Non-CTE-LDT GuardPoint to an CTE-LDT GuardPoint](#)

Creating an CTE-LDT GuardPoint for an Unguarded Directory

To create an CTE-LDT GuardPoint on what was previously an unprotected/unguarded directory:

1. Create an Live Data Transformation policy that transforms data from clear text to a versioned key. In the policy, set Current Key to clear_key and Transformation Key to the versioned key. For details, see [Creating CTE-LDT Policies](#).

2. Set, or modify, the Quality of Service (QoS) parameters in the Client Profile associated with the CTE client to account for CTE-LDT on all GuardPoints on this client. For details see [Quality of Service](#).
3. In the CipherTrust Manager Applications Page, select the **CTE** application.
4. In the Clients table, click on the name of the client you want to protect.
5. Above the GuardPoints table, click **Create GuardPoint**.
6. In the Create GuardPoint page:
 - a. In the **Policy** field, select the CTE-LDT policy you created earlier.
 - b. In the **Type** field, select the type of device. For CTE-LDT, you can select **Auto Directory** or **Manual Directory**.
 - c. In the **Path** field, enter the directories you want to protect with this policy or click **Browse** to select them from a Windows-style explorer.
If you want to enter multiple paths, put each path on its own line.
 - d. Keep the **Preserve Sparse Regions** option selected if you want CTE to ignore sparse regions during data transformation.
A sparse region is a region within the file size that has not yet been written to. Therefore, it is not allocated with disk blocks. Any attempt to read a sparse region reads stream of zeros as data. A file may have one or more sparse regions, or an entire file may be sparse.
If you select **Preserve Sparse Regions**, CTE-LDT detects and skips transforming sparse regions. Therefore, it does not change the number of blocks utilized in the file system. This is the default.
If you disable **Preserve Sparse Regions**, CTE-LDT transforms a file without checking or skipping sparse regions, if they exist. Consequently, as CTE-LDT operations transform and fill sparse regions with encrypted stream of zeros, sparse regions are allocated with disk blocks. This increases the number of disk blocks utilized in the file system.
 - e. Click **Create**.
 - f. If you want to use the same policy and GuardPoint type on another path, click **Yes** when prompted. Otherwise, click **No**.

The CipherTrust Manager pushes the GuardPoint configuration to the client and CTE immediately begins transforming the data in the specified folders from clear-text to cipher-text.

7. When the data transformation has finished, applications can resume accessing the now-protected data.

Converting a Non-CTE-LDT GuardPoint to an CTE-LDT GuardPoint

After enabling CTE-LDT on a client, you can change a non-LDT GuardPoint to a CTE-LDT GuardPoint. CTE-LDT GuardPoints provide the advantage of allowing users to access all files in the GuardPoints while encryption is occurring. There is no downtime for the user except for the time needed to apply the GuardPoint.

1. Write a new CTE-LDT policy that transforms data from the non-CTE-LDT/non-versioned key used in the existing GuardPoint, to an CTE-LDT versioned key. See [Creating CTE-LDT Policies](#).
2. Make sure there is no application activity within the GuardPoint.

Caution

This step is critical. Do not skip it. Make sure there is no application activity within the GuardPoint.

1. Remove the current GuardPoint.
 - a. Open the **CTE** application and click **Clients** in the left-hand menu bar.
 - b. Click on the name of the client whose GuardPoint you want to change.
 - c. Find the GuardPoint you want to change in the GuardPoints table, then click the (...) button at the end of the row and select **Remove**.
2. Guard the directory again using the new CTE-LDT policy. Use the steps in [Creating an CTE-LDT GuardPoint for an Unguarded Directory](#), but choose the policy that starts from the non-versioned/non-LDT key rather than a policy that starts from `clear_key`.

Rotating Encryption Keys (Rekey)

- [Manual Key Rotation](#)
- [Creating a Key Rotation Schedule](#)
- [Checking the Rekey Status](#)
- [Obtaining Information About Keys Applied to Files](#)
- [Showing GuardPoints During Rekey \(Linux\)](#)
- [Suspending and Resuming Rekey and/or Scan Phase](#)
- [Automatic Suspend and Resume of CTE-LDT Operations Due to Insufficient Disk Space](#)
- [Rotating Encryption Keys While a Rekey is in Progress \(Relaunch\)](#)

Manual Key Rotation

If you need to rekey your GuardPoint prior to the expiration of the current key version, you can manually start a rekey process.

On Windows, you cannot rotate a key until all current data transformations that use the same key have completed, whether those transformations are an initial data transformation or a rekey. On Linux, you can rotate the key at any time. For details, see [Rotating Encryption Keys While a Rekey is in Progress \(Relaunch\)](#).

1. In the CipherTrust Manager Applications Page, open the [Keys & Access Management] application.
2. Click **Keys** in the left-hand menu bar.
3. In the Keys table, click on the name of the key that you want to rotate.
4. On the Key Detail page, click the Add button (+) above the table to add a new version.



The new version appears in the Versions table.



ID	Aliases	Version	State	Updated	Links
7def7dc68d4449bab09d3b8afbb641527e8cbb6d24e64c0ab1dc3eed6a2d5db4		1	Active	Saturday, September 5th 2020, 11:08:22 pm	↩️ ⋮
6053541301aa4b0dac005881911addd4631a3c3e7baa4b59b061561172cb8c30		0	Active	Thursday, September 3rd 2020, 4:30:30 pm	↩️ ⋮

Creating a Key Rotation Schedule

In CipherTrust Manager, you can create an automatic key rotation schedule that will automatically rotate all of the keys included in the schedule on a periodic basis. As soon as CipherTrust Manager creates a new version of a key, it pushes the new version to any clients associated with the policies that use the key. If the policy is a Live Data Transformation policy, CTE automatically begins rekeying the data in the CTE-LDT GuardPoint with the new version of the key.

1. In the CipherTrust Manager Applications Page, open the **Admin Settings** application.

2. Click **Schedules** in the left-hand menu bar.

CipherTrust Manager displays the existing key rotation and backup schedules that have been defined in the system. Thales strongly recommends that you look at any existing key rotation schedules to make sure that the keys you want to rotate are not already included in one of those existing schedules.

If the existing key rotation schedules do not include the keys you want to rotate, continue with this procedure to create a new schedule.

3. On the **Schedules** page, click **Add Schedule**.

4. On the **Select Schedule Type** page, click **Key Rotation** and click **Next**.

5. Add a schedule name and description. Make sure you name the schedule as descriptively as possible so that other users can tell at a glance what keys that schedule includes. For example, you could name the schedule "Rotate-LDT-Keys" and use the description "Rotates all keys with "LDT" in their name on a yearly basis."

When you are done, click **Next**.

6. On the **Schedule Config** page, enter the following information:

- **Duration.** Enter the day and time on which CipherTrust Manager should start using the key rotation schedule in the **Schedule Starts** field. When this day and time is reached, CipherTrust Manager looks at the date in the **Frequency** section and automatically creates a new version on that date.

Enter the day the schedule should end in the **Schedule Ends** field, or select the **Never** check box to tell CipherTrust Manager there is no end date for this key rotation schedule. If you select a date in this field, CipherTrust Manager automatically stops creating new key versions when that day and time are reached.

- **Frequency.** Select the **Basic** radio button to specify the frequency (daily, weekly, monthly, or yearly) and the UTC time at which CipherTrust Manager should automatically rotate the key after the schedule starts.

Select the **Raw (Cron)** radio button to create a cron job to control the key rotation schedule. Specify cron format in the following order:

minute, hour, day of month, month, and day of week

These five values indicate when the job should be executed. These values are mandatory and must be specified in the order given. The allowable values are:

Field	Allowed Values
minute	0-59 or * / , -
hour	0-23 or * / , -
day of month	1-31 or * / , -
month	1-12 or JAN-DEC or * / , - This field is case in-sensitive, so JAN, Jan, or jan are all equally valid.
day of week	0-6 or SUN-SAT or * / , - This field is case in-sensitive, so SAT, Sat, or sat are all equally valid.

Examples:

- December 31 at 2:35 AM UTC could be specified as: `35 2 31 Dec *`.
- On the 25th of every April, August, and December at midnight UTC could be specified as: `0 25 APR, Aug, dec *`.
- Every Monday, Wednesday, and Friday at 8:00 PM UTC could be specified as: `0 20 * * 1, 3, 5.`

When you enter a **Duration** and a **Frequency**, CipherTrust Manager displays the next several times the key rotation will be run under the **Frequency** field. Make sure the key rotation will be done on the days you expect.

Note

While the specified time is in UTC, the scheduled run times are shown in your local time. Therefore there may not be an exact match between the time set in the **Frequency** field and the displayed run time.

When you are done, click **Next**.

7. On the **Key Rotation Page** page, specify the selection criteria you want CipherTrust Manager to use when it selects the keys it will rotate.

Caution

Make sure you are extremely careful when you specify the selection criteria so that you do not accidentally rotate keys that should not be rotated using this schedule. Whenever a new key is added to CipherTrust Manager, CipherTrust Manager compares its name and details to the filters set in all key rotation schedules defined in the Admin Settings application, and it automatically adds the new key to ALL key rotation schedules where it matches the selection criteria.

If your selection criteria is too broad, you could accidentally rotate keys that should not be rotated by this key rotation schedule.

You can enter a key name in the **Name** field, or select any of the available filters from the drop-down lists. The Selected Keys table shows the keys that will be rotated by this schedule.

CipherTrust Manager displays the current list of keys that match this criteria in the **Selected Keys** table. Make sure you go through this list carefully so that you do not rotate keys you do not want to rotate.

For example, you can rotate all keys with "ldt" anywhere in their name by specifying `*ldt*` in the **Name** field:

Add Key Rotation Schedule ✕

1 Select Schedule Type 2 General Info 3 Schedule Config 4 Key Rotation Query

A key rotation query is an object that specifies the keys that need to be rotated. All keys are rotated when it is not provided.

Name 🔍

Filters Basic Raw

Types Size Status Dates

Latest Version Only

Name: *ldt* Version: latest

Selected Keys

Key Name	Ver	Modified	Type	Alg	Size
LDT-Key-West-Coast	2	08 Sep 2020, 10:10	Symmetric	AES	256
LDT-Key-East-Coast	3	08 Sep 2020, 10:10	Symmetric	AES	256
My_LDT_Key	0	08 Sep 2020, 10:07	Symmetric	AES	256

[Back](#) Save

If you only want to rotate the keys whose names begin with "ldt", use the query `ldt*` in the **Name** field. In the example above, that removes `My_LDT_Key` from the list of keys that will be rotated.

The search is case insensitive, so "ldt" will also match `LDT`, `ldT`, `ldT`, or any variation thereof.

When you are certain the list of keys to be rotated is accurate, click **Save**.

Checking the Rekey Status

During a rekey, you can check the progress in the CipherTrust Manager Console, however GuardPoint status is not relayed to the CipherTrust Manager in real time. A delay of several minutes before the CipherTrust Manager displays events on the client is likely. When the number of GuardPoints on the managed host is high, for example 100+, the delay in relaying GuardPoint status is due to delays in scheduling and the execution of CTE-LDT operations on the managed host for GuardPoints.

To see the progress of a rekey and the estimated completion time:

1. In the CipherTrust Manager Applications Page, open the **CTE** application.
2. Click **Clients** in the left-hand menu bar.
3. Click the **Client Name** of the client whose CTE-LDT status you want to check in the Clients table.
4. Find an active CTE-LDT GuardPoint and click **Active** in the **Status** column. CipherTrust Manager displays the GuardPoint Health page, which includes transformation status information such as the basic status (Rekeyed or Rekeying), the last transformation completion time, the last transformation start time, and the estimated rekey completion time.

Obtaining Information About Keys Applied to Files

Key Report Option

In the following command, you can use the `report` option of the `voradmin` command to obtain information about all of the keys in use on the GuardPoint. The report lists all keys used in the GuardPoint. For each key, it gives the key name and key version number. It lists each unique key name and version combination only once, no matter how many files use the key.

The following example shows three keys used in the GuardPoint `/oxf-fs1/gp1`:

```
voradmin ldt key report /oxf-fs1/gp1
LDT_KEY1,1
LDT_KEY2,2
LDT_KEY3,5
```

For an overview of `voradmin ldt`, see [CTE-LDT Command-Line Administration: voradmin command](#).

Key Map Option

In the `voradmin ldt key [report|map] <key_name> <guard_path>` command, you can use the `map` option to obtain information about which files in a GuardPoint were transformed with a specific key, where:

- `<key_name>` is the name of the key.
- `<guard_path>` is the path of the GuardPoint where the key was used.

For example, to view information about the key `LDT_KEY2` in the GuardPoint `/oxf-fs1/gp1`, you would enter:

```
voradmin ldt key map LDT_KEY2 /oxf-fs1/gp1
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
/oxf-fs1/gp1/file12345.dat10
```

Keys without a version number are used by files in an exclusion key rule or files that have yet to undergo initial key rotation. Use `voradmin ldt key map` in conjunction with `voradmin ldt attr get` to determine if a file using a key without a version number is part of an exclusion key rule or awaiting initial key rotation.

Showing GuardPoints During Rekey (Linux)

Use the following command to display a list of known CTE-LDT metadata stores and any associated GuardPoints currently undergoing transformation.

```
voradmin ldt list all
MDS_1: type=file, nguards=0, name=/disk2/___vorm_mds___
Guard Table: version 1 nentries 0
MDS_2: type=file, nguards=0, name=/disk3/___vorm_mds___
Guard Table: version 1 nentries 0
MDS_3: type=file, nguards=0, name=/disk4/___vorm_mds___
Guard Table: version 1 nentries 0
```

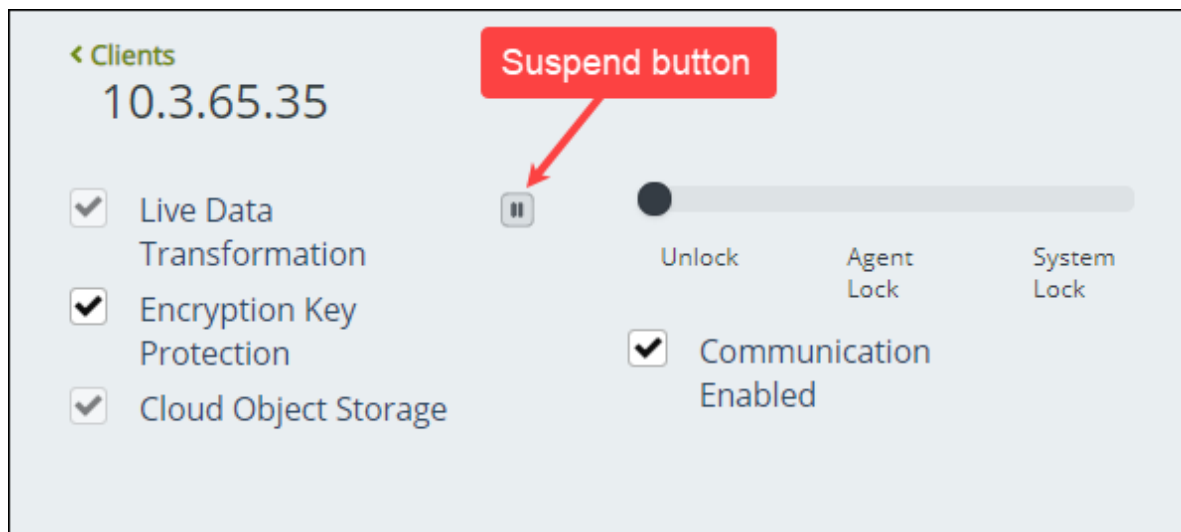
For an overview of `voradmin ldt` commands, see [CTE-LDT Command-Line Administration: voradmin command](#).

Suspending and Resuming Rekey and/or Scan Phase

The QoS schedule specifies certain time windows when CTE-LDT operations must be stopped temporarily. However, you can also suspend or resume CTE-LDT at the host/client level at any time.

To suspend CTE-LDT through the CipherTrust Manager Console:

1. Open the **CTE** application.
2. Click **Clients** in the left-hand menu bar.
3. Click the **Client Name** of the client on which you want to pause CTE-LDT in the Clients table.
4. In the Client Details area, click the **Suspend** button next to the Live Data Transformation field to suspend all CTE-LDT operations on the client.



After you pause CTE-LDT, the button turns into a Play button so that you can resume CTE-LDT operations when ready.

Considerations

- Disabling, or unguarding, an CTE-LDT GuardPoint where a rekey is in progress automatically suspends the CTE-LDT rekey operation.

- You can also use the `voradmin suspend/resume` command in Windows Powershell to suspend and resume the rekey. However, if you use the `voradmin ldt suspend` command to pause the rekey process, the CTE-LDT suspended state is not retained when the system reboots. If you want to retain the state after reboot, suspend the rekey from the CipherTrust Manager.
- On Linux, you can pause CTE-LDT during the scan phase of CTE-LDT. When paused during scan, CTE-LDT suspends operations that traverse through the namespace of GuardPoints in the scan phase of transformation. Suspending CTE-LDT during scan stops file lookup operations of CTE-LDT. This eliminates performance impact to I/O intensive production workloads, such as file serving type workloads, that access large number of files.

Automatic Suspend and Resume of CTE-LDT Operations Due to Insufficient Disk Space

CTE-LDT requires adequate storage space headroom to perform CTE-LDT operations such as rekeying on GuardPoints. On Linux, if available storage space drops below the threshold required for CTE-LDT operations to continue, the CTE-LDT operations are automatically suspended. On Windows, CTE-LDT will be suspended if there is less than 3 GB free space in the GuardPoint. Once additional storage space is available in the file system, CTE-LDT operations automatically resume.

As available space approaches the threshold for automatic suspension, CTE-LDT sends an alert to the CipherTrust Manager to notify you that you should free up more storage space before CTE-LDT operations are suspended. The alert on the CipherTrust Manager is:

```
Low space on guard point [GuardPoint], increase free space or CTE-LDT will be suspended.
```

Behavior of Automatic Suspend and Resume of CTE-LDT Operations on ext4 File Systems

By default, ext4 file systems reserve a portion of the storage space for use only by privileged processes to prevent running out of storage space in file systems. In this situation, non-privileged processes are automatically blocked from writing to the file system until the free disk space level reaches the minimum threshold. As CTE-LDT operates in privileged mode, CTE-LDT operations continue without blockage even if the free disk space threshold is low. Because of this ext4 feature, CTE-LDT operations on ext4 file systems may not be suspended due to low available storage space, even when the `df` command reports storage is 100% allocated.

Rotating Encryption Keys While a Rekey is in Progress (Relaunch)

On Linux, if a key is rotated (either manually or automatically when a key version expires) while CTE-LDT is in progress on a GuardPoint, the key rotation is processed and queued, and the GuardPoint is marked for relaunch. Relaunch indicates the need to restart CTE-LDT after the current transformation completes. If the GuardPoint has been rekeyed and is flagged for relaunch, CTE-LDT launches as soon as the GuardPoint is enabled.

When this event occurs, the following message appears in the log file: "`LDT: Deferred key rotation on GuardPoint [GuardPoint] until after completion of current transformation.`"

When the new key rotation request is queued, files can be in one of three states:

- **Undergoing rekey to a previous key:** Files already undergoing transformation to a previous key when the new rekey request is queued, are rekeyed to the key version already in progress.
- **Scheduled for rekey to a previous key:** Files that start transformation after the new key rotation request is queued are rekeyed to the newest key version.

- **Rekeyed to a previous key:** Files that have already been rekeyed to a previous key remain in that state until all files undergoing rekey, or scheduled for rekey, have been processed. Once the current CTE-LDT process for the GuardPoint is complete, CTE-LDT automatically relaunches to transform any files that are not rekeyed to the latest key version. This includes any files that were rekeyed before the new key rotation request was queued, or that were undergoing transformation when the new key request was queued.

For example:

```
voradmin ldt attr get /oxf-fs1/gp1
LDT stats: version=2, rekey_status=rekeying,relaunch
  Number of times rekeyed:                1 time
  Rekey start time:                       2024/03/18 16:54:00
  Last rekey completion time:             2024/03/18 16:53:38
  Estimated rekey completion time:        0 days 0 hours 6 minutes
  Policy key version:                     344
Data stats:
  total=9.8GB, rekeyed=0.0MB
  truncated=0.0MB, sparse=0.0MB
File stats:
  total=1000, rekeyed=0, failed=0
  passed=0, skipped=0, created=0, removed=0, excluded=0
```

Note

Relaunch is supported on Linux only. It is not supported on Windows.

Relaunch when Directories are renamed

Prior to v7.6.0, when a user renamed a directory, LDT would set a relaunch flag on the GuardPoint during the scan phase of LDT. This relaunch flag would trigger an additional LDT process on the GuardPoint when the current transformation job completed.

As the transformation process can impose significant overhead on production workloads, subsequent relaunch of LDT, as the result of renaming directories, is not efficient.

CTE v7.6.0 records renamed directories during the scan phase, so it can locate the files relocated to other paths during the rekeying phase of LDT. This prevents the GuardPoint from going through another transformation due to the directory being renamed during the scan.

Limitation

You can only rename a maximum of 16 concurrent directories without triggering a relaunch. If more than 16 concurrent directories are renamed during the scan phase, LDT sets the relaunch flag on the GuardPoint.

File System Operations

File system operations may require attention from the Administrator. These operations are divided into the following categories:

- [Renaming Files and Directories](#)
 - [Renaming Directories on Linux](#)
 - [Caveats](#)
 - [Example](#)
 - [Considerations](#)
- [Deleting a File](#)
- [File Handling \(Windows\)](#)
- [Enabling GuardPoints in Read-Only mounted file systems \(Linux\)](#)
- [Copying Files Into a GuardPoint](#)
- [Behavior of Hard Links Inside and Outside of GuardPoints \(Windows\)](#)

Renaming Files and Directories

On both Linux and Windows hosts, you can rename files during initial transformation, or rekey operations.

- On Linux, you can now, as of v7.1.1, rename directories in a GuardPoint during a rekey operation.
- On Windows, CTE-LDT stops if it is transforming the contents of a folder and a user attempts to rename/move that folder.

You can change the stopping behavior of CTE-LDT using the configuration parameter

```
oxf_stop_on_rename.
```

Using the Registry Editor, or the Windows command line, add a registry entry in

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Vmmgmt\Parameters for  
oxf_stop_on_rename.
```

Registry Name	Values	Comments
oxf_stop_on_rename	0 - Disabled	CTE-LDT does not stop if there is contention.
	1 - Enabled	CTE-LDT stops if there is contention. Default: Enabled.

For the target folder in rename operation, CTE-LDT is never stopped.

Renaming Directories on Linux

Prior to CTE-LDT v7.1.0, you could not rename a directory within a CTE-LDT GuardPoint, during the rekeying process. However, starting in CTE-LDT v7.1.1, users and applications can now rename directories within an CTE-LDT GuardPoint, during the rekeying process.

When a directory is renamed, CTE-LDT starts performing single file rekey jobs, on files in that renamed directory. A single file rekey job is a background process that rekeys a specific file, independent of the rekey process occurring on the entire GuardPoint. This process is similar to how files marked with a lazy rekey flag are rekeyed to the latest key version upon access to the marked file. LDT supports a maximum of 16 concurrent renamed directories per GuardPoint in the scan or rekey phases. If an additional directory is renamed while there are already 16 directories being processed, the GuardPoint is marked for relaunch (See [Rotating Encryption Keys While a Rekey is in Progress \(Relaunch\)](#) for more information on the relaunch state). This will make rekey start again upon completion to pick up any additional renamed directories.

Note

If a GuardPoint is already marked for relaunch, directory renames will not trigger single file rekeys. This is because the relaunch of LDT will pick up the files in the renamed directories.

You can view rekey jobs in a GuardPoint using the command `voradmin ldt list all`. Rekey operations for files marked as lazy, as part of CipherTrust Intelligent

Remediation, and for directory rename, are listed as separate single file rekey jobs. In the example below, the file `/ldt/renamed/file.0` is undergoing rekey as a single file rekey job, as the result of changing the pathname to the file.

```
voradmin ldt list all

MDS_1:  type=file, nguards=9, name=/ldt/___vorm_mds___

    Guard Table: version 1 nentries 9

    Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/ldt

        File List: count 51

    Guard 1: type=FILE, state=REKEYING DIRTY, flags=FILE LOCKED, gp=/
ldt/renamed_dir/file.0

        File List: count 1
```

LDT can run a maximum of eight concurrent single file rekey jobs for directory renaming. This limit is shared across directory renames for all of the GuardPoints on a system.

Furthermore, while directory rename is in progress on a target GuardPoint, the GuardPoint will not transition into the rekeyed state, even if CTE-LDT has completed on the rest of the files in the GuardPoint. As soon as the single file rekey jobs associated with the GuardPoint complete, the GuardPoint transitions into a rekeyed state.

Caveats

The following caveats exist when renaming directories that are in GuardPoints:

1. A directory rename operation changes path names to an unknown number of files. As such, CTE-LDT cannot accurately estimate completion time on the target GuardPoint. In such a situation, the rekey progress reflected on the key manager may show that the GuardPoint is in 100% rekeyed status, and the status remains in the 100% rekeying status, until the files in the renamed directories are rekeyed to completion.

2. Directory rename operations also affect the accuracy of LDT statistics. Those files, in a renamed directory that are not rekeyed when the directory is renamed, are no longer accessible as part of rekeying the GuardPoint. However, those files will be rekeyed as part of rekeying the renamed directory. As the GuardPoint level rekey operations fail to find those non-rekeyed files, LDT stats update to reflect those files as 'passed' and their size is added to the 'truncated' stat. When those files are rekeyed, as part of the renaming directory rekey process, the 'rekeyed' status will be correctly updated.
3. The net effect of renaming a directory during a rekey will be reflected in an inaccurate 'passed' count, and the amount of data truncated for each file not rekeyed in the renamed directory.

Example

If there are 1024 files totaling 1 GB in the directory `/ldt/gp/dir`, within the GuardPoint `/ldt/gp` when rekey starts, the stats should appear as follows:

```
ls -l /ldt/gp/dir | wc -l

1024

du -h /ldt/gp/dir

1.0G    /ldt/gp/dir

voradmin ldt stats /ldt/gp

LDT stats on /ldt/gp: version=3, rekey_status=rekeying

Number of times rekeyed:          3 times

Rekey start time:                 2021/06/14 23:53:56

Last rekey completion time:      2021/06/14 21:26:09

Estimated rekey completion time: N/A

Data stats:
```

```
total=13.8GB, rekeyed=726.0MB, truncated=0.0MB
```

File stats:

```
total=1043, rekeyed=0, failed=0
```

```
passed=0, skipped=0, created=0, removed=0, excluded=0
```

Upon renaming of the directory `/ldt/gp/dir`, any files that have not yet been processed by a GuardPoint level rekey, is added to the **passed** and **truncated** values in the stats:

```
voradmin ldt stats /ldt/gp
```

```
LDT stats on /ldt/gp: version=3, rekey_status=rekeying
```

```
Number of times rekeyed:                3 times
```

```
Rekey start time:                        2021/06/14 23:53:56
```

```
Last rekey completion time:2021/06/14 21:26: 09
```

```
Estimated rekey completion time:        N/A
```

Data stats:

```
total=13.8GB, rekeyed=10.6GB, truncated=1.0GB
```

File stats:

```
total=1043, rekeyed=21, failed=0
```

```
passed=1024, skipped=0, created=0, removed=0, excluded=0
```

Considerations

Consider the following issues when renaming directories during a rekey:

- If a directory rename operation occurs during the scan phase of CTE-LDT, the GuardPoint will be marked for relaunch and CTE-LDT will be launched again at the completion of the current rekey cycle. This is because it is not possible to begin rekey jobs while CTE-LDT is in the scan phase.
- When CTE-LDT is suspended on a GuardPoint with a renamed directory, additional single file rekey jobs will not be launched until rekey resumes. However, ongoing single file rekey jobs continue until each single file rekey is completely rekeyed. Consequently, suspending CTE-LDT while single file rekey operations are in progress only suspends rekey operations at the GuardPoint level, while CTE-LDT operations on single rekey files continues without disruption. As a result, stopping CTE service, which requires suspending CTE-LDT and disabling GuardPoints, may be delayed until the files currently undergoing rekey, due to directory rename, are completed.
- Ensure that CTE-LDT is suspended at the GuardPoint level and there is no single rekey file in progress before starting operations, such as a backup, which requires suspending CTE-LDT operations. Failure to do so can result in unexpected problems.
- If files in a renamed directory are moved out of the renamed directory, those files are processed as separate rekey jobs. Those rekey jobs cannot be suspended until the rekey is completed.
- A system crash during a rekey for a directory rename, also affects how CTE-LDT performs a recovery. Normally, CTE-LDT requires that, after the crash, the recovery completes for all of the affected files in the GuardPoint, before resuming rekey operations. However, with a directory rename, it is possible for the CTE-LDT recovery to skip files that are in the renamed directory because the path name to those files has changed since the scan phase of the key rotation. Instead, recovery for these files is deferred until the next access by CTE-LDT, or a user application, when a GuardPoint is enabled.
- CTE-LDT recovery log messages are not logged to CTE-LDT recovery log files, if the recovery log messages become necessary in the event of a failed recovery attempt. In the event of an error, CTE-LDT logs error messages in syslog, blocks access to the file that could not be recovered, and marks the file in rekey error. Files in rekey error must be restored from a backup. Additionally, as files that cannot be found for recovery are now treated as being deferred (only when the directory rename occurs prior to the crash), any orphaned files that might have been moved to the lost& found directory, in the target file system, will not be

discovered as orphaned files and consequently skipped by CTE-LDT. CTE-LDT cannot discern if the orphaned files are missing or were linked to a renamed directory.

Deleting a File

When a file is removed before it is rekeyed, the file is not included in the total number of files transformed in the GuardPoint Health dialog box. Any discrepancy between total number of files to transform and those transformed is due to the removal of files from GuardPoints during CTE-LDT.

The `voradmin` command provides more detailed file level statistics related to rekey operations on the host. You can run `voradmin` to get file level statistics:

```
voradmin ldt stats <guard_path>
```

File Handling (Windows Only)

It is critical that you understand how the CTE-LDT process handles read-only, binary (executable), NTFS encrypted and NTFS compressed files.

The CTE-LDT process is subjected to all of the File System policies and attributes set on the files. In some cases, this prevents CTE-LDT from encrypting a file. If users or applications are accessing files while CTE-LDT is in progress, CTE-LDT cannot change the attributes of the files and encrypt the file. It is critical that you understand how CTE-LDT handles various types of files:

- **NTFS Encryption and Compression**

If NTFS encryption and compression is enabled on a file or folder, the CTE-LDT process cannot encrypt these files. To maintain the data coherency, CTE-LDT skips the encryption of these files. These files display as “passthrough” files in the CTE-LDT statistics.

- **Read-Only Files**

When CTE-LDT encounters read-only files, it rekeys the file by resetting the read-only attribute and then setting the attributes back again when the rekey completes. If a file is open, CTE-LDT skips this file.

1. If the file is not opened, CTE-LDT changes the attributes of the file and stores the original attributes in the file metadata.
2. CTE-LDT starts Rekey on this file.

3. If a user requests to open a file for writing while rekey is in progress, access is denied. User can only open files for reading.
4. CTE-LDT restores the attributes once rekey is done.

- **Executable Files**

If an executable is running, or files are exclusively locked by the application, the CTE-LDT process cannot encrypt those files as it is unable to acquire the required locks on the files. CTE-LDT skips these files and changes to the INCOMPLETE state.

Enabling GuardPoints in Read-Only mounted file systems (Linux)

Access to a GuardPoint enabled in a read-only mounted file system is restricted to read operations. You cannot modify data in such file systems, therefore, you cannot perform CTE-LDT operations on GuardPoints in read-only file systems. For this reason, CTE-LDT automatically suspends operations when GuardPoints are enabled on read-only file systems. CTE-LDT ignores all attempts to resume CTE-LDT operations until the underlying file system is remounted with read/write access.

Warning

You must disable a GuardPoint before changing the read/write mount options of the underlying file system. After changing the mount options, you can re-enable the GuardPoint. CTE-LDT operations adapt to the read/write options of the underlying file system when you enable the GuardPoint. Changing the mount options while a GuardPoint is enabled is unsupported and may result in unexpected errors.

Copying Files Into a GuardPoint

If you copy a file into a GuardPoint without an Apply Key rule, make sure that the file was previously copied from the same GuardPoint, or a GuardPoint protected with the same policy/versioned key. A copy operation, without an Apply Key rule, is the same as a backup or restore of a file from/to an CTE-LDT protected GuardPoint.

CTE enforces key rules of an CTE-LDT policy while a GuardPoint is enabled. CTE cannot enforce the key rules while a GuardPoint is disabled. Modifying or adding data/files inside a disabled GuardPoint is not only unsupported, but it also results in unrecoverable data corruption.

Warning

Do not add new files or modify existing files inside CTE-LDT protected GuardPoints while the GuardPoint is not enabled. This results in unrecoverable data corruption and/or files that cannot be accessed when the GuardPoint is enabled.

Behavior of Hard Links Inside and Outside of GuardPoints (Windows)

When using hard links on Windows, all of the hard links to a file must be within the boundary of a GuardPoint and must use the same key. The following scenarios provide additional details:

- If hard links to the same file are inside a GuardPoint and outside a GuardPoint, the effect on the file depends on what process accesses which hard link first. If the hard link within the GuardPoint is opened first, the file is transformed. If the hard link outside the GuardPoint is opened first, the file won't be transformed.
- If hard links to the same file exist in different GuardPoints with different keys, the file will be corrupted.
- If hard links to the same file exist in the same GuardPoint but with different keys, such as if folder-based rules are used, there will be a conflict in the key.

Note

For v7.2, hardlinks are not supported on CIFS shares.

Excluding Files or Directories from Rekey

When you add a resource set to a key rule in a policy, you can select the **Exclude Rule** option. This informs CipherTrust Manager that the directories and/or files in the resource set use the define key that is excluded from the LDT rekey process. If you specify `clear_key` for the key rule, then the directories and/or files in the resource set will be left unencrypted. Otherwise, you can select any non-versioned key to apply to those directories and/or files.

For more on Exclusion rules, see the following topics:

- [About the Exclusion Attribute for Files Matching an Exclusion Key Rule](#)
- [Requirements for Exclusion Key Rules](#)
- [Usage Notes and Limitations for Configuring Exclusion Key Rules](#)
- [Examples of Exclusion Key Rules](#)
- [Listing All Files Included in an Exclusion Key Rule \(Linux\)](#)
- [Listing All Files Included in an Exclusion Key Rule \(Windows\)](#)
- [Using Dynamic Resource Sets](#)

About the Exclusion Attribute for Files Matching an Exclusion Key Rule

Files matching an exclusion key rule have the status `rekey_excluded` in the CTE-LDT attribute. For more information about CTE-LDT attributes, see [CTE-LDT Metadata in Extended Attributes](#). To check for the exclusion attribute on a file, see [Determining if a File is Included in an Exclusion Key Rule](#).

The Exclusion Attribute is not Persistent

Exclusion from rekey is no longer a persistent property as of v7.6.0.

See [Dynamic Resource Set](#) for more information.

Determining if a File is Included in an Exclusion Key Rule

Use the `voradmin ldt attr get <path to file>` command to check whether a file is associated with an exclusion key rule.

- On Linux, an excluded file will include the status `rekey_excluded` in the `voradmin` output. For example:

```
voradmin ldt attr get /oxf-fs1/gp1/foo.txt
CTE-LDT attributes: rekeyed_size=0, rekey_status=rekey_excluded
```

- On Windows, an excluded file will include the attribute `Rekey Status Excluded` in the `voradmin` output. For example:

```
C:\> voradmin ldt attr get c:\gp1\foo.txt
CTE-LDT attributes:
Rekey Status Excluded
Initial Rekeyed Size 9 Bytes
Data Transformed 0 Bytes

Key:
Current Key Name/Version (Clear_Key)
New Key Name/Version (Clear_Key)
```

Removing the Exclusion Attribute From a File

To disassociate a file from an exclusion key rule, you must copy the file to a new file that is not associated with the resource set, or exclusion key rule, of that same source file. You can then remove the original file. The new file is created under the default key rule of whatever policy applies to the new file. Moving or renaming a file does not disassociate a file from an exclusion key rule.

For example, assume the following exclusion key rule where `Key_TextFiles` is a non-versioned key:

Exclusion Key Rule: Resource set `=*.txt`, Key = `Key_TextFiles`

If you copy the file `test1.txt` to `test1.foo`, `test1.foo` is created with whatever key is specified in the policy that matches the new file. The key for the new file could be a non-versioned key, versioned key, `clear_key`, or, no key at all if the new file is outside of a GuardPoint. The original file `test1.txt` remains unchanged and encrypted with the `Key_TextFiles` non-versioned key because the file remains in the exclusion key rule.

Rename and Restore Operations (Linux)

The effect of rename, or backup/restore operations, involving files associated or excluded with exclusion key rules, is complex. In some cases, the operations to restore, or rename, causes conflicts between the key associated with the source file, and the key associated with the target file. For example, the result of a rename operation involving a source file, not associated with an exclusion key rule, and target file name associated with a resource set of an exclusion key rule is different from the result of the same operation when the target file name is associated with the resource set of an exclusion key rule with `clear_key`.

Note

Be sure to review the effect of the operations below and avoid administrative operations that cross associations of files across multiple resource sets with conflicting key rules.

Backup/Restore

The table below illustrates the status and the key effect of restore operations involving mixed keys associated with source files from backup image and existing target files inside GuardPoint.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	xattr_error (failed)
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	xattr_error (failed)

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Exclusion clear key	Versioned	Exclusion clear key
Exclusion clear key	Exclusion key	xattr_error (failed)
Exclusion clear key	Exclusion clear key	Exclusion clear key

The table below illustrates the status and the key effect of restore operations involving mixed keys associated with source files from backup image and new target files not present inside GuardPoint.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	xattr_error (failed)
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	xattr_error (failed)
Exclusion clear key	Versioned	Exclusion clear key
Exclusion clear key	Exclusion key	xattr_error (failed)
Exclusion clear key	Exclusion clear key	Exclusion clear key

Rename Operation

The table below illustrates the status and the key effect of rename operations for different combinations of versioned, exclusion key, and exclusion clear key.

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Versioned	Versioned	Versioned
Versioned	Exclusion key	Exclusion key
Versioned	Exclusion clear key	xattr_error (failed)
Exclusion key	Versioned	Exclusion key
Exclusion key	Exclusion key	Exclusion key
Exclusion key	Exclusion clear key	Exclusion key
Exclusion clear key	Versioned	Exclusion clear key
Exclusion clear key	Exclusion key	Exclusion clear key

Key Applied to File in Backup	Key Applied to existing file in GuardPoint	Key Effect of Restore Operation on Existing file
Exclusion clear key	Exclusion clear key	Exclusion clear key

Requirements for Exclusion Key Rules

Keep in mind the following requirements when configuring exclusion key rules:

- Before adding an exclusion key rule to an existing policy, you must disable all GuardPoints protected with the policy. Log on to the CipherTrust Manager to disable the GuardPoint.
- You cannot choose a versioned key for the key in an exclusion key rule. Only non-versioned keys or `clear_key` (no encryption) are valid for exclusion key rules.
- All exclusion key rules must be *above* all CTE-LDT transformation key rules in the **Key Rules** area in the policy.

Usage Notes and Limitations for Configuring Exclusion Key Rules

Keep in mind the information in the following sections when configuring an exclusion key rule.

Adding an Exclusion Key Rule to an Existing Policy with Versioned Keys (Linux)

When adding an exclusion key rule to an existing policy, the exclusion rule only applies to newly created files. Existing files that match the exclusion key rule remain encrypted with the same versioned key(s) specified in the non-exclusion key rule in the policy and will be rekeyed to the key in the exclusion key rule when the versioned key(s) rotates.

To force an existing file that matches an exclusion key rule to be transformed to the key in the exclusion key rule (non-versioned in this example), use one of the following methods:

- Rotate the versioned key specified in the policy to initiate rekey operations on the GuardPoint.

- Copy the existing file within the GuardPoint. The new file will be associated with the resource set in the exclusion key rule and will be encrypted with the non-versioned key. You can then delete the original file.

To perform a similar conversion on Windows, see [Changing a Folder or Files from Versioned to Non-Versioned Key \(Windows\)](#).

Adding an Exclusion Key Rule for LDT over NFS

For an LDT over NFS GuardPoint with exclusion rules, all files, including excluded files, can only be read in clear when the GuardPoint is enabled. Excluded files on an LDT over NFS GuardPoint will not be readable when the GuardPoint is unguarded/disabled.

Adding an Exclusion Key Rule That is Part of an Active GuardPoint (Linux)

To edit and/or add an exclusion key rule to an CTE-LDT policy, all GuardPoints using the policy must first be disabled before the new key rule can be added.

Changing an Exclusion Key Rule That is Part of an Active GuardPoint (Windows)

Changes that you make to an exclusion key rule that is part of an existing policy in an active GuardPoint do not take effect until the GuardPoint that the exclusion key rule is part of is disabled and enabled again.

Conflicting Keys as the Result of Rename Operations

Do not attempt to move or rename a file encrypted with a versioned key to a name associated with an exclusion key rule with `clear_key`. If you attempt such a move or rename, the original file is unaffected but following error is output on Linux systems and a log entry is created:


```
<command name>: setting attribute 'user::secfs:xattr:' for 'user::secfs:xattr:': Invalid argument
<command name>: failed to close '<filename>': Invalid argument
```

No error is displayed on Windows systems. The target moved or renamed file is corrupted and should be deleted. The target file is also flagged with the `xattr_error` flag on Linux and `Rekey Status Excluded` on Windows. This flag prevents subsequent read/write access to the file. You can check the CTE-LDT attributes for the presence of this flag. See [About the Exclusion Attribute for Files Matching an Exclusion Key Rule](#).

Also, a log entry is sent to the CipherTrust Manager on Linux systems when this occurs. For example, if you moved the versioned file `/gp/foo.txt` into the GuardPoint `/gp/subdir/foo.txt` with an exclusion key rule that excludes matching files with the `clear_key`, the following log message would be created on the CipherTrust Manager:

```
[CGA] [WARN ] [29261] [CGS3268W] LDT-ALERT: encrypted data detected in filename [foo.txt] inode [35720037] in guard point [/gp] under clear exclusion key rule
```

Overlapping Exclusion Key Rules

Multiple exclusion key rules in the same policy may overlap each other. For example on Linux, if the non-versioned key `Key_A` is associated with resource `/oxf-fs1/gp1/Folder_Enc_With_KEY_A` and the non-versioned key `Key_B` is associated with resource `*.txt`, placement of the file `/oxf-fs1/gp/Folder_Enc_With_KEY_A/foo.txt` overlaps the two key rules. In such a case, the first rule in the policy is enforced on `/oxf-fs1/gp/Folder_Enc_With_KEY_A/foo.txt` when the file is created and in subsequent file access.

On Windows, if the non-versioned key `Key_A` is associated with resource `c:\oxf-fs1\gp1\Folder_Enc_With_KEY_A` and the non-versioned key `Key_B` is associated with resource `*.txt`, then they would overlap on the file `c:\oxf-fs1\gp\Folder_Enc_With_KEY_A\foo.txt`. In such a case, the first rule in the policy is enforced on `c:\oxf-fs1\gp\Folder_Enc_With_KEY_A\foo.txt` when the file is created and in subsequent file access.

Application of Exclusion Key Rules on GuardPoint over NFS

When protecting files using an Exclusion Key Rule with `clear_key` in local file systems, files can be accessed in clear even if the GuardPoint is not enabled. Clear access is possible because data is not encrypted in such files and the LDT metadata is managed as part of the `secfs` extended attribute for such files. Clear access is not possible for such files in a GuardPoint over NFS because LDT metadata is part of the `secfs` extended attribute that is stored in the first 4096 bytes of each file. CTE skips the embedded `secfs` header when reading or writing such files when a GuardPoint is enabled. With a GuardPoint disabled, access to files is not through CTE and reading or writing files in a GuardPoint will not skip the embedded `secfs` attribute.

Caution About Applications That Create Temporary Files (Windows)

Some applications on Windows create a temporary file version of the original file when you open and modify a file. This behavior can affect how you implement exclusion key rules.

If you have an exclusion key rule that uses a file extension to exclude files that may be opened and modified by such an application, exclude the temporary file name extension also. If you don't exclude the temporary file, the temporary file may be encrypted by another policy that matches the temporary file extension. Then the original file, which is copied from the temporary file, will be unreadable. Also keep in mind that other applications that may create temporary files with the same file extension and consider what policies should affect those temporary files.

This situation can happen with Microsoft Office files such as the `.docx` files used by Microsoft Word. When you open and modify a `.docx` file, Word creates a `.tmp` file version of that file. So for Microsoft Word you should add `*.tmp` to the exclusion key rule resource set if you add `*.docx`.

Rename Operations Crossing Key Rules (Linux)

On Linux, when a rename operation crosses key rules, the rename operation copies the source file to a new file using the new name and removes the original file. If the

source file is flagged for exclusion key rule property, the target new file is disassociated with the exclusion key rule if the new name no longer matches the resource set associated with exclusion key rule. For more information, see [About the Exclusion Attribute for Files Matching an Exclusion Key Rule](#).

Using Linked Files with Exclusion Key Rules (Linux)

On Linux, do not create multiple hard links to the same target file, such that the pathname of each hard link is associated with a resource set of an exclusion key, and the key rules have different keys. Accessing the file through the pathname of each hard link results in a different key being applied to the file, resulting in data corruption in the target file due to the application of multiple keys to the same data.

If you must create hard links, be sure the hard link pathnames and the target file are within the same resource set.

Changing a Folder or Files from Versioned to Non-Versioned Key (Windows)

Exclusion key rules provide a way to convert a subset of guarded files or the contents of a folder from being encrypted by a versioned key to being encrypted by a non-versioned key. After this conversion, the excluded files or directory contents will be encrypted at the last version of the versioned key but the encryption keys for those items will not rotate to a new version when the keys for other non-excluded files are rotated. In other words, the excluded files or folder contents will be encrypted by a non-versioned key.

Follow these steps to change selected files or folder contents to a non-versioned key:

1. In the CipherTrust Manager, clone the versioned key that is used in the policy that you plan to edit. Cloning a key creates a non-versioned copy of the existing version of the versioned key.
2. In the CipherTrust Manager, disable the GuardPoint. Disabling the GuardPoint is required before modifying a policy applied in that GuardPoint.

- Configure one or more resource sets to match the files that you want to exclude. For example, the resource set `star-dot-text` could contain `*.txt` and the resource set `sales-folder` could contain `\sales*`.
- Add one or more exclusion key rules to convert matching files to a non-versioned key. In the following example, files matching `*.txt` and files in the `sales` folder (as defined in resource sets) will be converted from the current version of the versioned key `AES256_versioned` to `AES256_clone`, assuming `AES256_clone` is a clone of `AES256_versioned`.

Order	Resource	Current Key	Transformation Key	Exclusion Rule
1	star-dot-txt	AES256_clone	AES256_clone	Y
2	sales-folder	AES256_clone	AES256_clone	Y
3		clear_key	AES256_versioned	N

Exclusion key rules must be ordered before other rules.

- On the command line, run the following command to remove the CTE-LDT metadata from the files that you want to convert from versioned key to non-versioned key encryption:

```
voradmin ldt attr delete <path_to_files>
```

To recursively delete the CTE-LDT metadata from all files matching a pattern in all subfolders, use the following form, including `*` as a wildcard where needed:

```
voradmin ldt attr delete <path_to_files> -filter
<filename>.<extension>
```

Note

Be careful when deleting the CTE-LDT metadata from files. If you delete the metadata from a file that does not match an exclusion key rule policy, the file will be unreadable after the next rekey.

Given the example exclusion key rule described in step 4, you would need to run this command on all files with the extension `.txt` (first example below) and on the files in `\sales` (second example below).

```
voradmin ldt attr delete c:\gp1 -filter *.txt
voradmin ldt attr delete c:\gp1\sales
```

For more information about using `voradmin` on CTE-LDT metadata, run `voradmin ldt attr /?` on the command line.

6. In the CipherTrust Manager, re-enable the GuardPoint that includes the new exclusion key rule.
7. In the CipherTrust Manager, rotate the key for the policy that includes the new exclusion key rule.

Using the example exclusion key rule in step 4, after completing this procedure, files matching `*.txt` and files in the `sales` folder will have the exclusion attribute set and will be excluded from rekeying (see [About the Exclusion Attribute for Files Matching an Exclusion Key Rule](#)). Files not matching the exclusion key rule will be rekeyed to the next version of the `AES256_versioned` key.

To perform a similar conversion on Linux, see [Adding an Exclusion Key Rule to an Existing Policy with Versioned Keys \(Linux\)](#).

Examples of Exclusion Key Rules

This section describes some examples of how to use an exclusion key rule.

Encrypt Files With Exclusion Property Using a Non-Versioned Key

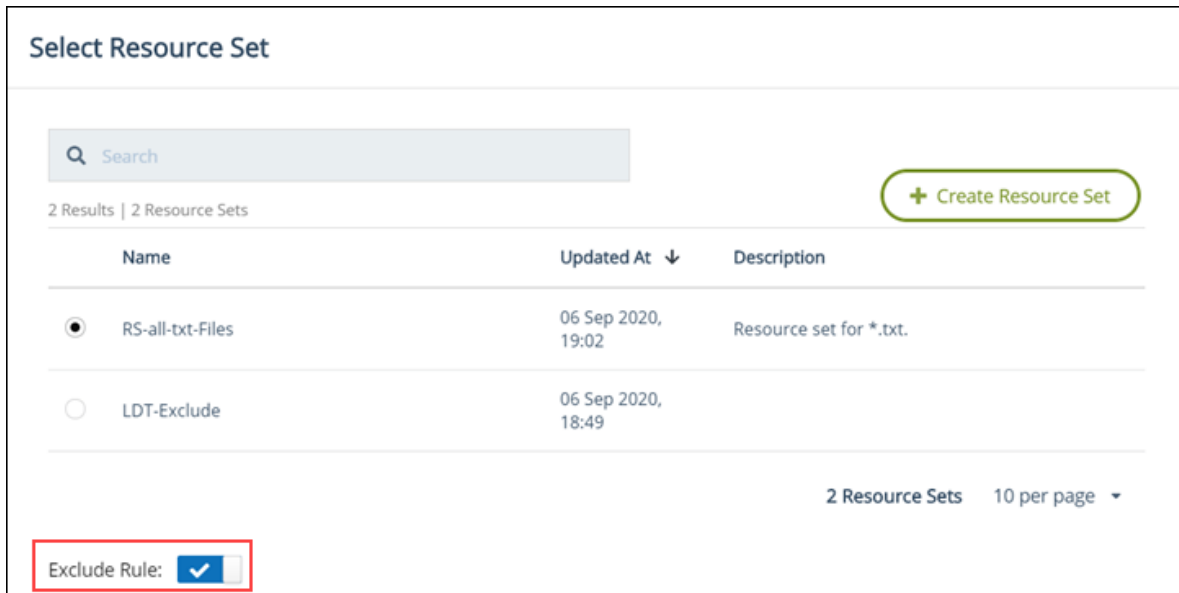
The following exclusion key rule applies the non-versioned key, `Key_TextFiles`, to any new files that are created with a `*.txt` extension.

Note

Existing `*.txt` files in the GuardPoint during the initial CTE-LDT data transformation process are assumed to be already encrypted with the same key that you specify in the exclusion key rule. These existing files are not transformed during the initial encryption or during any subsequent rekeys. The key you specify in the exclusion key rule is applied to new `*.txt` files only.

In the Policy, do the following:

1. Add a Key Rule.
2. In the **Resource Set** field, click **Select**.
3. Select a resource set that specifies *.txt in the File field and click the Exclude Rule** option.



4. Click **Select**. CipherTrust Manager displays the resource set name in the Resource Set field with an icon indicating that this is an Exclusion Rule.
5. Select the `Key_TextFiles` key in the **Current Key Name** field.
6. Click **Add**.
7. Add any other key rules or exclusion key rules to the policy that you want. You must add at least one non-exclusion key rule that specifies the current encryption key and the versioned encryption key that you want to use with this Live Data Transformation policy.
For example, you could add another exclusion key rule that specifies all **.doc** files should be encrypted with the key `Key_DocFiles`, or one that specifies all **.zip** files should be left unencrypted by specifying the key `clear_key`.

Exempt Excluded Files from Encryption (Set to `clear_key`)

The following exclusion key rule sets all files in the resource set `/oxf-fs1/gp1/Clear_Files_Folder` (Linux) or `\oxf-fs1\gp1\Clear_Files_Folder` (Windows) to `clear_key`

(in other words, not encrypted). Files in other directories that match other key rules in the same policy may be encrypted. This could allow unrestricted access to the files in `/oxf-fs1/gp1/Clear_Files_Folder` (Linux) or `\oxf-fs1\gp1\Clear_Files_Folder` (Windows) while access may be restricted to files in parallel directories.

- Linux Exclusion Key Rule: The Resource Set **Directory** field should contain `/oxf-fs1/gp1/Clear_Files_Folder`, and the **Key** should be `clear_key`.
- Windows Exclusion Key Rule: The Resource Set **Directory** field should contain `\oxf-fs1\gp1\Clear_Files_Folder`, and the **Key** should be `clear_key`.

Listing All Files Included in an Exclusion Key Rule (Linux)

Determining the files that match an exclusion key rule involves two steps. You list all the keys in a GuardPoint, choose the key that you're interested in, and then run a command to list the files that match that key. This process works for both standard key rules and exclusion key rules.

1. Decide on the GuardPoint that you want to check for excluded files. For example, for the GuardPoint `/oxf-fs1/gp1`, type the following on the command line:

```
voradmin ldt key report /oxf-fs1/gp1
LDT_KEY1,1
LDT_KEY2,2
LDT_KEY3,5
NON_VERSIONED_KEY
```

The number after the comma is the key version number. See [Key Report Option](#) for more information about the `voradmin` key report.

2. From the key report output, choose the key rule for which you want to list matching files. For example, to see the files associated with the `NON_VERSIONED_KEY` in the previous step, you would type:

```
voradmin ldt key map NON_VERSIONED_KEY /oxf-fs1/gp1
/oxf-fs1/gp1/file1.dat10
/oxf-fs1/gp1/file2.dat10
/oxf-fs1/gp1/file3.dat10
```

```
/oxf-fs1/gp1/file4.dat10  
/oxf-fs1/gp1/file5.dat10
```

See [Key Map Option](#) for more information about the `voradmin` key map report.

Listing All Files Included in an Exclusion Key Rule (Windows)

Determining the files that match an exclusion key rule involves two steps. You list all the keys in a GuardPoint, choose the key that you're interested in, and then run a command to list the files that match that key. This process works for both standard key rules and exclusion key rules.

1. Decide on the GuardPoint that you want to check for excluded files. For example, for the GuardPoint `c:\gp1`, type the following on the command line:

```
C:\> voradmin ldt key report c:\gp1  
Keys used for GP, c:\gp1 :  
clear_key,0  
CS1-CTE-LDT-AES256,11
```

The number after the comma is the key version number. The `clear_key` key is not versioned, so the version number is 0.

2. From the key report output, choose the key rule for which you want to list matching files. For example, to show the files associated with the `clear_key` listed above, you would type:

```
C:\> voradmin ldt key map c:\gp1 clear_key,0  
Files rekey with key (clear_key,0)-  
c:\gp1\file1.txt  
c:\gp1\file2.txt  
c:\gp1\file3.txt  
c:\gp1\file4.txt  
c:\gp1\file5.txt
```


Dynamic Resource Sets (Inclusion of New Key Rules)

CTE v7.6, and subsequent versions, support inclusion of a new resource set in a new key rule in an LDT policy already applied to GuardPoint. The new key rule allows LDT to launch and rekey the files associated with the resource set. Before inclusion of the key rule, the files associated with the resource set were in clear-text. Following is an example of a policy applied to GuardPoint `/oxf-fs1/gp1` which includes subdirectories `dir1`, `dir2`, `dir3`, `dir4`, and `dir5`.

The screenshot shows the configuration page for the LDT-DYNRES-POL policy. It displays two key rules in a table:

Order	Resource Set	Current Key Name	Transformation Key Name	Exclusion Rule
1	dir1	clear_key	LDT-KEY-1	No
2	dir2	clear_key	LDT-KEY-2	No

At the bottom of the table, it indicates "2 Key Rules" and "10 per page".

Applying this policy to `/oxf-fs1/gp1` will launch and rekey the files under `/oxf-fs1/gp1/dir1` and `oxf-fs1/gp1/dir2` using the specified key, while the remaining files in `/oxf-fs1/gp1` remain in clear-text. To rekey the files under `/oxf-fs1/gp1/dir3` and `/oxf-fs1/gp1/dir4`, you can update the policy as follows:

The screenshot shows the updated configuration page for the LDT-DYNRES-POL policy. It now displays four key rules in a table:

Order	Resource Set	Current Key Name	Transformation Key Name	Exclusion Rule
1	dir1	clear_key	LDT-KEY-1	No
2	dir2	clear_key	LDT-KEY-2	No
3	dir3	clear_key	LDT-KEY-2	No
4	dir4	clear_key	LDT-KEY-3	No

At the bottom of the table, it indicates "4 Key Rules" and "10 per page".

Once the updated policy is applied to the GuardPoint, the rekey is launched to encrypt the files associated with `/oxf-fs1/gp1/dir3` and `/oxf-fs1/gp1/dir4` using the encryption keys `LDT-KEY2` and `LDT-KEY3`, respectively. Remaining files in the GuardPoint remain unchanged.

Temporary Exclusion of files in Dynamic Resource Set

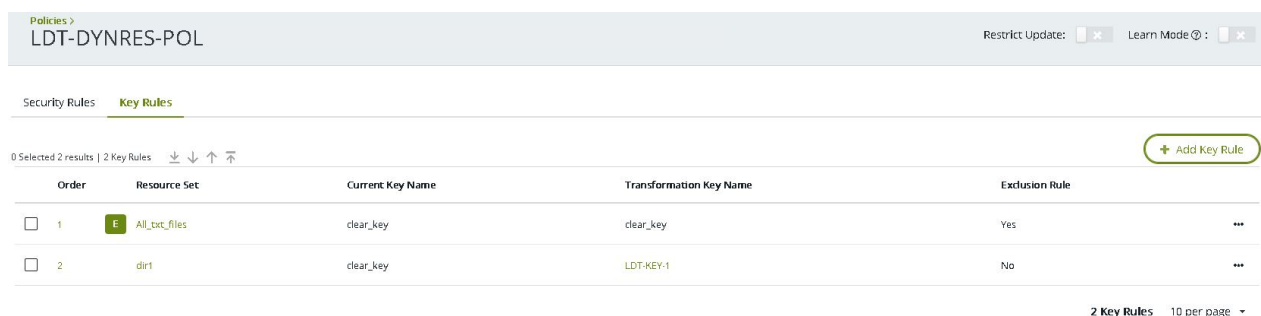
Files associated with Resource Sets included in the key rules are encrypted with the key rule. Other files are implicitly excluded from rekey. LDT enforces the exclusion property on the files not included for rekey in any key rule. However, the exclusion property is only enforced until a new non-exclusion key rule is added that covers the implicitly excluded files. For example, the following output shows the exclusion property on the file `/oxf-fs1/gp1/dir5/foo.txt` which is a file not included in any key rule, in the policy under the previous screen shot. As illustrated, the file is temporarily excluded from rekey.

```
# voradmin ldt attr get /oxf-fs1/gp1/dir5/foo.txt
LDT attributes: rekeyed_size=0, rekey_status=rekey_excluded
Key: name=clear_key, version=none
```

The exclusion property will be cleared on the file, and the file will be encrypted, once a new key rule with a defined resource set for dir5 is included in the policy.

Conflicts with Resources under Exclusion Key Rules

Adding a new key rule for rekeying a resource set, in a policy that already includes a key rule with the exclusion property, may result in a conflict. For example:



The screenshot shows the configuration page for the policy 'LDT-DYNRES-POL'. It displays two key rules in a table:

Order	Resource Set	Current Key Name	Transformation Key Name	Exclusion Rule
1	All_txt_files	clear_key	clear_key	Yes
2	dir1	clear_key	LDT-KEY-1	No

Before adding the second key rule, the effect of the initial rekey is explicit exclusion of all `*.txt` files and implicit exclusion of other files. For example,

`/oxf-fs1/gp1/dir1/my_file` is temporarily excluded from rekey after the initial rekey:

```
# voradmin ldt attr get /oxf-fs1/gp1/dir1/my_file
LDT attributes: rekeyed_size=10485760, rekey_status=rekey_excluded
Key: name=clear_key, version=0
```

The effect of the rekey, after adding the second key rule, is encryption of the files under `/oxf-fs1/gp1/dir1` and the exclusion of all `*.txt` files from the rekey. If the resource set in the second key rule includes files that also belong to the resource set under the first key rule, then all of the `*.txt` files remain excluded, due to the first key rule. However, other files (non `*.txt`) are not excluded due to the second key rule. This rekey process encrypts only those files that were implicitly excluded from rekey prior to the addition of the second key rule. For example, inclusion of the second key rule encrypts `/oxf-fs1/gp1/dir1/my_file` and removes exclusion from the file:

```
# voradmin ldt attr get /oxf-fs1/gp1/dir1/my_file
LDT attributes: rekeyed_size=10485760, rekey_status=none
Key: name=LDT-KEY-1, version=0
```

Once the exclusion property is removed, the restrictions enforced under the exclusion property will no longer be in effect.

Renaming files can also result in conflicts. For example, relocating `/oxf-fs1/gp1/dir1/foo.txt` to another path breaks the association of `foo.txt` with `/dir1`. The resource set will only re-enforce the exclusion property under the first key rule.

Note

Avoid key rules, or file operations, that may result in conflicts in enforcement of the exclusion property or breaking the association of the files with the intended resource sets.

Limitations with Dynamic Resource Sets

1. Users cannot change the order of the rule, or the encryption key, applied to a file.
2. Inclusion of dynamic resource-set may clear the exclusion flag on files shared under the exclusion key rule and the dynamic resource set.

- Files which are present in the excluded resource-set, if moved to a directory which is encrypted, will result in encryption of the file on the next policy push (either next key rotation or any update made in the policy or policy elements).
- Support for Dynamic resource sets **does not** include GuardPoints over NFS.

Using CTE-LDT with SAP HANA Fibre Channel Systems (Linux Only)

You must add additional CTE commands to the HANA administrator entry. To do so, edit the `/etc/sudoers` with a text editor and add entries for `/usr/bin/voradmin` and `/usr/bin/vmsec`:

Example:

```
hanadm ALL=NOPASSWD:  
/usr/bin/secfsd,/usr/bin/voradmin,/usr/bin/vmsec,/sbin/multipath,/sbin/  
multipathd,/etc/init.d/multipathd,/usr/bin/sg_persist,/bin/mount,/bin/  
umount,/bin/kill,/usr/bin/lsof,/sbin/vgchange,/sbin/vgscan
```

If you are using an ext3 file system, you must mount it with extended attributes. To do so, edit the storage section of the `global.ini` file using a text editor and add the following lines:

```
partition_*_data__mountOptions = -o user_xattr  
partition_*_log__mountOptions = -o user_xattr
```

CTE-LDT Administration

This section contains the following topics:

- [CTE-LDT Metadata in Extended Attributes](#)
- [DFS\(R\) and Replication \(Windows\)](#)
- [Backing Up and Restoring CTE-LDT GuardPoints](#)
- [CTE-LDT Command-Line Administration: voradmin command](#)
- [Migrating a GuardPoint to a Different CTE-LDT Policy](#)

- [Removing CTE-LDT and Security Encryption](#)
- [Uninstalling the Agent while CTE-LDT is Rekeying GuardPoints](#)
- [Detecting Loss of NAS connection to an LDT GuardPoint Group](#)

CTE-LDT Metadata in Extended Attributes

An *extended attribute* is a name/value pair permanently associated with a file or directory stored in a file system. CipherTrust Transparent Encryption (CTE) creates and maintains its own user extended attributes on CTE-LDT GuardPoint directories and files. The extended attributes are used to store metadata related to each file or directory that is protected using an CTE-LDT policy.

On Linux, CTE-LDT sets extended attributes on GuardPoint directories. The CTE-LDT attribute of an CTE-LDT GuardPoint stores the following metadata:

- Current key version.
- Rekey status.
- Rekey start and end times.
- Estimated completion time.
- Total amount of data transformed.
- Total number of files transformed.
- Current key signature and applied key signature.

On both Linux and Windows, CTE-LDT sets extended attributes on files. The CTE-LDT attribute of a file stores the following metadata:

- Name of the current key.
- Name of the versioned key.
- Version number of the versioned key.
- CTE-LDT rekey status of the file.

In most cases, the current and new key names are the same. The exception is during initial transformation from a legacy policy to an CTE-LDT policy, when the file has been encrypted with the current key and is being transformed to the current version of the transformation key.

Note

Before you set up a GuardPoint for CTE-LDT, ensure that there is sufficient disk space available in your file system for CTE-LDT metadata. The amount of disk space you need depends on the number of files in your GuardPoint. For more information about the disk space requirement, see [Planning for CTE-LDT Attribute Storage](#).

The state of a file changes during CTE-LDT operations. The extended attributes are continually updated to reflect the current file status, which falls into one of the following categories:

- Rekeyed to the current version of the key.
- Rekeyed to the previous version of the key, or the initial key state (before the first CTE-LDT rekey has been performed).
- Partially rekeyed, where some regions of the file are rekeyed to the new key version and other regions are still keyed to the previous key version or the initial key.

Listing Extended Attributes

You can list extended attributes of files by using native operating system commands, or system calls. As part of GuardPoint administration, CTE can modify or delete extended attributes.

Note

This functionality is only available for local file systems. It is not supported for files in NFS Share GuardPoints.

In Linux, CTE-LDT attributes are set on GuardPoint directories and regular files in GuardPoint directories protected with CTE-LDT policies. The CTE extended attribute name is `::secfs:xattr:`.

The following examples use the native Linux operating system command `attr` to display the CTE-LDT attribute for the GuardPoint `/oxf-fs1/gp1` and the file `/oxf-fs1/gp1/File_1.txt`.

Example Getting File Attributes

```
attr -l /oxf-fs1/gp1/File_1.txt
Attribute "::secfs:xattr:" has a 1044 byte value for /oxf-fs1/gp1/
File_1.txt
Attribute "selinux" has a 37 byte value for /oxf-fs1/gp1/File_1.txt
```

Example Getting GuardPoint Attributes

```
attr -l /oxf-fs1/gp1
Attribute "::secfs:xattr:" has a 1044 byte value for /oxf-fs1/gp1
Attribute "selinux" has a 37 byte value for /oxf-fs1/gp1
```

Example of `voradmin ldt attr get` for Linux File Attributes

In the following example, the file `/oxf-fs1/gp1/File_1.txt` has the same name for current and new keys at the same key version. In the following example, if the versioned key `LDT_KEY` is at version 755, the file is rekeyed to the latest key version under the CTE-LDT policy.

```
voradmin ldt attr get /oxf-fs1/gp1/File_1.txt
CTE-LDT attributes: rekeyed_size=4096, rekey_status=none
Key:      name=LDT_KEY, version=755
```

Example of `voradmin ldt attr get` for Linux GuardPoint Attributes

The following is example of an CTE-LDT attribute on a GuardPoint directory on Linux:

```
voradmin ldt attr get /oxf-fs1/gp1
LDT stats: version=1, rekey_status=rekeying
  Number of times rekeyed:      3 times
  Rekey start time:             2018/08/04 16:24:45
  Last rekey completion time:   2018/07/04 16:24:04
  Estimated rekey completion time: N/A
  Policy key version:           2043
Data stats:
  total=3.3GB, rekeyed=1.5GB, truncated=0.0MB
File stats:
  total=4307, rekeyed=1181,
  passed=2, skipped=0, created=0, removed=0
```

Example of `voradmin ldt attr get` for Linux NFS Share GuardPoint Attributes

The following example shows how to use the `voradmin ldt attr get` command to view the LDT attribute on GuardPoint directories over NFS shares:

```
secfsd -unguard /nfs-oxf-fs1/gp2
secfsd: Path is not guarded
voradmin ldt attr get /nfs-oxf-fs1/gp2
LDT stats: version=3, rekey_status=rekeyed
    Number of times rekeyed:                1 time
    Rekey start time:                       2021/01/04 08:19:02
    Last rekey completion time:             2021/01/04 08:19:03
    Estimated rekey completion time:        N/A
    Policy key version:                     0
    Policy ID:
        23785
    Data stats:
        total=0.0MB, rekeyed=0.0MB
        truncated=0.0MB, sparse=0.0MB
    File stats:
        total=3, rekeyed=1, failed=0
        passed=0, skipped=0, created=0, removed=0, excluded=0
```

Example of `voradmin ldt attr get` for Windows GuardPoint Attributes

The attribute for the GuardPoint `c:\GP 1` contains the status (rekeyed) and statistics specific to the GuardPoint and CTE-LDT. Following is sample output of `voradmin` command on Windows for statistics on a file:

```
C:\> voradmin ldt attr get c:\GP\Test.txt
LDT attributes:
    Rekey Status          Rekeyed
    Initial Rekeyed Size  10 Bytes
Key:
    Key Name/Version      (LDT_KEY, 28)
```

The attribute for GuardPoint `C:\GP` contains the status (rekeyed) and statistics specific to the GuardPoint and CTE-LDT:

```
C:\> voradmin ldt attr get c:\gp\
LDT Stats
```



```

-----
Rekey Status                                LDT_ST_REKEYED
Last rekey completion time                  10/2/2017 4:26:50
Rekey Start time                            10/2/2017 4:26:17
Estimated rekey completion time            000:00:00

```

File Stats:

```

Total          444
Rekeyed        444
Skipped        0
Errored        0
Passed         0
Removed        0

```

Data Stats:

```

Total          11 GB (12649143417 Bytes)
Rekeyed        11 GB (12649143417 Bytes)
Truncated      0 Bytes

```

Example of `voradmin ldt attr get` for Windows CIFS Share GuardPoint Attributes

The following example shows how to get the CTE-LDT attributes for the CIFS GuardPoint `\\myhost\share\HR-Files\`.

```
C:\>voradmin ldt attr get \\myhost\share\HR-Files\
```

Live Data Transformation Stats

```

-----
Rekey Status                                LDT_ST_REKEYED
Last rekey completion time                  2/24/2021 13:42:40
Rekey Start time                            2/24/2021 13:36:47
Estimated rekey completion time            000:00:00

```

File Stats:

```

Total          19087
Rekeyed        19087
Skipped        0
Errored        0

```

```
Passed      0
Removed     0
Excluded    0
```

Data Stats:

```
Total      1 GB (1083187108 Bytes)
Rekeyed     958 MB (1005006756 Bytes)
Truncated   0 Bytes
```

Example of `voradmin ldt attr get` for Windows CIFS Share File Attributes

The following example shows how to get the CTE-LDT attributes for the file `employees.doc` on the CIFS share `\\myhost\share\HR-Files\`.

```
C:\>voradmin ldt attr get \\myhost\share\HR-Files\employees.doc
```

LDT attributes:

```
Rekey Status          Rekeyed
Initial Rekeyed Size  0 Bytes
```

Key:

```
Key Name/Version      (AES_256_LDTKey_CBC,15)
```

MDS File (Linux)

In addition to CTE-LDT attributes, the CTE-LDT process on Linux requires persistent storage for additional metadata related to encrypting, or rekeying, files in GuardPoints. CTE-LDT allocates the storage space as soon as the CTE-LDT process starts on a GuardPoint. It maintains this storage space during the entire transformation process, until the GuardPoint is completely transformed.

Storage for this metadata is allocated and managed in a special file, called the MDS (metadata store) file. The MDS file resides inside a GuardPoint directory so each GuardPoint has its own MDS file.

The MDS file is a CTE protected file with the name `__vorm_mds__`. For example:

```
ls -l /oxf-fs1/gp1/__vorm_mds__
-rwxr-xr-x. 1 root  root 31754474496 Dec  8 09:09 /oxf-fs1/
__vorm_mds__
```

```
du -B 1024 /oxf-fs1/gp1/___vorm_mds___
25056 /oxf-fs1/___vorm_mds___
```

As shown above, the MDS file is sparse. In the example, the file size is approximately 30GB, however the file is allocated with approximately 25MB of disk storage. CTE-LDT automatically creates the MDS file the first time the CTE-LDT process starts on any GuardPoint in the file system namespace. It populates the MDS file with all of the metadata for the GuardPoint at the beginning of the CTE-LDT process. Disk space allocated to the MDS file is freed and the MDS file in the GuardPoint directory is removed when the CTE-LDT process completes on the GuardPoint.

Warning

The MDS file is protected. You cannot remove it unless the administrator runs the `voradmin` command to manually remove the MDS file once it is no longer needed. See [Deleting CTE-LDT Metadata \(Linux\)](#) for more information.

CTE-LDT automatically allocates and de-allocates disk space for the MDS file as part of the CTE-LDT process. De-allocation of disk space for a GuardPoint does not change the MDS file size, although it frees the disk blocks. MDS files are sparse and very large in size. The MDS file is automatically removed from GuardPoints when the files have been successfully rekeyed.

Note

The metadata storage (MDS) file name has changed from `___vorm_mds___` to `___vorm_mds___`. GuardPoints will automatically rename the MDS file when guarding with a CTE 7.6.0 agent. LDT NFS GuardPoints will rename once all of the hosts, with the same share guarded, have been upgraded.

Planning for CTE-LDT Attribute Storage

Before a GuardPoint is enabled for CTE-LDT, make sure that there is sufficient free disk space in the file system to which the GuardPoint belongs. Free space is required for CTE-LDT attributes and (in Linux) metadata in the MDS file. CTE-LDT attributes are created during the initial encryption and are never freed until the GuardPoint is

permanently unguarded and removed from the protection of an CTE-LDT policy. In contrast, disk space for metadata in the MDS file is temporary, kept only during the live transformation process.

When planning how much free disk space to reserve for CTE-LDT on a GuardPoint, consider the following:

- Number of files in the GuardPoint
- (Linux) Average length of absolute pathnames of files in the GuardPoint

The CTE-LDT process pre-allocates disk space for the Linux MDS file based on a minimum of 200K files with an average pathname of 1024 bytes per GuardPoint. The minimum space amounts to 325MB of disk space for the MDS file for each GuardPoint, even if file count is very low. (In Windows, CTE-LDT reserves the space when the file is rekeyed.)

Using voradmin To Estimate Disk Space Required for CTE-LDT

In **Linux**, you can use the `voradmin ldt space` command to estimate the amount of disk space required for CTE-LDT attributes and the MDS file. The result is rounded to the nearest MB. The syntax of the command is:

```
voradmin ldt space <guard path>
```

Note

In **Windows**, LDT requires 10% of the underlying file system space, or a minimum of 2GB disk space, for LDT to start.

The following example shows how to estimate the required disk space to perform CipherTrust Transparent Encryption - Live Data Transformation on 1501 files in the GuardPoint `/oxf-fs1/gp1`. Estimate the disk space before protecting `/oxf-fs1/gp1` using an CTE-LDT policy. For example:

```
voradmin ldt space /oxf-fs1/gp1
/oxf-fs1/gp1: found 1501 files without CTE-LDT extended attributes
```

```
CTE-LDT disk space requirements: total 169MB (CTE-LDT attributes=6MB,
MDS=163MB)
```

The voradmin command reports that 1501 files in `/oxf-fs1/gp1` without CTE-LDT attributes. These files are new to CTE-LDT. 6MB of space is required for the CTE-LDT extended attributes and 163MB for metadata in the MDS file.

The following example shows the output of the same command after encryption completes. This estimates the additional disk space needed for the next CTE-LDT rekey operation. For example:

```
voradmin ldt space /oxf-fs1/gp1
/oxf-fs1/gp1: found 0 files without CTE-LDT extended attributes
CTE-LDT disk space requirements: total 163MB (CTE-LDT attributes=0MB,
MDS=163MB)
```

The voradmin command still reports on the same 1501 files, but because encryption has occurred using CTE-LDT, the files all have their CTE-LDT attributes. No additional space is required for CTE-LDT extended attributes on the next run. However, since the MDS file is transient and deletes after it finishes encrypting the GuardPoint, it requires additional space for the next key rotation.

Note

Windows estimates space using the following equations:

```
Permanent Space = Number of Files * 4K
```

```
Temporary Space = 128K * (Number of CPU * 2)
```

Displaying Metadata

Use the following command to see CTE-LDT file attributes or GuardPoint attributes:

```
voradmin ldt attr get [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see [CTE-LDT Command-Line Administration: voradmin command](#).

Verifying Metadata (Windows only)

Use the following command to verify if the metadata is corrupt:

```
voradmin ldt attr verify [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see [CTE-LDT Command-Line Administration: voradmin command](#).

DFS(R) and Replication (Windows)

The Distributed File System Replication (DFS(R)) service is a multi-master replication engine used to maintain synchronized folders on multiple servers. Replicating data to multiple servers increases data availability and provides users in remote sites with fast, reliable access to files.

If you are creating GuardPoints in a DFS(R) environment, you must first add the DFS(R) private folder guard path to the exclusion list in the Windows Registry. CTE-LDT should not attempt data transformation on this read-only directory.

1. Using the Registry Editor, or the Windows command line, add a registry entry in:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Vmmgmt\Parameters` named "LDTEExclusionGPList" of type REG_MULTI_SZ.

2. Add the `DfsrPrivate` and `System Volume Information` directory path to the `LDTEExclusionGPList`.

For example, if the DFS(R) private directory path is `D:\data\DfsrPrivate`, add this string in LDTEExclusionGPList.

3. Reboot the system to make the change take effect.

Note

CTE-LDT does not perform a rekey on the DFS(R) private directory. Its rekey status is always "N/A".

If an application, or user, is performing a rename or a delete folder operation inside a GuardPoint with an CTE-LDT policy, this may restart the rekey process. Files which are already rekeyed will not be rekeyed again.

- If a rename or a delete operation is already performed on rekeyed files/folder, then the rekey process does not restart.
- If a rename or delete operation is performed in a folder where a rekey is in progress, CTE-LDT needs to stop the rekey process and restart the rekey again.

Backing Up and Restoring CTE-LDT GuardPoints

This section describes procedures and considerations related to backing up and restoring data in CTE-LDT GuardPoints.

In addition to the files in a GuardPoint, CTE-LDT stores metadata in extended file attributes. These CTE-LDT attributes contain information that is required for decrypting the files and for the proper operation of CTE-LDT. Therefore, it is critical that your backup application also backs up the extended attributes of the files along with file data.

Related Topics

- [Clear Text Backup and Restore](#)
- [Encrypted Backup and Restore](#)
- [CTE-LDT Policy for Encrypted Backup and Restore](#)
- [Backup/Restore of Metadata Store File \(MDS\) in GuardPoints Undergoing Rekey](#)
- [Restore an Encrypted Backup](#)
- [Restoring Non-CTE-LDT Backup Data to an CTE-LDT GuardPoint](#)
- [Using fsfreeze \(Linux only\)](#)
- [CTE-LDT Backups Using a File System or Storage-Level Snapshot Tool](#)
- [Windows Backup and Snapshots](#)
- [CTE-LDT Backup and Restore Troubleshooting](#)

Note

For backing up and restoring a CTE LDT over NFS GuardPoint, see [Backup and Restore LDT on NFS GuardPoints](#).

Clear Text Backup and Restore

A policy that applies a Security Rule with the Apply Key effect on backup/restore operations does not require any special rules for data access by backup applications. Under such a policy, backup applications always read clear data and store clear data in backup media. The backup application is not required to back up the CTE-LDT extended attributes, and a QoS schedule is not required to suspend CTE-LDT during backups.

Encrypted Backup and Restore

Note

Only the host designated as the primary host for an LDT GuardPoint Group can perform encrypted backup or restore operations on GuardPoints over NFS.

When a policy does not enforce a Security Rule with the Apply Key effect on backup/restore operations, the policy does not decrypt data on I/O operations from that backup application. Under such a policy, the backup application stores encrypted data and the CTE-LDT extended attributes of the file on the backup media.

In Linux, CTE-LDT operations must be suspended during backup. Suspending CTE-LDT completes the ongoing rekey operations on regions of files before starting the backup. During live transformation, CTE-LDT first preserves those regions of a file to be rekeyed in the MDS file. Then it updates some of the metadata in order to update the status of the data preserved in the MDS file in preparation for the rekey. Then it starts rekeying and updating those regions in the underlying file.

Suspending CTE-LDT waits for ongoing rekey operations to complete, and saves the metadata in the CTE-LDT extended attribute section of the MDS file. Suspending CTE-LDT ensures that the rekey status stored in the CTE-LDT extended attribute accurately reflects the rekey status of the data in the entire file during backup.

Note

This requirement does not apply to CTE-LDT for Windows.

The following table summarizes the state of the data in files in backup media:

CTE-LDT State of File	Security Policy	Backup Metadata or Alternate Data Streams Along with File data	Effect
Not rekeyed	Permit	Yes	Data in backup may be in clear format or encrypted with older key version.
Rekey in progress	Permit	Yes	File in backup storage is partially rekeyed. Some parts are in clear format or encrypted with older key version, and other parts are encrypted with current key version.
Rekey complete	Permit	Yes	File in backup storage is in the encrypted format with new key version.

CTE-LDT Policy for Encrypted Backup and Restore

Suppose you have an CTE-LDT policy allowing a backup user, or a backup process, to perform an encrypted backup. The backup user, or the backup process, reads encrypted data from files and stores the same encrypted data in backup media.

For example, suppose you wanted to back up GuardPoint `/oxf-fs1/gpl` protected by `My_LDT_Policy_1`. The key version before the key rotation is 8. The following steps occur:

1. Key version 8 expires and is rotated to version 9.
2. Rotating the encryption key triggers CTE-LDT to start running on the data under the GuardPoint.
3. The QoS schedule suspends CTE-LDT because a backup process is running.

4. The backup process begins, runs and later ends.
5. The QoS schedule resumes CTE-LDT.
6. CTE-LDT on the GuardPoint completes.

Note

Although the backup/restore scenarios and examples described in the rest of this section are specific to CTE-LDT on Linux platform, the concepts also apply to CTE-LDT on the Windows platform.

Consider the state of the three files in the GuardPoint during the CTE-LDT process, right after CTE-LDT is suspended in preparation for the backup. You can obtain the state of each file through the `voradmin ldt attr get` command, which examines the CTE-LDT state of each file captured in the CTE-LDT extended attributes of those files. (For an overview of `voradmin CTE-LDT`, see [CTE-LDT Command-Line Administration: voradmin command](#)).

- *File_1.dat* is rekeyed/encrypted to completion. The applied and new key versions are at version 9 of the key.

```
voradmin ldt attr get /oxf-fs1/gp1/data_files /file_1.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=9
```

- *File_2.dat* is partially rekeyed/encrypted. The applied key version is at version 8 and the transformation key is at version 9.

```
voradmin ldt attr get /oxf-fs1/gp1/data_files/file_2.dat
LDT attributes: rekeyed_size=1440743424,
rekey_status=rekeying, state_saved
Current Key:      name=LDT_AES256_KEY, version=8
New Key:          name=LDT_AES256_KEY, version=9
```

- *File_3.dat* has not been rekeyed/encrypted. The applied key and transformation key are both at version 8, which is the version before the current key rotation.

```
voradmin ldt attr get /oxf-fs1/gp1/data_files/file_3.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=8
```

While CTE-LDT is suspended on the GuardPoint, the backup process starts and archives these three files, including extended attributes, in the backup media.

```
my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_1.dat \
    /backup-media/oxf-fs1/gp1/data_files/file_1.dat
my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_2.dat \
    /backup-media/oxf-fs1/gp1/data_files/file_2.dat
my_backup --preserve=xattr /oxf-fs1/gp1/data_files/file_3.dat \
    /backup-media/oxf-fs1/gp1/data_files/file_3.dat
```

Note

- On Linux platforms, you must perform backups on a GuardPoint during periods when CTE-LDT is not actively transforming data in the GuardPoint. You must schedule backups in conjunction with CTE-LDT schedules or during periods when CTE-LDT is paused. Make sure to coordinate with the backup administrator and schedule accordingly.
- If the preferred method of backup is to create device level snapshots of the file system on the managed host or in the storage array subsystem, you must ensure that the schedule for creating snapshots pauses CTE-LDT before a snapshot is created.

Backup/Restore of Metadata Store File in GuardPoints Undergoing Rekey

- [Restoring a GuardPoint from a Backup](#)
- [Potential Warnings During Restore Operation](#)

Backing up an entire GuardPoint using commands such as `cp`, `tar`, or `rsync`, or even commercial backup products that sweep the file system namespace for backing up files

is supported. This method of backup performed while LDT is in progress over GuardPoints poses some challenges when files in GuardPoints are backed up in an encrypted format. The challenge with this method of backup is the existence of the CTE-LDT Metadata Store file (MDS) in the GuardPoint during the backup or restore process, and strict protection enforced on MDS files preventing deletion or modification to MDS files.

Secondly, your backup utility must backup and restore extended attributes of files and the GuardPoint directory. Because CTE-LDT extended attributes are also protected similarly to MDS files, your backup must be enabled to restore extended attributes for files. To overcome both restrictions, your policy on the GuardPoint must include a security rule without Apply Key Effect to enable your backup utility to replace MDS files, and CTE-LDT extended attributes for files, as part of backup/restore operations. Additionally, your backup utility must be signed with a signature set from CipherTrust Manager for added security, and be executed with the option to preserve extended attribute options available with the standard Linux utilities.

The following table lists the supported backup utilities, required options to preserve extended attributes, and supported versions of the utilities.

Command	Required options	Supported version
<code>cp</code>	<code>--preserve=all</code>	OS default
<code>tar</code>	<code>--xattrs</code>	OS default
<code>rsync</code>	<code>-vapIXWP --inplace</code>	OS default
<code>netbackup</code>	overwrite existing files	v7.6.1 and v7.7.3

You can also backup and restore GuardPoint data, including the MDS file, if the requirements listed above on the backup utility are satisfied.

Note

You must suspend LDT on the GuardPoint before performing a backup or restore.

You can check your GuardPoint to determine if you can use this method of backup. If the directory of your GuardPoint is a mount point, MDS files reside inside your GuardPoint and this method of backup can be used for backing up or restoring your GuardPoint. However, GuardPoint directories below file system mount points must be checked and verified to use this method of backup. To verify, you can run the `voradmin` command to determine the MDS file associated your GuardPoint.

Example:

```
voradmin ldt list all
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp1/___vorm_mds___

Guard Table: version 1 nentries 1
Guard 0:  type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED,
gp=/oxf-fs1/gp1
File List: count 4308
```

You cannot restore to a GuardPoint during LDT, even after suspending LDT. Restoring to a GuardPoint in rekeying state will result in loss of data and potentially, data corruption. Follow the steps below to perform a full level restore of the GuardPoint.

Restoring a GuardPoint from a Backup

To properly restore GuardPoints along with the MDS file:

1. With the same tool that was used for backup, restore to the GuardPoint.
2. Once restore is complete, the GuardPoint needs to be disabled and enabled again. For manual GuardPoint, run the `secfsd` command as shown below. For auto-guards, disable and enable GuardPoints on CipherTrust Manager.

```
# secfsd -unguard GuardPoint
# secfsd -guard GuardPoint
```

3. Disabling GuardPoint will remove the MDS file restored to the GuardPoint and flag the GuardPoint for RELAUNCH to restart LDT. Although the restored MDS file includes critical metadata for execution of LDT, the restored file does not hold data for any files in partial rekeyed status at the time the MDS file is restored. As such, the restored MDS file is automatically removed when GuardPoint is disabled. Once the GuardPoint is enabled, LDT is automatically relaunched on the GuardPoint as the result of RELAUNCH flag. Relaunching LDT will encrypt every file to the latest key version under LDT policy.

Potential Warnings During Restore Operation

When using `cp` for backup/restore, the `cp` command may report a failed attempt to preserve permissions on the Metadata Store File (MDS) during a restore. If you encounter the below message, continue to proceed with the restore steps since this will not affect the MDS file or dataset that is being restored.

```
cp: preserving permissions for '/oxf-fs1/gp1/__vorm_mds__ ':  
Permission denied
```

When using `rsync` for backup/restore, the `rsync` command may report a failed attempt to set extended attribute when restoring the MDS file on system with selinux configured in enforced mode. If user encounters the below message, continue to proceed with the restore steps since this will not affect the MDS file or dataset that is being restored.

```
rsync: copy_xattrs: lsetxattr("/oxf-fs1/gp1/__vorm_mds__  
", "security.selinux") failed: Permission denied (13)  
rsync: rsync_xal_set: lsetxattr("/oxf-fs1/gp1/__vorm_mds__  
", "security.selinux") failed: Permission denied (13)
```

Restore an Encrypted Backup

[Restore a File Fully Rekeyed to the Latest Key Version](#)

[Restore a Partially Rekeyed/encrypted File](#)

[Restore a File Not Rekeyed/encrypted with an Older Key Version](#)

This section illustrates restoration of the three files from the backup media to the same GuardPoint, `/oxf-fs1/gp1`. The files are restored to a different directory under the GuardPoint.

Restore a File Fully Rekeyed to the Latest Key Version

Recall that `file_1.dat` was archived in the backup media when it was fully rekeyed to version 9 of the key. As the current version of the key is also 9, `file_1.dat` is restored from backup without any changes. After restoring the file, the state of the restored file

and its applied and current key versions remain unchanged, as compared to the original file that was backed up.

```
my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files \
/file_1.dat /oxf-fs1/gp1/restored_files/file_1.dat

voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_1.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=9

voradmin ldt attr get /oxf-fs1/gp1/data_files/file_1.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=9
```

Restore a Partially Rekeyed/encrypted File

Recall that *file_2.dat*, archived in the backup media, was partially rekeyed between versions 8 and 9. As the current version of the key was 9 at the time of backup, *file_2.dat* is restored to the GuardPoint with the same version of the key from the time of backup. The file is flagged for *lazy rekey*, meaning that a background rekey operation is scheduled to transform the file to the latest key version the next time an application tries to access the file.

At the completion of restoration, the file is fully transformed to the key version (v9). The key version is also the latest one in the policy. Although the file is flagged for lazy rekey (LAZY_RK), the file does not need to be transformed to the latest key version because it's already there. Had the file been partially rekeyed from version 7 to version 8 of the key at the time of backup, the restored file would have completed rekeying to version 8 at the end of the restoration. Therefore, the LAZY_RK flag would initiate a background transformation to update the key version to the latest key version when the file is accessed.

If this file is not accessed by any application, the file remains unchanged in the GuardPoint. It is not transformed to the latest key version. To trigger a rekey, either re-push the CTE-LDT policy from the CipherTrust Manager, or access the file with an application, such as a file explorer.

```
my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files/ \
file_1.dat /oxf-fs1/gp1/restored_files/file_2.dat
```

```
voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_2.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=lazy_rekey
Key:      name=LDT_AES256_KEY, version=9
```

```
voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_2.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=9
```

Restore a File Not Rekeyed/ encrypted with an Older Key Version

Recall *file_3.dat* was archived in the backup media when it was keyed to version 8 of the key, one version below the latest version at the time of backup. At completion of the restoration, *file_3.dat* is restored from backup to the same version, version 8, that it was keyed to when it was backed up.

However, the file is flagged for lazy rekey. After restoring the file, it is keyed to version 8 and flagged for lazy rekey (LAZY_RK). The file is rekeyed to the latest key version, version 9, as soon as an application accesses the file. If this file is not accessed by any application, the file remains unchanged in the GuardPoint.

```
my_backup --preserve=xattr /backup-media/oxf-fs1/gp1/data_files \
/file_3.dat /oxf-fs1/gp1/restored_files/file_3.dat
```

```
voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_3.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=lazy_rekey
Key:      name=LDT_AES256_KEY, version=8
```

```
sum /oxf-fs1/gp1/restored_files/file_3.dat
39994 1406976
```

```
voradmin ldt attr get /oxf-fs1/gp1/restored_files/file_3.dat
LDT attributes: rekeyed_size=1440743424, rekey_status=none
Key:      name=LDT_AES256_KEY, version=9
```


Restoring Non-CTE-LDT Backup Data to an CTE-LDT GuardPoint

This section describes how to restore data encrypted with a non-versioned key to an CTE-LDT GuardPoint.

If the backup was performed with the Apply Key effect, the backup files are in clear text. Simply restore the clear text files to the CTE-LDT GuardPoint with the Apply Key effect. All files will be encrypted with the versioned key.

If the backup of the non-CTE-LDT GuardPoint was performed without the Apply Key effect, the backup is encrypted, and you must do the following:

Note

The following example is for a manual guarding. The steps may differ slightly if your GuardPoint is configured for auto guard.

1. Create a temporary directory for restoring the files, type:

```
mkdir -p /oxf-fs1/tmp_restore
```

2. Restore the encrypted backup files into the temporary directory, type:

```
cp -pr /backup-media/oxf-fs1/gp1/data_files/* /oxf-fs1/tmp_restore
```

3. Create a Standard Policy with the Apply Key effect for all operations, using the same key as the policy applied on the GuardPoint at the time of backup.
4. Create and enable a new GuardPoint for the temporary directory using the Standard Policy just created.

```
secfsd -guard /oxf-fs1/tmp_restore
```

5. Ensure that the temporary GuardPoint and CTE-LDT GuardPoint are both enabled.

```
secfsd -status guard
GuardPoint          Policy      Type      ConfigState
Status      Reason
-----
-----
/oxf-fs1/gp1      LDT_AES256  manual    guarded
guarded      N/A
/oxf-fs1/tmp_restore  AES256      manual    guarded
guarded      N/A
```

6. Move the restored files from the temporary folder to the GuardPoint enabled with the CTE-LDT policy. The CTE agent encrypts the files in the CTE-LDT GuardPoint using the current key version in effect for the CTE-LDT policy.

```
mv /oxf-fs1/tmp_restore/* /oxf-fs1/gp1
```

7. Disable the temporary GuardPoint and remove the temporary restore directory.

```
secfsd -unguard /oxf-fs1/tmp_restore
rm -fr /oxf-fs1/tmp_restore
```

8. Delete the temporary GuardPoint on the CipherTrust Manager.

Using fsfreeze (Linux only)

If you use the `fsfreeze` command to quiesce access to the file system before creating a snapshot, refer to the `fsfreeze` section in CTE Admin Guide in the Linux Utilities chapter on how to run the `fsfreeze` command to quiesce access to both the file system and the GuardPoint(s).

Warning

Do not use CTE-LDT schedules and do not pause CTE-LDT to align backup schedules with CTE-LDT. Instead, use the `fsfreeze` command.

- Running `fsfreeze -f` on the GuardPoint directory pauses CTE-LDT operations in-progress and freezes access to both the GuardPoint and the underlying file system.
- Running `freeze -u` reverses `freeze -f`, allowing access to underlying file system and resuming CTE-LDT operations.

CTE-LDT Backups Using a File System or Storage-Level Snapshot Tool

You can make a file system snapshot using a Logical Volume Manager service or mirroring/splitting storage level LUNs of a file system inside the storage subsystem. CTE-LDT does not have requirements for where and how you create a file system snapshot. However, it is required that you **suspend CTE-LDT processes before you take the file system snapshot**. Suspending CTE-LDT ensures data and metadata consistency between files and CTE-LDT extended attributes.

You may choose to suspend CTE-LDT manually on the managed host using `voradmin ldt suspend` command or `fsfreeze -f`, or suspend CTE-LDT on the CipherTrust Manager.

Note

Be aware that suspending CTE-LDT on the CipherTrust Manager suspends CTE-LDT on the entire host.

After creating a file system snapshot, you can resume CTE-LDT processes on the GuardPoint using `voradmin ldt resume`, or `fsfreeze -u`, or resuming CTE-LDT on the CipherTrust Manager. Do not mix the use of `fsfreeze` and `voradmin ldt suspend` to pause and resume CTE-LDT. CTE suspends or resumes CTE-LDT processes during

live transformation when freezing or unfreezing GuardPoint access using `fsfreeze -f` or `-u` option.

Note

You can make sure CTE-LDT is suspended at backup time by setting the QoS schedule

You can mount a file system snapshot for data recovery. Configuration for GuardPoints must be duplicated over the mount point of the snapshot file system. Make sure to use the same CTE-LDT policy. Enabling GuardPoints over or under the snapshot mount point provides access to the protected files for recovery. You can choose to manually resume key rotation on the GuardPoints of the snapshot file system, although this is not necessary.

Following is an example of the `fsfreeze` command used to freeze access to the file system `/oxf-fs1` in order to create a snapshot of the file system device. This examples illustrates three GuardPoints enabled inside the file system namespace, `/oxf-fs1/gp-1`, `/oxf-fs1/gp-2`, and `/oxf-fs1/gp-3`. Executing the command `fsfreeze -f` targets any of the GuardPoints in the `/oxf-fs1` mount point and suspends CTE-LDT processes on all of the GuardPoints. Then it freezes access to the file system.

```
fsfreeze -f /oxf-fs1/gp-1
voradmin ldt list all
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp-3/__vorm_mds__
      Guard Table: version 1 nentries 1
      Guard 0:  type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED, gp=/oxf-fs1/gp-3
      File List: count 4308

MDS_2:  type=file, nguards=1, name=/oxf-fs1/gp-2/__vorm_mds__
      Guard Table: version 1 nentries 1
      Guard 0:  type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/oxf-fs1/gp-2
      File List: count 4308

MDS_3:  type=file, nguards=1, name=/oxf-fs1/gp-1/__vorm_mds__
      Guard Table: version 1 nentries 1
      Guard 0:  type=GP, state=REKEYING SUSPENDED (vadm), flags=GP LOCKED
```

```
CKED, gp=/oxf-fs1/gp-1
```

```
File List: count 4308
```

After the file system snapshot is created, executing the `fsfreeze -u` command on any of the GuardPoints in the file system namespace unfreezes access to the file system and resumes CTE-LDT processes on all of the GuardPoints.

```
fsfreeze -u /oxf-fs1/gp-1
voradmin ldt list all
MDS_1:  type=file, nguards=1, name=/oxf-fs1/gp-3/__vorm_mds__
Guard Table: version 1 nentries 1
Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/o
xf-fs1/gp-3
File List: count 4308
MDS_2:  type=file, nguards=1, name=/oxf-fs1/gp-2/__vorm_mds__
Guard Table: version 1 nentries 1
Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/o
xf-fs1/gp-2
File List: count 4308
MDS_3:  type=file, nguards=1, name=/oxf-fs1/gp-1/__vorm_mds__
Guard Table: version 1 nentries 1
Guard 0: type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/o
xf-fs1/gp-1
File List: count 4308
```

Support for Volume Level Snapshots on NAS Shares with LDT on Linux

CTE supports volume level snapshot capabilities of NAS shares. You may choose to enable the snapshot service on an LDT-protected GuardPoint as long as the GuardPoint is enabled before the snapshot service is activated. Note that the snapshot service is deactivated before the GuardPoint is disabled. Keeping the snapshot service active when the GuardPoint is not enabled may result in data corruption or inconsistent LDT metadata at file or GuardPoint levels.

Windows Backup and Snapshots

On Windows, most backup applications use Volume Snapshot Service (VSS) for the backup. Using VSS is required for backing up files in CTE-LDT GuardPoints. VSS service provides a consistent view of the data to backup and restore applications by taking a snapshot of the volume. Windows Backup uses the snapshot volume, while other applications can continue using the original volume for normal I/O operations. VSS snapshot volume uses a “Copy on write” mechanism to provide a consistent view of the data. Some of the high-level steps of the Windows backup process are:

1. Backup application takes a snapshot of the volume using the VSS service.
2. Backup application mounts this VSS volume to read the data to be backed up.
3. The CTE Agent uses the same policy as that of the original volume to protect these snapshots.
4. The CTE Agent applies the policy and rules to all the I/O requests coming from the backup and restore applications.

CTE-LDT Backup and Restore Troubleshooting

Restored files to a GuardPoint protected with conflicting key rules

When restoring an encrypted file from backup media to an CTE-LDT protected GuardPoint without the Apply Key effect, and the file in the backup media does not have an CTE-LDT extended attribute, the file restored to the GuardPoint is set with an CTE-LDT extended attribute that specifies the current key version of the key in the policy associated with the data in the restored file. As the key and key version in the policy do not match the key that was applied to the data at the time of backup, the file restored to the CTE-LDT protected GuardPoint is unreadable.

When restoring an encrypted file from backup media to an CTE-LDT protected GuardPoint without Apply Key effect, and the key specified in the CTE-LDT extended attribute of the file in backup media conflicts with the key rules of the policy on the GuardPoint, the restore operation fails and flags the restored file in error. You can only

remove the file, or truncate it, to clear the error status on the file. Access to such files, except remove or truncate, fail with an EINVAL error.

CTE-LDT Command-Line Administration: voradmin command

Use the `voradmin` utility to gather statistics and administer CTE-LDT on a host/client . It has slightly different syntax and command capabilities depending on whether the host/client is running Linux or Windows.

To use `voradmin`:

1. Log in to the host running the CTE Agent with CTE-LDT enabled.
2. At the command line, type `voradmin`.
3. Follow the usage outputs onscreen to find the available commands and their syntax. The general syntax is:

```
voradmin ldt <command> [args]
```

Command	OS	Description
<code>voradmin ldt attr get delete <guard path> <object path></code>	W/ L	Get or delete attributes for a GuardPoint or Object path.
<code>voradmin ldt exlist <add remove get> <guard path></code>	W	GuardPoint path to add in exclusion list.
<code>voradmin ldt group <check info remove repair comm_info> [hostname] <guard path></code>	W/ L	Various info about the group.
<code>voradmin ldt ior <threshold></code>	W/ L	QoS Rekey rate threshold value in MB/sec.
<code>voradmin ldt key <report map <key> <guard path></code>	W/ L	Creates a report, or map, containing statistics after each key rotation on a GuardPoint.
<code>voradmin ldt list all</code>	W/ L	List all MDS files and GuardPoints undergoing CTE-LDT processing.
<code>voradmin ldt monitor [interval]</code>	W	

Command	OS	Description
		Monitor CTE-LDT progress. If an interval is specified (in units of seconds), it continually updates the monitoring output at the specified interval.
<code>voradmin ldt rekey report <guard path> [<output file>]</code>	W/ L	Generate a rekey report manually for a Guardpath.
<code>voradmin ldt report <guard path> <log file></code>	W	Full GuardPoint or log file path.
<code>voradmin ldt rescan <guard path></code>	W	Rescan a GuardPoint for CIP remediation.
<code>voradmin ldt resume <guard path> all</code>	W/ L	Resume guarding the GuardPoint directory.
<code>voradmin ldt rmltd <guard path></code> <code>voradmin ldt rmstore <guard path></code>	L	Both the <code>rmstore</code> and <code>rmltd</code> options remove the LDT metadata information remaining in a GuardPoint directory after permanent unguarding of the specified GuardPoint. The <code>rmstore</code> option removes the MDS file, if it exists. The <code>rmltd</code> removes the MDS file and the files in LDT Private Directory for GuardPoints over NFS shares. Note: GuardPoints over NFS shares are only supported if you are using the CM as your key manager.
<code>voradmin ldt skip <show> <guard path> <filename></code>	W	Display the files skipped during data transformation.
<code>voradmin ldt space <guard path></code>	W/ L	Estimates disk space needed for metadata and MDS store information.
<code>voradmin ldt summary <guard path></code>	W/ L	Reports the exact number of files encrypted in the GuardPoint in real time.
<code>voradmin ldt suspend <all><guard path></code>	W/ L	Suspend the rekey on a specified GuardPoint, or all the GuardPoints, on this host.
<code>voradmin ldt stats [<guard path>] [<interval>]</code>	W/ L	Obtains transformation CTE-LDT statistics such as: • Current rekey status

Command	OS Description
	<ul style="list-style-type: none"> • Start time • Estimated completion time • Percentage completed • Total data • Amount of data transformed • Total files • Number of files transformed • Number of files skipped • Number of files remaining for rekey

Migrating a GuardPoint to a Different CTE-LDT Policy

To change the CTE-LDT policy that an CTE-LDT GuardPoint uses, complete the following steps to ensure that the GuardPoint is migrated properly from one CTE-LDT policy to another CTE-LDT policy.

Note

This procedure is for the migration of local file system GuardPoints only. For information about migrating NFS GuardPoints to `clear_key`, see [Migrating GuardPoints over NFS from or to an LDT Policy](#).

Scenario

The GuardPoint is currently attached to LDT-Policy-1, which rekeys from `clear_key` to `LDT-Key-1`. The objective is to migrate the data in the GuardPoint to another versioned key, `LDT-Key-2`. Migration to `LDT-Key-2` requires detaching the GuardPoint from `LDT-Policy-1`, and then attaching it to `LDT-Policy-2`, assuming `LDT-Key-2` is the versioned key specified in `LDT-Policy-2`. To do so:

1. Clone the latest version of the key `LDT-Key-1` to a non-versioned key such as `LDT-Key-1-Clone`.
 - a. From the Products page in the CipherTrust Manager Console, click **Keys** in the left hand pane.

Tip

To navigate to the Products page from anywhere in the CipherTrust Manager Console, click the App Switcher icon in the top left corner.

- b. Click the name of the versioned key that you want to clone.
 - c. In the Key Details area, find the version that you want to clone.
 - d. Click the (...) button at the end of the row and select **Clone** to clone the selected version of the key.
2. Identify or create a new versioned key that you want CTE-LDT to use to re-encrypt the data. For example, `LDT-Key-2`.
3. Create a new Live Data Transformation policy that specifies `LDT-Key-1-Clone` as the Initial Key and `LDT-Key-2` as the Transformation Key.
4. Make sure that all data transformation has completed on the GuardPoint. To verify this, use the `voradmin ldt attr get <GuardPoint>` command.
5. In your key manager, unguard the GuardPoint.
6. On the host, remove the existing CTE-LDT attributes on the GuardPoint using the `voradmin ldt attr delete <GuardPoint>` command.

```
voradmin ldt attr delete /oxf-fs1/gp1
LDT metadata has been removed from all files in GuardPoint [/oxf-fs1/gp1]
LDT attributes deleted from 10 files in /oxf-fs1/gp1
LDT: Metadata has been removed from all files in guard point [/oxf-fs1/gp1]
```

7. Guard the directory using the new LDT policy.
 - If you have selected Auto Guard, data transformation begins as soon as the host gets the new policy information from the key manager.
 - If you have selected Manual Guard, use the `secfsd -guard <GuardPoint>` command on the host to begin data transformation.

Removing CTE-LDT and Security Encryption

If you want to stop using CTE-LDT on a GuardPoint or on a whole host/client , follow the instructions in the following sections.

- [Migrating a GuardPoint Out of CTE-LDT](#)
- [Deleting CTE-LDT Metadata \(Linux\)](#)
- [Deleting CTE-LDT Metadata \(Windows\)](#)
- [Removing CTE-LDT from a Host](#)

Migrating a GuardPoint Out of CTE-LDT

Migrating a GuardPoint from CTE-LDT removes the security encryption. It also provides an Administrator with the flexibility to relax the compliance requirement, when strict compliance for frequent key rotation on specific data is no longer mandatory. The following sections describe how to migrate a GuardPoint from CTE-LDT to a non-CTE-LDT policy, or to remove encryption protection from it.

Converting a GuardPoint from an CTE-LDT Policy to a non-CTE-LDT Policy

If you want to do more than just change the policy on a GuardPoint from an CTE-LDT policy to a non-CTE-LDT policy, see [Deleting CTE-LDT Metadata \(Linux\)](#) or [Deleting CTE-LDT Metadata \(Windows\)](#).

Note

Converting GuardPoints from Live Data Transformation policies to Standard CTE policies is not supported. Similarly, CTE-LDT protected GuardPoints cannot be migrated to clear_key.

1. Clone the versioned key associated with the CTE-LDT GuardPoint to a non-versioned key.

The clone function creates a new key with the same cryptographic encryption material as the current version of the cloned versioned key.

This allows CTE-LDT to use the cloned key in a non-CTE-LDT policy to convert the GuardPoint from an CTE-LDT to a non-CTE-LDT managed policy.

- a. In the CipherTrust Manager Applications Page, open the **Keys & Access Management** application.
 - b. Click the name of the versioned key that you want to clone.
 - c. In the Key Details area, click the (...) button at the end of the row showing the current version of the key and select **Clone** to clone the current version.
 - d. Enter a new name for the key in the **Key Name** field. Do not select the **CTE Versioned** option for the clone.
 - e. Click **Clone**.
2. Open the **CTE** application and click **Clients** in the left-hand menu bar.
 3. Click on the name of the client whose GuardPoint you want to migrate.
 4. Find the GuardPoint you want to migrate in the GuardPoints table, then click the (...) button at the end of the row and select **Disable**.
 5. After CTE-LDT disables the GuardPoint, click the (...) button at the end of the row and select **Remove**.
 6. (Linux only) Ensure that the GuardPoint is removed on the managed host:

```
secfsd -status guard
No GuardPoints configured
```

7. CTE-LDT creates extended attributes for every file under the GuardPoint as well as the GuardPoint directory. Now that the CTE-LDT policy does not manage the GuardPoint, you must remove the extended attributes for every file in the GuardPoint, type:

```
voradmin ldt attr delete /<GuardPoint>
```

8. The command may take some time depending on the number of files in the GuardPoint.

Note

For all of the file system mount points that contain an CTE-LDT protected GuardPoint, you must clean up the metadata first. See [Deleting CTE-LDT Metadata \(Linux\)](#).

9. Create a non-CTE-LDT policy using the cloned key you created earlier in this procedure.
 - a. In the **CTE** application, click **Policies** in the left-hand menu bar and then click **Create Policy**.
 - b. Enter a name for the policy in the **Name** field.
 - c. In **Policy Type**, choose **Standard**.
 - d. Click **Next**.
 - e. On the Security Rules page, click **Next** to skip adding a security rule.
 - f. On the Key Selection Rules, click **Create Key Rule**.
 - g. In the **Key Name** field, select the cloned key you created earlier.
 - h. Click **Add**.
 - i. Click **Next**
 - j. Confirm that the key rule is correct and click **Save**.
10. Apply the non-CTE-LDT policy to the GuardPoint.

Caution

Make sure that you have removed all of the CTE-LDT metadata from the GuardPoint before applying the non-CTE-LDT policy.

- a. Open the **CTE** application and click **Clients** in the left-hand menu bar.
- b. Click on the name of the client whose GuardPoint you want to migrate.
- c. Click **Create GuardPoint**.
- d. In the *Create GuardPoint* dialog box, select the Policy that you just created in the **Policy** field.
- e. Select the **Type: Auto Directory** or **Manual Directory**.
- f. In the **Path** field, and enter or browse to the directory to protect.
- g. When you are done, click **Create**.

Remove Protection from a GuardPoint

When compliance may no longer require protecting data in a GuardPoint, you may choose to unprotect/decrypt it. Before removing protection from your GuardPoint, you must decrypt the data in your GuardPoint by setting it to clear. You have two options to decrypt your data:

- While a GuardPoint is protected and enabled under an CTE-LDT policy, you can use copy or backup/restore commands to save files in your GuardPoint to a location outside of your GuardPoint.
- Use the `dataxform` command to transform your GuardPoint to clear in an offline transformation process.

For GuardPoints over NFS, you must backup the entire GuardPoint before you unguard the GuardPoint. Then you can restore the files from backup over the GuardPoint directory after you remove the protection on the GuardPoint.

Copying Files to Decrypt Them

If you choose to copy your files, you must create a directory outside of the GuardPoint and then copy the files into the GuardPoint directory. After finishing copying, complete the following steps:

1. Open the **CTE** application and click **Clients** in the left-hand menu bar.
2. Click on the name of the client whose GuardPoint you want to remove.
3. Find the GuardPoint you want to remove in the GuardPoints table, then click the (...) button at the end of the row and select **Disable**.
4. After CTE-LDT disables the GuardPoint, click the (...) button at the end of the row and select **Remove**.
5. (Linux only) Ensure that the GuardPoint is removed on the managed host:

```
secfsd -status guard
No GuardPoints configured
```

This completes removal of the GuardPoint under an CTE-LDT policy. You can now remove the original files and data within the GuardPoint namespace.

Using Dataxform Command to Transform the Files

If you choose to use the dataxform command to transform data in your GuardPoint to clear, use the `voradmin` command to verify that earlier versions of the versioned key are not in use on your GuardPoint. Complete the following steps to clear all metadata in your GuardPoint. Then, transform your GuardPoint to clear.

1. Clone the versioned key associated with the CTE-LDT GuardPoint to a non-versioned key.
 - a. In the CipherTrust Manager Applications Page, open the **Keys & Access Management** application.
 - b. Click the name of the versioned key that you want to clone.
 - c. In the Key Details area, click the (...) button at the end of the row showing the current version of the key and select **Clone** to clone the current version.

directory. Now that the CTE-LDT policy does not manage the GuardPoint, you can remove the extended attributes for every file in the GuardPoint.

7. Remove the extended attributes of files in a GuardPoint, type:

```
voradmin ldt attr delete <GuardPoint>
```

The command may take some time depending on the number of files in the GuardPoint. After metadata deletion is complete, you can apply the offline transformation policy on the GuardPoint.

8. CipherTrust Manager, guard and enable the GuardPoint with the Standard policy you created earlier.
9. After enabling your GuardPoint, run the dataxform command on the managed host to transform the GuardPoint to a `clear_key`, type:

```
dataxform --rekey --gp /<GuardPoint>/ --preserve_modified_time --  
preserve_access_time --cleanup_on_success
```

10. After completion of dataxform, unguard the GuardPoint.
11. Remove the GuardPoint from the dataxform policy in CipherTrust Manager.

Deleting CTE-LDT Metadata (Linux)

To remove the metadata associated with the GuardPoint, you must run the voradmin command on the managed host to remove the CTE-LDT metadata associated with the GuardPoint.

Warning

Before you attempt to remove an MDS file, make sure that no CTE-LDT-protected GuardPoints remain configured under the file system mount point.

In the following example:

- `/oxf-fs1` is the mount point

- `/oxf-fs1/gp1` is the CTE-LDT-protected GuardPoint

To delete the metadata associated with the GuardPoint:

1. Ensure that the GuardPoint is not enabled on the host. Run the command below and verify that the GuardPoint pathname does not appear in the output of the `secfsd` command, type:

```
secfsd -status guard
```

2. Run the `voradmin` command to remove CTE-LDT attributes on the GuardPoint, type:

```
voradmin ldt attr delete <GuardPoint path>
```

For example:

```
voradmin ldt attr delete /oxf-fs1/gp1
LDT metadata has been removed from all files in GuardPoint /oxf-
fs1/gp1
```

Deleting the LDT Private Space Directory for NFS Shares

As described in [LDT Metadata Management Over NFS/CIFS Shares](#), CTE-LDT creates and manages LDT NFS metadata in the LDT Private Space Directory inside the GuardPoint. To remove the LDT Private Space Directory and the files it contains, use the `voradmin ldt rmltd <GuardPoint>` command. For example:

```
voradmin ldt rmltd /nfs-oxf-fs1/gp1
Enter YES if GuardPoint /nfs-oxf-fs1/gp1 is no longer associated with a
n LDT policy -> YES
LDT metadata on /nfs-oxf-fs1/gp1 has been removed.
```

Deleting CTE-LDT Metadata (Windows)

Use the following command to delete the CTE extended attribute for a file or a GuardPoint. This is useful when removing a GuardPoint from under an CTE-LDT policy (see [Migrating a GuardPoint Out of CTE-LDT](#)).

```
voradmin ldt attr delete [<file name path> | <guard path>]
```

For an overview of `voradmin ldt`, see [CTE-LDT Command-Line Administration: voradmin command](#).

Removing CTE-LDT from a Host

Once you have registered a host and enabled CTE-LDT, you cannot disable the CTE-LDT feature by unchecking the CTE-LDT box. You must unregister the host from the CipherTrust Manager, then register it again without CTE-LDT. When you remove the CTE-LDT feature from a host entirely, the host's CTE-LDT license becomes available for use on another host.

1. Stop all applications from accessing data in CTE-LDT GuardPoints on the host.
2. Migrate data in every CTE-LDT GuardPoint using the steps described in the section [Remove Protection from a GuardPoint](#).

Warning

Potential data loss. Ensure that you have decrypted the data and, optionally, copied it out of the GuardPoint. Once the CTE Agent software is removed, access to data is no longer controlled by CTE. If the data was encrypted, it remains encrypted, and there is no way to read it.

3. Remove the GuardPoints on the host from the CipherTrust Manager.
 - a. Remove the CTE-LDT metadata from those GuardPoints.
 - b. Remove the MDS files associated with those GuardPoints, if necessary. See [Deleting CTE-LDT Metadata \(Linux\)](#) for more information.

4. Remove the host from the CipherTrust Manager. For details, see the *CTE Agent for Linux Advanced Configuration and Integration Guide* or the *CTE Agent for Windows Advanced Configuration and Integration Guide*.
5. Re-install the agent on the host.
6. Register the host with the CipherTrust Manager. This time, do not select the CipherTrust Transparent Encryption - Live Data Transformation option. See [Enabling CTE-LDT on a Protected Host](#).

Uninstalling the Agent while CTE-LDT is Rekeying GuardPoints

You cannot uninstall the CTE Agent if CTE-LDT is rekeying any GuardPoints on that host/client .

If you want to uninstall the CTE Agent from a host/client that contains CTE-LDT GuardPoints, you must first unencrypt the files on each CTE-LDT GuardPoint using the steps noted in the previous sections.

Detecting Loss of NAS connection to an LDT GuardPoint Group

This section describes the LDT over NFS mechanism for detection of loss of a NAS (Network attached storage) connection to an LDT over NFS GuardPoint on all members of the LDT GuardPoint Group. LDT over NFS monitors LDT operations on NAS servers for detection of loss of NFS connection and implements steps to recover from loss of NAS access. This section describes the procedures and considerations for recovering an LDT over NFS GuardPoint from loss of NAS connection on a member of LDT GuardPoint Group.

Brief Overview of LDT Rekey Operations

Rekey operations within LDT GuardPoint Groups are executed on the primary host in coordination with other members of the LDT GuardPoint Group. coordination includes interlocking access to files by applications and rekey operations. It also includes the primary host updating persistent LDT metadata on behalf of the application file or IO operations on any member of the LDT GuardPoint Group. Such active coordination is critical to the correct operations of LDT over NFS. Loss of NAS connection on any of the members in LDT GuardPoint Group disrupts the coordination between primary and other members. LDT manages loss of the NAS access on all members of the LDT GuardPoint Group to ensure data and metadata correctness with minimal impact to production workloads.

This section describes the LDT over NFS mechanism for detection of loss of NAS connection to an LDT over NFS GuardPoint on a member (primary/secondary) host of the group and also describes the procedures and considerations for recovering an LDT over NFS GuardPoint from the loss of NAS connection on a member host of the group.

Effects of loss of NAS connection on a GuardPoint

Loss of NAS connection on any member of an LDT GuardPoint Group, including primary hosts, will have negative effects on active LDT operations on the target NAS, and across the entire LDT GuardPoint Group and applications accessing files inside the affected GuardPoints. Regardless of the LDT status (active or suspended), connection loss of one member ripples through all of the LDT GuardPoint Group members. The following scenarios can occur during loss of NAS access while LDT operations are in active or suspended state:

- Application file operations (read/write, truncate, rename, etc.) on files undergoing rekey can be blocked and consequently affect subsequent LDT operations on the same files. Rekey operations will be delayed, or even not started, as the result of pending file IO operations by applications.
- Pending LDT rekey IO operations will not complete and consequently, block or prevent, subsequent rekey operations from starting as well as block applications attempting to access such files.
- Rotation of a key will block launching rekey on the affected GuardPoints.

LDT over NFS I/O monitoring system to detect loss of NAS connection

To avoid the ripple effect of loss of NAS connection across entire LDT GuardPoint Groups, LDT monitors LDT level operations that access LDT metadata, such as MDS and LDT rekey attributes, and reads/modifies cipher-text data inside files undergoing rekey. LDT monitors progress on LDT operations in-progress. If an operation fails to complete within 30 seconds, LDT declares the pending operation incomplete and takes action to recover from the operation.

LDT over NFS Monitoring Control Thread

LDT starts a special control thread on every CTE host to monitor LDT operations for potential timeouts due to NAS connection loss. The monitoring begins when the first LDT over NFS protected GuardPoint is enabled. This thread monitors LDT operations on all LDT protected GuardPoints on the CTE host.

The LDT over NFS IO monitoring system monitors only LDT level operations. The control thread does not monitor application file operations on the files in the same GuardPoints that LDT over NFS monitoring thread is monitoring. If an application file operation hangs due to loss of NAS connection, the LDT over NFS monitoring control thread will detect it when the LDT operation starts and then timeouts on the same file.

Managing loss of NAS connection to a GuardPoint on the PRIMARY host

LDT takes action to recover from the loss of the NAS connection. The action depends on whether the loss of connection is detected on the primary host or other members of the LDT GuardPoint Group.

Managing Primary host losing NAS connection to a GuardPoint - Primary Demotion

As soon as the monitoring thread detects the loss of the NAS connection to a GuardPoint on the primary host, the control thread initiates Primary Self-Demotion to demote the primary host from the primary role and responsibility for the GuardPoint. In this scenario, LDT on the primary host will start the demotion process for the affected GuardPoint in order to elect another member of LDT GuardPoint Group for the primary role. Election of another member triggers the failover process of LDT operations from the demoted primary host to the newly elected primary host. The election and the

failover process is transparent and enables LDT to continue rekey operations using the newly elected primary host for all healthy members of the LDT GuardPoint Group.

The detection and recovery action resumes LDT operations and is fully transparent. It does not involve administrator intervention except when finishing rekey operations that were in progress on the demoted primary host when the loss of the NAS connection was detected. From the perspective of the newly promoted primary host, those files are in `LDT INCOMPLETE` status and resuming rekey on `LDT INCOMPLETE` files requires administrator intervention.

When the failover from primary role occurs, the newly elected primary host identifies the files undergoing rekey and marks those files as `LDT INCOMPLETE` status. At this stage, the self-demoted primary host remains in the LDT GuardPoint Group as an active member but without access to the NAS server. Due to the loss of the NAS access, LDT on the self-demoted primary host blocks all new user attempts to access files in the GuardPoint. It then continues to participate in the transformation of the remaining files in the GuardPoint with the newly elected primary host.

Rekey Completion on GuardPoint with INCOMPLETE Files

Although the newly-elected primary host resumes rekey and transforms the remaining files in the GuardPoint, it does not resume transformation on `INCOMPLETE` files until the self-demoted primary host has been recovered through the manual administrator intervention. Files in `INCOMPLETE rekey` status are the last files in the GuardPoint after administrator intervention.

Recovering self-demoted primary host

The self-demoted primary host can be recovered manually by shutting down the host with the host membership removed from the LDT GuardPoint Group. A successful shutdown process that halts the host automatically removes the host membership from the LDT GuardPoint Group. Therefore, it is critical for the self-demoted primary host to reach the halt state at the completion of the shutdown. However, the shutdown process may not reach the halt state due to pending IO operations as the result of the loss of NAS connection.

Shutdown failure on self-demoted primary host

If you are using LDT over NFS with GuardPoints across multiple NAS shares, and a primary node loses the connection with a NAS server but does not reach the halt state, then the other GuardPoints from the other NAS shares, that are also guarded on that self-demoted primary node, must be manually removed from the LDT GuardPoint Group for recovery.

When you must manually force the host to halt, this is referred to as an ungraceful failure. Remedy this issue from the newly promoted primary server:

1. Remove the self-demoted primary host from the GuardPoint. Type:

```
# voradmin ldt group remove <hostname_of_self-demoted primary>
<guardpoint_path>
```

Note

- **When you run the `voradmin ldt group remove` command for other GuardPoints, from other NAS servers present on that node, and if the node is the primary for any of those GuardPoints, then primary failover is triggered.**

2. Repeat the command on any other GuardPoints on other NAS servers that are guarded by the self-demoted primary host.
3. Run the recover command for the GuardPoints that were on the node that lost connection to the NAS server. Type:

```
# voradmin ldt demotion recover <guardpoint_pathname>
```

For more information, see the following two sections:

- [Failover for LDT GuardPoint Group](#)
- [Secondary Host Failure](#)

Self-demoted primary host reaches HALT state

If the shutdown process on the self-demoted primary does reach the halt state, then the self-demoted primary host has already been removed from LDT GuardPoint Group. Your next step in the recovery process depends on the status of NAS connection of the host.

If a loss of NAS connection persists, disable the GuardPoints on NAS servers that cannot be accessed on the self-demoted primary host:

- Change the guard status of the GuardPoints configured for Auto Guard to remain disabled on the self-demoted primary host
- Do not enable GuardPoints configured for Manual Guard if the self-demoted primary host has rebooted

You can then proceed with the following voradmin command to resume rekey operations on the files in INCOMPLETE status. This command must be executed on the newly elected primary host.

```
# voradmin ldt demotion recover <guardpoint_pathname>
```

If the loss of NAS connection has been resolved by the time the self-demoted primary host reboots, you can allow CTE start-up services to enable the GuardPoints configured for Auto-Guard on the host, or you can manually enable a GuardPoint configured for Manual Guard. After enabling the GuardPoints, you can perform the following command on the newly-elected primary host to resume rekey on files currently in INCOMPLETE status: use # voradmin ldt demotion recover

Upon completion of the recovery steps, the remaining files in INCOMPLETE status will be rekeyed and the GuardPoint itself will transition to rekeyed state.

Note

Self-demoted primary hosts must be rebooted after the promotion of a member to primary host.

Handling loss of the NAS connection to a GuardPoint on a non-PRIMARY host

In the event of a non-primary host losing NAS connection, the LDT over NFS IO monitoring thread on the non-primary host will detect the loss of the NAS connection and trigger self-isolation from LDT GuardPoint Group.

The outcome of self-isolation is that LDT on the non-primary host blocks all users' access to the affected GuardPoint and begins acknowledging all LDT requests from the primary host. Effectively, it does not perform subsequent operations requested by the

primary host, but it sends positive responses to the primary host so that the primary host can continue rekey operations.

Recovery steps for recovery from a self-isolated secondary host are the same as those for the recovery of a self-demoted primary host.

LDT over NFS demotion-related Alarms to CipherTrust Manager

LDT sends the following alarms to CipherTrust Manager during primary host self-demotion, election of a new primary host after demotion, recovery of the self-demoted primary host, and recovery and transformation of the INCOMPLETE files on the new primary host:

[CGS3347i] LDT over NFS-ALERT: Primary host of GuardPoint [GuardPoint] demoting

LDT on the primary host sends this alarm to notify of the loss of the NAS connection on the primary host, for the specified GuardPoint, and the initiation of the self-demotion process.

[CGS3348i] LDT over NFS-ALERT: Primary host of GuardPoint [GuardPoint] demoted

LDT on the self-demoted primary host sends this alarm for the specified GuardPoint when the self-demotion process completes.

[CGS3349e] LDT over NFS-ALERT: Demotion of Primary host of GuardPoint [GuardPoint] failed, error [ErrorNumber]

LDT on the primary host sends this alarm if the self-demotion attempt fails for the specified GuardPoint. The error code included in the alarm indicates the reason for failure. Notify Thales Support in the event of this alarm.

[CGS3355i] LDT over NFS-ALERT: INCOMPLETE file. After DEMOTION, Admin intervention is required for full recovery, GuardPoint [GuardPoint] objID [InodeNumber] error [ErrorNumber]

LDT on the newly-elected primary host sends this alarm for each file detected in INCOMPLETE status.

[CGS3357i] LDT over NFS-ALERT: Manual recovery invoked to recover the DEMOTED GuardPoint[GuardPoint]

LDT sends this alarm from the newly-elected primary host when the administrator executes the `voradmin` command to begin the recovery process.

[CGS3358e] LDT over NFS-ALERT: DEMOTED GuardPoint recovered on the newly-elected primary host while the old primary host is still alive, objID [InodeNumber]

LDT sends this alarm from the self-demoted primary host before the recovery steps, if the self-demoted host receives a request for an LDT operation on a file in INCOMPLETE status from the newly-promoted host. This alarm indicates that the newly promoted host has begun recovery of files in INCOMPLETE status without first recovering the self-demoted primary host.

[CGS3354e] LDT over NFS-ALERT: Manual recovery failed on DEMOTED GuardPoint [GuardPoint] objID [InodeNumber] error [ErrorNumber]

LDT sends this alarm from the newly elected primary host, during recovery of an INCOMPLETE file, when it encounters any errors during LDT rekey related to the recovery of the file or during the resumption of rekey on the file.

[CGS3359I] LDT-NFS-ALERT: Secondary host DEMOTED due to loss of NAS connection, GuardPoint [GuardPoint]

LDT on a self-demoted secondary host sends this alarm for the specified GuardPoint when the self-demotion process completes on the self-demoted secondary.

Introduction to LDT over CIFS

Live Data Transformation is now extended to network share deployments through a distributed architecture system, which allows users to encrypt data on CIFS/NFS shares without any application downtime. CTE agents can support encryption of data workloads for both direct-attached storage and network shares with zero encryption downtime and automated key rotation capabilities. This allows you to meet compliance requirements, without disrupting business applications.

The administration of LDT protected GuardPoints over NFS/CIFS shares is similar to the administration of GuardPoints in local file systems. The requirements related to policies, versioned keys, security and key rules, QoS, and file level rekey operations,

for GuardPoints over local file systems, also apply to GuardPoints over NFS/CIFS shares.

The distributed aspect of LDT over NFS/CIFS shares extends the LDT administration across multiple hosts sharing access to the same GuardPoints. This section describes the extended administration of LDT across multiple hosts and how you can use LDT for data transformation across those multiple hosts.

LDT over CIFS/NFS High-Level Overview

As mentioned, the LDT operation for transformation of data over CIFS/NFS is almost the same as for GuardPoints in local file systems. LDT uses the same workflow for transforming data in files and GuardPoint management. The difference between local file systems and NFS/CIFS shares is the number of hosts involved in the transformation of GuardPoints. As multiple hosts can access the same GuardPoint over NFS/CIFS, LDT operations for transforming files in shared a GuardPoint must be coordinated across the CTE hosts enabling the GuardPoint. Coordination entails multiple file-level and LDT operations to ensure the data transformed from the previous key to the new key is not accessed by any other application on any hosts on which the GuardPoint is enabled.

Note

LDT support for NFS shares is restricted to Linux, and LDT support for CIFS shares is restricted to Windows. This means that CTE-LDT on Linux does not support LDT protected GuardPoints on CIFS shares, and similarly, CTE-LDT On Windows does not support LDT protected GuardPoints on NFS shares.

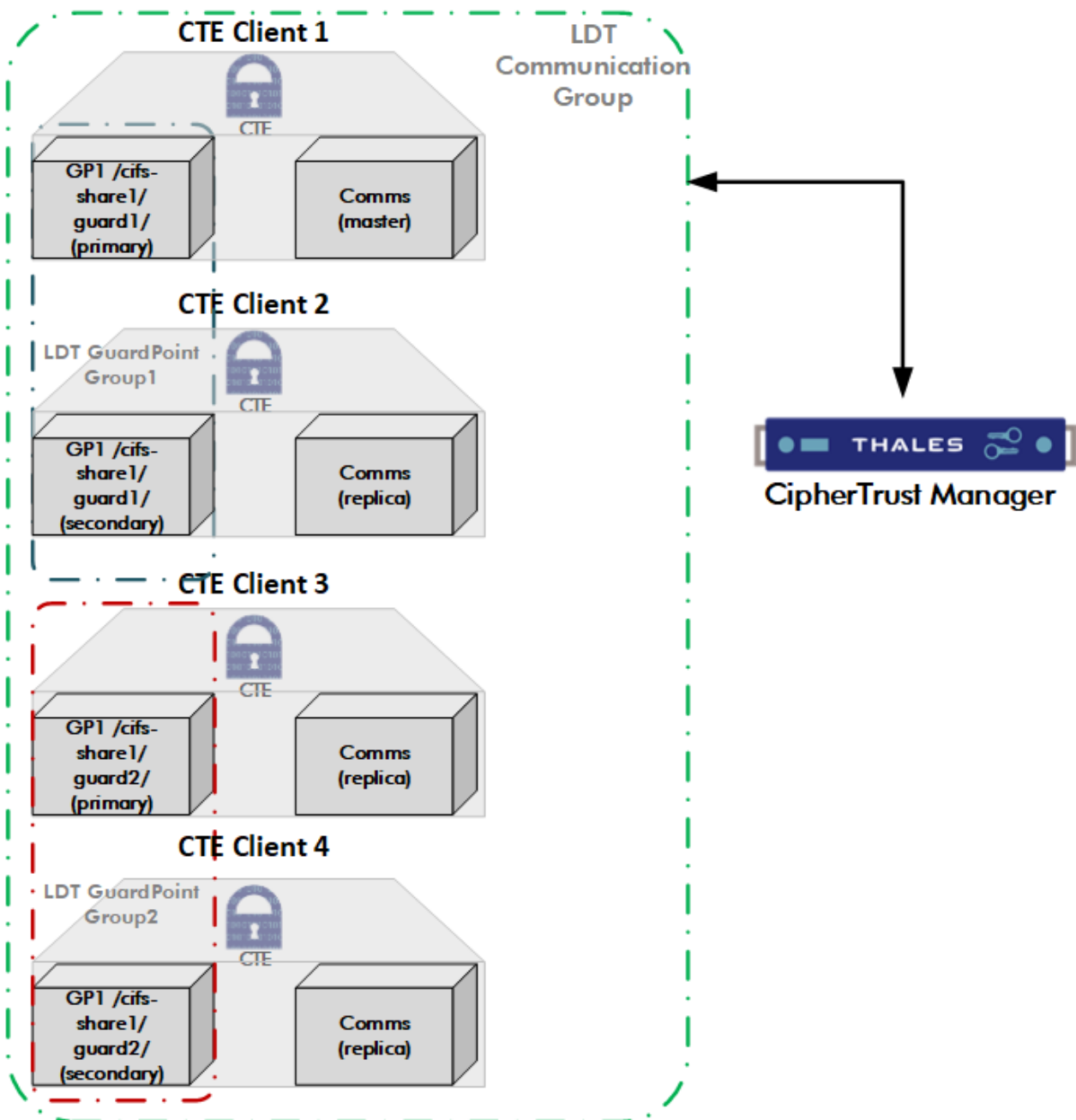
LDT architecture for supporting NFS/CIFS shares is based on a distributed architecture. In this architecture, a single CTE host is delegated the responsibility for transforming the entire dataset in a GuardPoint. Other CTE hosts, sharing access to the same GuardPoint, participate in the data transformation process by the transforming host. The transforming host is referred to as the **LDT GuardPoint Group primary host**, and the participating hosts in the data transformation process are **LDT GuardPoint Group secondary hosts**. In this architecture, an LDT GuardPoint Group is composed of at least three or more CTE hosts (all Linux or Windows) sharing access to the same LDT protected GuardPoint (over NFS or CIFS shares, respectively).

Within an LDT GuardPoint Group, the LDT primary host executes the same operations for data transformation as those executed on GuardPoints in a local file system. The extended architecture of LDT over NFS/CIFS involves the secondary hosts in the group for execution of LDT operations initiated by primary host. For example, when a file is selected and initialized for rekey, the selection and initialization process is not local to the primary host. The primary host performs an extended selection and initialization operation across the members of the group. Once the operation is successfully performed and acknowledged by each member, the primary host proceeds with the next step in the transformation process on the selected file. Below are other extended LDT operations performed by the primary host in conjunction with members of the LDT GuardPoint Group:

1. Locking/unlocking data for rekey.
2. Suspending/resuming LDT.
3. Applying a new key to the GuardPoint.
4. Launching LDT, or concluding LDT, on the GuardPoint associated with the LDT GuardPoint Group.
5. Certain file operations, initiated on a secondary host, that change LDT metadata. For such operations, the secondary host performing the file operation notifies the primary host to update and commit all metadata changes for the operation.

LDT Communication Groups and LDT GuardPoint Groups

The LDT Communication Group provides the services for management and monitoring of the members of LDT GuardPoint Groups, and supports the LDT Messaging Services. An LDT Communication Group contains a collection of LDT-enabled CTE clients that are all on the same network, managed by one CipherTrust Manager and can communicate with each other over that network. An LDT GuardPoint group consists of CTE clients guarding a common NFS/CIFS share. An LDT Communication Group enables the primary host of an LDT GuardPoint Group to communicate with the secondary hosts that are members of that GuardPoint group.



LDT Communication Group

An LDT Communication Group is mandatory when using LDT over NFS/CIFS across all CTE clients managed through CipherTrust Manager. CipherTrust Manager pushes the LDT Communication Group details to all of the CTE clients in a group.

The first CTE client added to the group is initially designated as the master node. The remaining CTE clients are replicas. All active communications go through the master node.

All CTE clients intending to guard NFS/CIFS shares with an LDT policy must be able to communicate with each other. In order for the CTE clients to communicate, they must

be on the same network and in a single LDT Communication Group. The agents can specify the LDT Communication Group to join at the time of registration.

Note

Even if the CTE clients do not guard the same share, they must be in the same LDT Communication Group on CipherTrust Manager.

LDT Communication Group Limitations

1. You must create an LDT Communication Group on your CipherTrust Manager and manually add all of the CTE clients to it.
2. Minimum of three CTE clients required for membership to the LDT Communication Group for proper and continuous operations of LDT services across all LDT GuardPoint Groups.
3. The majority of the CTE clients must be fully operational for proper LDT operations across all LDT GuardPoint Groups.

The LDT Communication Group requires that the majority of CTE clients are active, in order for the LDT Communication Group to work properly. If 50% of the LDT Communication Group servers, or 2 out of 3 **triad** nodes fail, then the LDT Communication Group cannot recover from that situation. To fix this issue, you must either:

- Reboot all of the CTE clients in the entire LDT Communication Group.
 - In Windows, restart `secfsd` through the **Control Panel > Services** page on all of the CTE clients in the entire LDT Communication Group.
4. The name of the LDT Communication Group cannot contain special characters.
 5. If you are repurposing a CTE client that is currently being used for LDT over NFS/CIFS, but still runs CTE and is still registered to the same CM, then you must:
 - a. **Remove all CIFS/NFS LDT GuardPoints from the host prior to removing the host from the LDT Communication Group.**
 - b. **Reboot the host.**
 6. Remove a host from the LDT Communication Group if the host is being decommissioned, meaning that you plan to power it down for the foreseeable future, or while CTE is being uninstalled.

LDT Communication Group Considerations

- The LDT Communication Group must be specified at the time of client registration.

Note

For upgrades, add the agents to the LDT Communication Group using the **Add Client** function on CM. See [Adding Clients to an LDT Communication Group](#) for more information.

- Do not shut down the entire LDT Communication Group at once. This is so that maintenance cycles can activate and run. Perform maintenance in a round robin fashion.
- Remove a host from the LDT Communication Group if the host is being decommissioned, meaning that you plan to power it down for the foreseeable future, or while CTE is being uninstalled.
- The CTE host that is the LDT Communication Group master is critical for proper operations of LDT across all LDT GuardPoint Groups. Thales recommends that you dedicate a separate host as your LDT Communication Group master, although it is not mandatory. This isolates the host from unexpected system failures, hence preventing failover of the LDT Communication service to another CTE client critical for your production workload.
- Thales recommends using a minimum of three nodes for optimal performance.

To create and manage LDT Communication Group in the UI, see: [Managing LDT Communication Groups](#).

Multiple LDT Communication Groups Considerations

CipherTrust Transparent Encryption now supports multiple LDT Communication Groups. Be careful to not misconfigure the multiple LDT Communication Groups:

- You cannot use identical NFS/CIFS share paths in multiple LDT Communication Groups registered to a **single** CipherTrust Manager. Each share path must be unique.
- You cannot use identical NFS/CIFS share paths in an LDT Communication Group with the same name on **multiple** CipherTrust Managers. LDT Communication Group names must be unique.

Moving Client Nodes from one LDT Communication Group to another LDT Communication Group

If you are repurposing a CTE client node that is currently used for LDT over NFS/CIFS in an LDT Communication Group, but still runs CipherTrust Transparent Encryption and is still registered to the same CipherTrust Manager, then you must **remove** all CIFS/ NFS LDT GuardPoints from the host prior to removing the host from the LDT Communication Group. Then you can reboot the host and add that CTE client node to another LDT Communication Group.

LDT GuardPoint Group

As described, an LDT GuardPoint Group is a group of three or more CTE hosts sharing access to the same GuardPoint over NFS/CIFS. A CTE host with n number of GuardPoints has membership to n number of distinct GuardPoint Groups and is specific to the GuardPoint directory. A CTE host guarding multiple GuardPoints over NFS/CIFS is a member of multiple LDT GuardPoint Groups.

LDT GuardPoint Group Primary

One CTE host in each LDT GuardPoint Group assumes the **Primary** role for that group. The CTE host that enables a GuardPoint directory first is assigned the primary role for the group associated with the GuardPoint directory. Subsequent hosts enabling the same GuardPoint are assigned **Member** or **Secondary** status. A CTE host can be a primary for a group, or multiple groups, and a secondary for other groups.

LDT manages primary host departure from the group during transformation. Departure of a primary host can occur as the result of disabling a GuardPoint directory on the primary host, or an unexpected failure of a primary host. In the event of the primary host departure from the group, LDT elects and delegates the primary role to one of the members of the group. Refer to [Failover](#) for additional information.

LDT GuardPoint Group Capability Level

In CTE 7.6.0, LDT NFS introduces a feature called: Capability Level, which is for an LDT Communication Group. The feature is based on the compatibility level for each member in the group. The Capability Level of a member determines what features that host can access and use, and allows the group to decide on a common capability level at which to run. This allows support for multiple versions of CTE in the same LDT

Communication Group as the group runs at the capability level of the least capable member. The capability level of the group is calculated when a member leaves the group, or a primary failover occurs. As lesser capable members depart the group, the overall capability level of the group increases. The capability level of a group cannot be decreased. As such, members at a lesser capability level cannot join once the capability level of the group has increased. New members joining an existing group will attempt to join with the highest capability level supported by the host and, if rejected, retry with progressively lower capability levels until the group accepts the join request. If the group is running at a capability level that exceeds the joining member's highest capability, that host is not allowed to join the group.

When a new feature is introduced that is unavailable to, or incompatible with, older versions of CTE agents, the maximum capability level of the newer agents is increased to a value greater than the maximum capability level of the older agents. If there is a group with a mix of the new and old agents, the group runs at the capability level of the older agent. This means that no member of the group will be allowed to use the new feature, regardless of the actual version of CTE running on the member. Only after all of the members running the older agent have left the group will the group be upgraded to the new capability level and use of the new feature is enabled. Older agents will no longer be allowed to join the group as there are now features in use that are not available to, or compatible with, older agents. If the LDT GuardPoint Group needs to contain the older agent, then a member running that older agent must be present at all times to ensure that the compatibility level of the group is not upgraded to the level of the newer agents.

Note

CTE 7.5.0 members are automatically assigned to a maximum capability level of **two**, while CTE 7.6.0 members can run at a maximum capability level of **three**. The difference in capability between CTE 7.5.0 and CTE 7.6.0 is due to the inclusion of sharing LDT NFS GuardPoints with Windows. See [Windows/Linux Compatibility](#) for more information on this feature.

To find the capability level of an LDT GuardPoint Group, type:

```
voradmin ldt group capability <GuardPoint Path>
```

This command outputs the capability level that the current host is running at for the specified GuardPoint. If the host is the primary host, then the value also represents the capability level for the entire group.

Designation of Primary role to a host within an LDT GuardPoint Group

The first host that enables a GuardPoint over NFS/CIFS share is designated as the primary host for the LDT GuardPoint Group. The primary designation persists until the GuardPoint is disabled or CTE services on the primary host are stopped. When needed, LDT designates another member of the LDT GuardPoint Group to assume the primary role for the group.

A secondary host is not a proper candidate for promotion to primary status if any of the following conditions exist for the GuardPoint on the secondary host:

- The NFS share is mounted as read-only on the CTE host
- The CTE host has not received the latest policy information
- The latest key version available to the CTE host is not the most recent key version available to other members of the group

If the CTE host fails to perform the required operations for promotion to primary role, the host rejects the promotion request. The election process continues and it selects another host for promotion to primary role.

The read-only access to the GuardPoint directory restricts the designated host from performing LDT operations. LDT operations require read and write access to the NFS share.

The second and third conditions are most likely a transient communication failure between CipherTrust Manager and the designated CTE host. Once the communication issue between CipherTrust Manager and the CTE host is resolved, the CTE host will accept subsequent designation for primary status.

The last condition is usually the result of IO or networking issues between CTE host and the NAS share. If this occurs, the CTE host must be disabled and removed from the LDT GuardPoint Group. Removal of the CTE host will trigger election of another CTE host for primary role. If you are unable to disable the GuardPoint on the CTE host,

you must reboot the host, and then remove the host from the LDT GuardPoint Group using the `voradmin ldt group remove` command.

Responsibilities of the Primary host

As described earlier, the primary host in the LDT GuardPoint Group is responsible for performing data transformation operations. In addition:

1. On Linux, the primary host drives execution of committing a new key version to all members of the LDT GuardPoint Group. During this process, the primary host sends a message to all members, inquiring if each member has received the new key version from CipherTrust Manager. If all members have received the new key, then the primary host, and the members, commit the new key version to their GuardPoints. If at least one member does not have the new key version, the primary host delays committing the new key version and retries the commit process in 30 seconds.

The delay in committing the new key version avoids applications accessing files encrypted with the new key version. Without the delay, the primary host would proceed to rekey the GuardPoint, and consequently, applications running on those member(s), without the latest key version, would be blocked when accessing files encrypted with the latest key version.

Note

The delay only affects rekey operations re-encrypting existing files to the new key version. It does not affect newly created files which inherit the latest key version, that may not have been committed across the entire LDT GuardPoint Group by the primary host. If newly created files are accessed on CTE hosts which do not have the latest key version, the process of accessing such files will be blocked until the latest key version becomes available on those CTE hosts.

2. On Linux, the primary host is also responsible for updating and committing LDT metadata for file operations in progress on the secondary hosts. For example, truncating a file while the primary host is rekeying the file requires changing the LDT metadata for that file. As the primary host is updating the LDT metadata during the rekey process, the member host truncating the file does not update the metadata that the primary host is changing. Instead, the secondary host sends a

message to the primary host, requesting LDT metadata changes for the truncate operation, and the primary host provides the metadata updates for the truncate operation triggering the secondary to proceed. The host then commits or discards the metadata changes for the truncate operation after the secondary hosts completes it and notifies the primary host of their status. This process forces the primary host to engage with the hosts that perform truncate operations, strictly on those files that the primary host is rekeying.

3. Operations such as file or directory rename initiated on a member of the group also involves engagement of the primary host, if the primary host is actively rekeying the affected files or GuardPoints. Renaming a subdirectory on a secondary host engages the primary host to scan a list of files under the renamed subdirectory. The primary host performs the scan outside of the scope of the GuardPoint level rekey process and starts separate file level rekey operations for each file not undergoing rekey in the renamed subdirectory. The primary host conducts up to 8 concurrent file level operations, and the GuardPoint level rekey operation does not complete until every file in the renamed subdirectory is rekeyed. File level rekey operations are slow as compared to GuardPoint rekey operations, therefore, subdirectories with a large number of renamed files delay rekey completion time.

Responsibilities of non-primary hosts

When secondary hosts in the LDT GuardPoint Group perform file operations on files that the primary host is actively rekeying, it must notify the primary host of those operations. Truncating a file on secondary hosts, as described earlier, is an example of a secondary host notifying the primary host for updating LDT metadata.

Secondary hosts can also be selected for promotion to the primary role within an LDT GuardPoint Group. A host elected for promotion checks whether it's capable of assuming the primary role for the LDT GuardPoint Group, and if capable, it accepts and promotes itself to assume the primary role for the LDT GuardPoint Group.

Note

Ensure that at least one secondary host is always qualified to accept promotion to primary status. Connectivity to CipherTrust Manager and mounting NFS shares with read and write access are critical for qualifying secondary hosts for promotion to the primary role.

GuardPoint Directories (Linux Only)

The pathname of a GuardPoint directory does not have to be fixed across all members of the group. For example, if host_1 mounts the share at `/nfs-host-1/gp-dir` and host_2 mounts the same share at `/nfs-host-2/gp-dir`, the LDT GuardPoint Group for the GuardPoint represents both pathnames.

GuardPoint Directories (Windows Only)

GuardPoints for Windows directories use the UNC format. See [Creating a GuardPoint on a CIFS Share Drive on CipherTrust Manager](#) for more information.

Overhead in LDT over NFS/CIFS

The primary host performs LDT operations across all members of an LDT GuardPoint Group. It sends a message, for each operation, to all members of the LDT GuardPoint Group and waits for acknowledgment from all members before executing the operation. A message holds information specific to an operation that the primary host wants to perform, such as file initialization for rekey, lock/unlock operations, LDT launch on GuardPoint, etc. Each member of the LDT GuardPoint Group validates each operation requested by the primary host before responding to the request. This coordination incurs significant overhead in execution of LDT operations on GuardPoints over NFS/CIFS shares in contrast to the same operations performed over GuardPoints in local file systems. It also increases network traffic across all members of LDT GuardPoint Group and potentially increasing overhead in your network infrastructure.

This overhead can pose challenges for LDT to achieve the desired rekey IO rate specified under QoS. Delays incurred in delivery of requests and responses can slow down LDT operations such that the expected rekey IO rate becomes difficult to achieve. Consider the bandwidth of your network infrastructures, number of GuardPoints on the primary host and number of members sharing access to those GuardPoints before setting the rekey IO rate. With multiple GuardPoints on the same primary host such that some of those GuardPoints are over NFS/CIFS and others on locally attached storage devices, rekey IO rate on GuardPoints over NFS/CIFS shares will likely be significantly lower.

Failover

CTE clients in an LDT Communication Group have a master/replica relationship. The first client that joins the group is initially designated as the master node. The other

clients are replica nodes. The master node must be functional in order for communications to persist. All communications happen through the master node.

A host can leave the LDT Communication Group in one of two ways:

- **Graceful Exit**

One CTE client is powered off, but there are other live CTE clients in the LDT Communication Group so failover can occur and another CTE client in the group is elected to become the master.

- **Ungraceful Failure**

One or more of the CTE clients in an LDT Communication Group fail abruptly. (Typically due to a kernel crash, hardware failure or power cycle.)

Graceful Exit for the LDT Communication Group Master

There are two cases for graceful exits:

- **Reboot**

If the master node becomes unavailable because it's rebooted by the administrator or as part of a maintenance cycle, but there are other live CTE clients in the LDT Communication Group, then the LDT Communication Group automatically elects a replica CTE client to become the new master node and fails over to it.

- **Power Off**

When a master node powers off, a new master is automatically elected, but if the initial master node is being decommissioned, then you should remove it from the LDT Communication Group.

Note

Any CTE client that is being decommissioned, regardless of its role in the LDT Communication Group, needs to be removed from the LDT Communication Group. See [Remove].

Ungraceful Shutdown/Failure for the LDT Communication Group Master

An ungraceful failure is when a CTE client fails abruptly due to a hardware or software failure. It means that the administrator did not trigger the client to either shutdown, reboot or become unreachable, but the client still transitioned into that state. This is applicable to the master node failing as well.

If an LDT Communication Group master node fails ungracefully, it is handled automatically. A new master is elected by the remaining nodes. Meanwhile, LDT operations may continue to fail for all LDT GuardPoints. Operations will resume once a new LDT Communication Group master node is functional.

In the event that all, or a majority of, the CTE clients in an LDT Communication Group fail (**ungraceful** shutdown):

Ensure that **all** of the CTE clients in the LDT Communication Group are active (rebooted, powered on and accessible through the network), to reestablish the cluster.

1. All LDT NFS/CIFS GuardPoints must be unguarded before you remove a CTE client from the LDT Communication Group.
2. Once a CTE client is removed from the LDT Communication Group, it must either be rebooted or shut down.
3. Whenever a graceful exit, or failover event, is in progress for the LDT Communication Group, LDT operations will be affected across the entire LDT Communication Group, which means that GuardPoints could be affected.

Failover for LDT GuardPoint Group

If a primary host exits **gracefully**, the LDT GuardPoint Group automatically fails over to a secondary host. All CTE clients guarding the share, or trying to guard the share, depend on the primary being available.

If the primary host fails **ungracefully**, then an administrator will need to manually remove the primary from the group so that another host can become the primary host and guarding can resume normally.

- To identify the primary node, type:


```
# voradmin ldt group info /<guardpoint>    \\Windows
# voradmin ldt group list <guardpoint>    \\Linux
```

- To remove a primary host from the group and automatically select another host as the primary, type:

```
# voradmin ldt group remove <hostname> <guardpoint>
```

Note

Do not use this command in the case of a temporary failure like a crash and reboot.

If a secondary host fails ungracefully, you can use the same command to remove that host.

- If a primary node crashes and reboots quickly, and is trying unsuccessfully to guard a share, you can repair it with the following command:

```
# voradmin ldt group repair /<guardpoint> (Windows)
# voradmin ldt group repair <guardpoint> (Linux)
```

You can also use the repair command to remove a secondary node that is no longer guarding a GuardPoint but still shows in the Group Info list. This command repairs the host and keeps it in the group.

Secondary Host Failure

In the event that a secondary host crashes, you must first remove that failed secondary host from the entire LDT GuardPoint Group that the host was a member of before allowing that CTE host to rejoin that same LDT GuardPoint Group after the host is restored. The following steps must be performed in exact order for each GuardPoint that was enabled on the host prior to the crash. (You can get the list of GuardPoints from CipherTrust Manager.) For each GuardPoint:

1. On any CTE client that is a member of the LDT GuardPoint Group for that GuardPoint, run the `voradmin ldt group list` command to identify the failed host. The role of the failed host is UNRESPONSIVE within the group, type:

```
# voradmin ldt group list <GuardPoint Path>
```

Role	State	Hostname	GuardPoint Path
UNRESPONSIVE	N/A	Host-3	N/A
PRIMARY	JOINED	Host-1	/nfs/gp

2. On the primary CTE client for the GuardPoint , run the `voradmin ldt group remove` command to remove the failed host from the LDT GuardPoint Group , type:

```
# voradmin ldt group remove <hostname> <GuardPoint Path>
```

LDT GuardPoint Group Commands

Prerequisites for CIFS Share drives

Prior to running the `voradmin ldt group` commands, you must:

1. Ensure that you have access and permission to read/write to/from the CIFS share.
2. Mount the CIFS share using the `net use` command, or login to the CIFS shares using Windows Explorer.

Group check

The `group check` command is used to send out an internal ping to all hosts that are registered with the master as part of the LDT Communication Group. Each functioning host responds. Then the host displays a message of the hosts in the LDT Communication Group that did not respond. This allows an administrator to easily find the problem.

Syntax

```
voradmin ldt group check <gp_path>
```

Example (Windows)

```
voradmin ldt group check \\192.168.1.160\share
```

Hostname	Role	State	GuardPoint Path
192.168.1.161	PRIMARY	JOINED	\\192.168.1.160\share
192.168.1.162	SECONDARY	JOINED	\\192.168.1.160\share

Example (Linux)

```
voradmin ldt group check /nfs1/gp
```

Role	State	Hostname	GuardPoint Path
PRIMARY	JOINED	host1.domain	/nfs1/gp1
SECONDARY	JOINED	host2.domain	/nfs1/gp1

Info

Use the `info` command to display information about the cluster.

Syntax

```
voradmin ldt group comm_info <gp_path>
```

Example (Windows)

```
voradmin ldt group comm_info \\192.168.1.160
```

```
LDT Communication Group Master IP:      10.171.34.111
LDT Communication Group Master Port:    7024
LDT Communication Group Name:          tanmay_windows
LDT Communication Group Servers:
10.171.34.111,10.171.34.120,10.171.34.119
```

Example (Linux)

```
voradmin ldt group comm_info /nfs1/gp1
```

```
LDT Communication Group Name:          tanmay-linux
LDT Communication Group Master IP:    10.171.47.121
LDT Communication Group Master Port:  7024
```

```
LDT Communication Group Servers:  
10.171.47.241,10.171.47.121,10.171.47.75
```

Remove

This command allows a client to be manually removed from the group safely. Use the `Remove Host` command if a client dies without properly leaving the group.

Note

Only run this command on clients which will be powered off. Do not run it on a client which may simply have temporary network issues.

Syntax

```
voradmin ldt group remove <hostname> <GuardPoint pathname>
```

Example (Windows)

```
voradmin ldt group remove host42 \\server\share
```

```
WARNING: This command will forcibly remove host42 from the LDT group  
for gp_path  
You should only perform this step if the host is guaranteed to be  
powered off, otherwise a  
loss of data may result. Do you wish to continue? (Y/N)
```

Example (Linux)

```
voradmin ldt group remove host2 /nfs-ldt  
  
host2 has been removed from the group for GuardPoint /nfs-ldt
```

Sharing LDT NFS/CIFS GuardPoints between Linux and Windows

An LDT GuardPoint Group is used to organize clients guarding the same NFS or CIFS share. In CTE 7.5.0 or previous versions, each group could only contain Linux or Windows members, and not a mix of the two. Starting with CTE 7.6.0, it is now possible to have Windows clients as AccessOnly clients to guard a share that is part of a Linux LDT GuardPoint Group. Refer to [Configuring CTE Windows LDT AccessOnly nodes](#) for more information on AccessOnly clients.

Windows LDT is capable of mixing AccessOnly, and non-AccessOnly clients, in an LDT GuardPoint Group accessing a share through CIFS protocol. This capability is possible because of the CIFS protocol supporting exclusive access to a file from a remote node. Linux LDT does not support AccessOnly clients because the NFS protocol does not support exclusive file access capability. Therefore, an AccessOnly client is not privy to LDT messages between clients. The Windows client can only access a Linux based LDT GuardPoint Group while a GuardPoint is rekeyed. As such, Windows clients can only guard after the Linux primary client has completed rekey on the GuardPoint. Subsequently, the GuardPoint must be unguarded on Windows clients before any subsequent key rotation begins. Failure to do so will cause data corruption as any write operation from the Windows client may write without any coordination with the Linux primary client performing transformation.

The capability to allow Windows and Linux clients to guard the same GuardPoint is supported on GuardPoint directories which have been guarded for the first time, and completed the initial transformation, by a CTE 7.6.0 Linux client. This restriction is in place because there are metadata format changes in newly created GuardPoints to support sharing access between Linux and Windows clients.

Note

Access to shared GuardPoints is available through NFS on Linux and CIFS on Windows.

Windows clients are blocked from accessing files marked with a lazy_rekey flag. This is because the files marked with a lazy_rekey flag will launch the data transformation

process the next time a Linux member of the LDT GuardPoint Group accesses the file without any awareness of Windows clients sharing access to the GuardPoint.

Although there is no communication between Windows and Linux clients for sharing access to GuardPoints, there is awareness of shared access across the clients because of the LDT metadata changes applied to the newly created GuardPoints for sharing on both platforms. Both platforms apply compatible LDT metadata when creating new files.

Finally, suspending rekey does not qualify Windows clients to guard and share access. Rekey must be completed before Windows clients can attempt to guard.

LDT Communication Groupit

All members of an LDT Communication Group must be Linux CTE clients. Windows clients do not belong to LDT Communication Groups because of AccessOnly mode configuration.

Limitations

- Only new GuardPoints are supported in v7.6.0.
- All existing CTE clients must be upgraded to CTE Agent v7.6.0, or a subsequent version.
- Guarding a GuardPoint for the first time on Windows disqualifies the GuardPoint for sharing between Windows and Linux, and subsequent attempts on a Linux client to guard.
- Windows AccessOnly client fails to guard the GuardPoint while rekey is suspended.
- All Windows clients must unguard before the key rotates on CipherTrust Manager.

Limitations

The following limitations exist for LDT over CIFS/NFS:

Windows

- Only supported for customers using unstructured data.
- LDT over CIFS is not supported on single processor systems.

- All nodes sharing a CIFS drive **must** have homogenous system drivers installed. You **cannot** have some nodes installed with the `vmlfs.sys` driver and some installed with the `vmfiltr.sys` driver. That scenario is not supported.

Note

To determine which driver your node has installed, type: `fltmc` from the administrative command line.

Linux

- Although CTE supports GuardPoints over NFS shares using NFS v3 and v4 protocols, Thales strongly recommends using NFS v4 protocols, or a subsequent version, when using LDT policies.
- CTE does not support mixing NFS v3 and v4 protocols on different CTE hosts for accessing the same GuardPoint. All members of an LDT GuardPoint Group must access the same GuardPoint using the same NFS protocol version.
- When installing CTE patches, make sure that the new patch is compatible with CTE latest release. Follow instructions in the release note for a patch release, if any, for upgrading CTE hosts sharing access to the same GuardPoints across multiple hosts.
- LDT can be slower when rekeying GuardPoints over NFS as compared to GuardPoints in local file systems. Communications between a primary host and other members of an LDT Communication Group add significant delay in execution of LDT operations. Delays in LDT operations slow down LDT progress to the point that QoS rekey IO rate may not be achievable, especially in environments with slow network bandwidth.
- Other current limitations will be removed over the next several patches. Refer to the release notes to determine which limitations, noted below, have been removed:
 1. Applications will experience performance degradations on secondary hosts when targeting files undergoing rekey, even while LDT is suspended. LDT disables read-ahead on files undergoing rekey on secondary hosts resulting in lower IO throughput.
 2. Loss of network connectivity to the NAS server from the primary host during active rekey can block access to files undergoing rekey across all members of LDT GuardPoint Groups for all GuardPoints on the NAS server.

3. If the secfsd daemon fails on a secondary host, it will require rebooting the host and removing the secondary host from each LDT GuardPoint Group on that secondary host prior to reboot. If the secfsd daemon fails on the primary host, it also requires rebooting the primary host and the delegation of the primary role to one of the secondary members of the LDT GuardPoint Group.

LDT Communications Group

- When you create an LDT Communications Group, add the CTE clients sequentially.
- CTE 7.5 has architectural design changes. Due to these changes:
 - The first three nodes of the LDT Communication Group are critical and **cannot** be removed or shutdown/reboot simultaneously.
 - All nodes, Windows and Linux, **must be running** CTE 7.5.0 in order for the nodes to communicate properly.
- LDT over CIFS/NFS **DOES NOT** support rolling back migrations.

LDT GuardPoint Group

Failover, after an ungraceful failure, is manual. If a primary node becomes unavailable and LDT cannot automatically reassign the primary to another CTE client, the administrator must manually assign another CTE client as the primary node. See [Failover](#) for more information.

Prerequisites

For LDT over CIFS/NFS to work properly, see Installation Prerequisites of the CipherTrust Quick Start Guide.

Space Required

The space requirements for share drives is the same as the requirements for local file systems. However, you must plan for additional space for metadata. Refer to [Planning for CTE-LDT Attribute Storage](#) for calculating space.

Supported Versions

Linux

Upgrade your CTE Agents to CTE v7.2.0 or later, and your CipherTrust Manager to version 2.7 or later, in order to support LDT GuardPoints on NFS shares. It is not supported for earlier versions. You can use the standard upgrade procedure for both the CTE Agents and CipherTrust Manager. See the [CTE Agent for Linux Quick Start Guide](#) for more information.

Windows

On Windows, you must install the full version of CTE 7.2. Guarding CIFS/SMB shares with LDT GuardPoints is not supported in the upgrade version of CTE. Refer to the [CTE Agent for Windows Quick Start Guide](#) for more information. Requirements for GuardPoints

On Windows, the requirements for GuardPoints created on CIFS shares using CTE-LDT are different from those created on local directories or for those created on CIFS shares using standard CTE policies. You cannot mix the two on a single host. Instead, each host must contain either CTE-LDT CIFS share GuardPoints or local directory/standard policy GuardPoints.

If you want to use CTE-LDT on a CIFS share which contains GuardPoints on a host on which the CTE Agent is already installed, you must:

1. Remove any existing local directory GuardPoints.
2. Uninstall the current version of CTE Agent from the host.
3. Install CTE Agent version 7.5.0 or higher and respond **Yes** when prompted about guarding network shares.

If you do not see this prompt during installation, make sure that you have fully uninstalled the previous version of the CTE Agent.

Administering LDT over CIFS/ NFS with CipherTrust Manager

The following sections contain information about the administration tasks required for using LDT over CIFS/NFS.

- [Adding a CIFS Connector for CipherTrust Manager](#)
- [Configuring CTE Windows LDT AccessOnly nodes](#)
- [Policy and GuardPoint Management](#)
- [Key Rotation Commitment](#)
- [LDT Metadata Management Over NFS/CIFS Shares](#)
- [Backup and Restore LDT on NFS GuardPoints](#)

Adding a CIFS Connector for CipherTrust Manager

In order to create GuardPoints on a CIFS share, you must obtain credentials for the CIFS share and add it as a connector in CipherTrust Manager.

CIFS Credentials

Prior to guarding data on the CIFS share, LDT needs permissions to access the share drive and its data and for saving metadata to the CIFS share. CipherTrust Manager stores the CIFS Share credentials centrally. Credentials are protected similarly to the way CTE protect keys and policies. These credentials are associated with a GuardPoint. The CM pushes the credentials to the host and LDT uses them. Thales recommends customers create a new LDT account that contains a user name, password and domain, to access the CIFS share with read/write permissions. Users must not share their user accounts.

Adding the CIFS Share to CipherTrust Manager

In order for LDT to access that CIFS share and encrypt the data, you must provide the credential data when you add a CIFS share as a connector in CipherTrust Manager:

1. Log in to CipherTrust Manager as an administrator.
2. On the main screen, in the sidebar on the left, click **Access Management > Connections**.
3. Click **Add Connection**.
4. Click **File-Share**.
5. In the Select File-Share type field, select **CIFS/SMB** and click **Next**.
6. Enter a name and description for the connection and click **Next**.
7. Enter appropriate information for the fields:
 - **Host**: IP or FQDN of the CIFS share server
 - **Port**: Port where the CIFS service is running on the host
 - **Username**: Username to access the CIFS share
 - **Password**: Password to access the CIFS share
 - **Domain**: Domain under which the username is configured (Optional)
 - **Path**: Path to the CIFS file-share for which the credentials need to be tested (Optional)

Note

If you want to change a user name, domain or password of a CIFS credential in a CM connection, then you need to create a new connection on CM and change the GuardPoint to use the new connection.

Add Connection



- 1 Select Connection Type 2 General Info 3 **Configure Connection** 4 Add Products

Host* Port*

Username*

Password*
 Show Password

Domain

Test Path*

● Status: OK
Tested 19 November 2021 08:52

[Back](#)

[Next](#)

8. Click **Next**.

9. In the Products list, select **CTE** and click **Save**.

LDT AccessOnly nodes

- [Configuring CTE Windows Deployment for LDT AccessOnly nodes](#)
- [Configuring LDT for CIFS shares \(with LDT AccessOnly nodes\) Mapped to Multiple IP addresses](#)

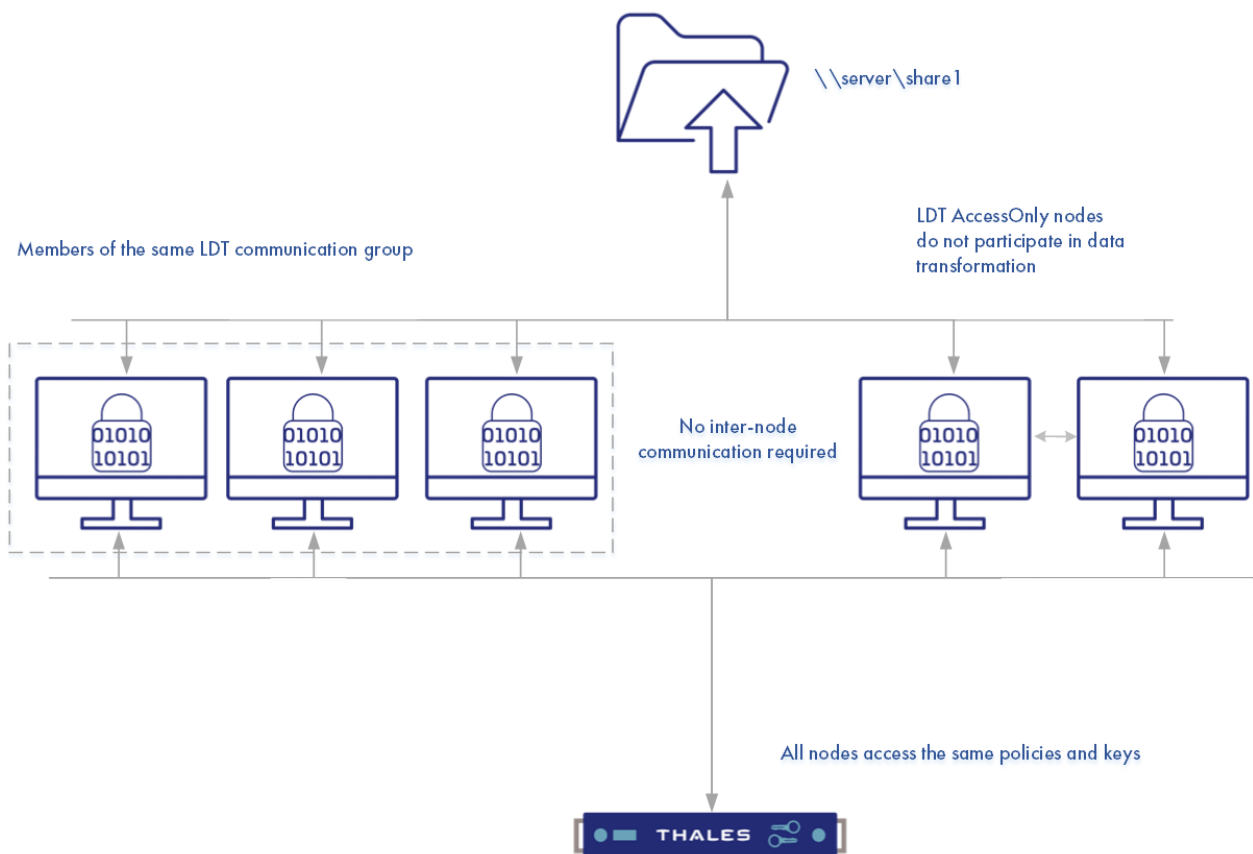
CTE Windows Deployment for LDT AccessOnly nodes

Currently, Windows clients that attempt to access CIFS paths must be part of an LDT Communication Group. The LDT Communication Group assigns the role of primary node to one of the clients, which performs LDT rekeying of data. All of the other clients are assigned as secondary nodes, and they do not participate in rekeying data.

There are use cases that restrict users from adding all of the nodes accessing a GuardPoint to the same LDT Communication Group, such as:

- All of the nodes accessing the GuardPoint are not able to communicate to other servers in the LDT Communication Group due to differences in security policies.
- Certain nodes, which go through frequent network disconnects, are not recommended for participation in data transformation using LDT. For example, windows endpoint protection clients or laptop users.

To overcome these limitations, CipherTrust Transparent Encryption has a new feature where systems will not be allowed to become part of an LDT Communication Group or participate in data transformation, but they will continue to access a protected LDT CIFS GuardPoint.



The new role, **LDT AccessOnly**, can be assigned to any CipherTrust Transparent Encryption Windows client and this client will not be added into an LDT Communication Group and will not participate in LDT data transformation or rekey.

Note

- An AccessOnly node can set a GuardPoint, however, that system will not be the system to rekey the GuardPoint.
- When an LDT rekey is in-progress, AccessOnly nodes cannot access the files currently being rekeyed.

Warning

All CTE agents in an LDT Communication Group that contains CTE agents with AccessOnly nodes, **must** have CTE v7.4.0 or subsequent versions, installed. If you have agents with versions previous to CTE v7.4.0 in that LDT Communication Group, those agents must be upgraded.

LDT will operate in either of two modes for rekeying the data in the CIFS GuardPoint paths:

- **Hybrid mode:** LDT will operate in **hybrid mode** when a node with the **LDT AccessOnly** role is guarded with a CIFS path. In this mode, a primary node will open a file exclusively for LDT operations. If any node designated as a **LDT AccessOnly** or Secondary node tries to access the same file, then that node will be denied access to that file. After access is denied, LDT will wait for a short while and then try to obtain access again.

Note

If the LDT process fails to open a file exclusively, it will skip the file and retry on the next pass of LDT.

- **Full mode:** LDT will operate in **full mode** when all of the nodes in the LDT Communication Group are designated as only primary or secondary nodes.

Note

- Thales recommends that you use QoS and configure the schedule so that LDT is running in the background, during off hours, to avoid any contentions.
- All LDT nodes in the LDT Communication Group, and all of the AccessOnly nodes, must all have CipherTrust Transparent Encryption v7.4.0 or a subsequent version. LDT AccessOnly nodes is not a backwards compatible feature.

Setup and Configure on CipherTrust Transparent Encryption

- During Agent registration, a new option to check for **AccessOnly (No LDT Transformation)** displays. If user enables it, then the option to specify an LDT Communication Group name is disabled.

Setup and Configure on CipherTrust Manager

- Each client will have a client-specific option to check that enables **AccessOnly (No LDT Transformation)**. Once checked, that client cannot be added to an LDT Communication Group.

Backup/Restore from Access-Only nodes when LDT is in progress

- While LDT is in progress, backup/restore from access-only nodes is not recommended. Backup/Restore must be performed from **Primary** or **Secondary** nodes in the LDT Communication Group. This is because while LDT is in progress, files under rekey may not be accessible from access-only nodes. This may cause the Backup/Restore process to fail as file access is denied.

Configuring LDT for CIFS shares Mapped to Multiple IP addresses

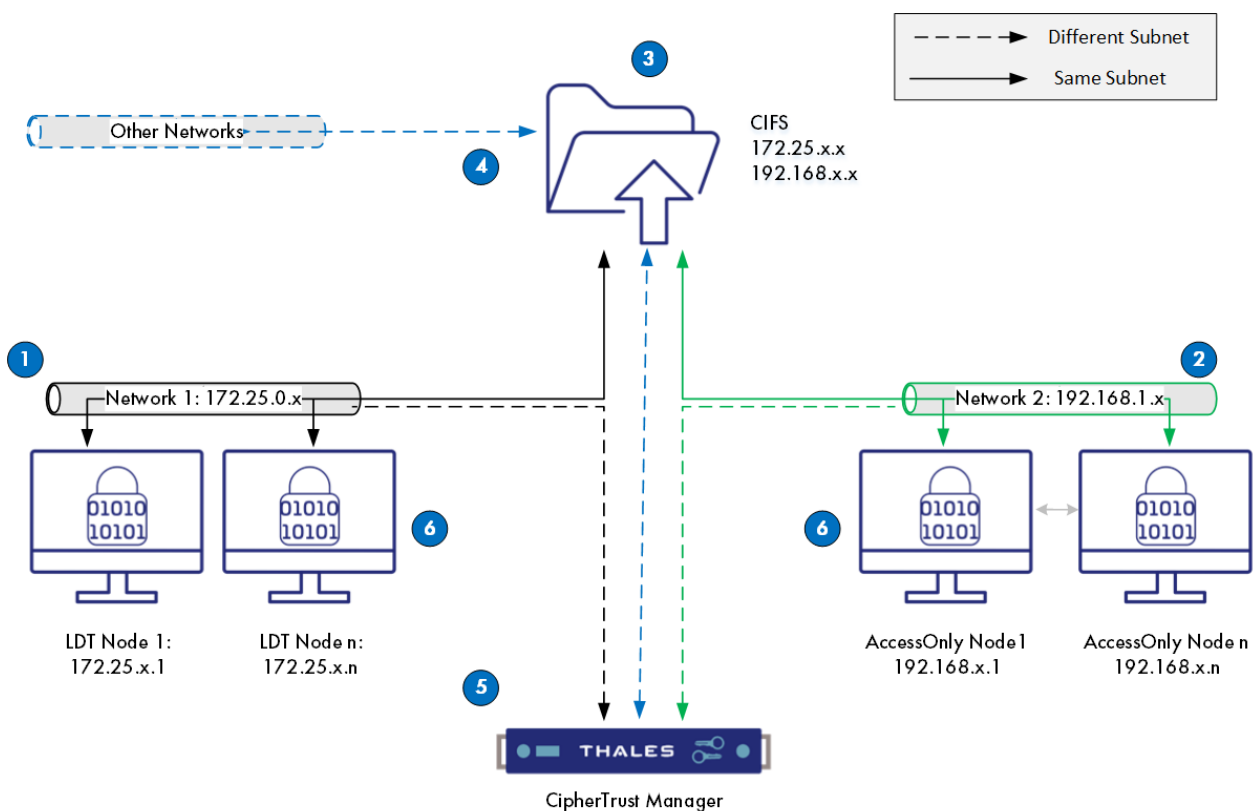
This feature allows you to apply the same GuardPoint to two IP addresses that point to the same CIFS share. This can be useful because different applications may use different IP addresses for accessing data.

Note

This feature is available only when using LDT AccessOnly nodes.

Network Setup for CIFS shares Mapped to Multiple IP addresses

The following illustration shows how to setup your network. The table below explains the setup in more detail:



In the above sample diagram:

#	Description
1	LDT nodes are connected to Network 1
2	LDT AccessOnly nodes are connected to Network 2
3	CIFS server is connected to both Network 1 and 2, and have one IP address corresponding to each network
4	CIFS server may also be connected to more networks, based on use case and requirement
5	CipherTrust Manager can be connected to Network 1, 2 or some other network
6	Network 1 and 2 nodes can reach CipherTrust Manager and vice versa

Creating the Setup

1. [Register LDT nodes](#) from Network 1 on CipherTrust Manager and add them to the LDT Communication Group.
2. [Create a CIFS\SMB share connection on CipherTrust Manager](#), using one of the two IP addresses of the CIFS machine, preferably IP 1.
3. [Register the LDT AccessOnly nodes](#) from Network 2 to CipherTrust Manager.
4. Create two [Client Groups](#): one for the LDT nodes and one for the LDT AccessOnly nodes
5. [Apply GuardPoints](#) in both client groups using the respective network IP addresses.
6. Access the protected CIFS path using the respective IP addresses.

Policy and GuardPoint Management

You can use the same CTE-LDT policies that you use for protecting GuardPoint directories in local file systems and NFS/CIFS shares. The only exception is that for CTE-LDT protected GuardPoints, it only supports backup/ restore of files that do not use Apply Key.

Security Rule Ordering for Policies

If you want to enforce restrictions when guarding NFS shares using an LDT or standard policy with a **CBC-CS1** key, note the following:

CipherTrust Transparent Encryption embeds and hides LDT and/or IV (initialization vector) attributes in the first 4K of files for NFS shares guarded with an LDT or standard policy with a **CBC-CS1** key. Embedding CipherTrust Transparent Encryption attributes increases the actual file size by 4K, and CTE hides that extra 4K when reporting the file size. The exception to this is when a backup/restore process reads/writes such files. This requires embedded attributes to be read/restored by the backup/restore process. In such cases, CipherTrust Transparent Encryption does not hide the 4K attribute space in the file. The backup user/process views the actual file size. Non-backup users/applications view the file size as less than 4K.

If you want a security rule to enforce restricted access for reading file level attributes on such GuardPoints, you must specify the **Apply Key effect**. Alternatively, you can place the security rule that is enforcing the restricted access after the rule granting read/write access. This avoids application failure if the Apply key effect is not desired. For example, the order of the two rules in a policy that does not hide the user user-name would be:

- **Security Rule n**

Rule	Value
User	<user-name>
Action	Read-file-attribute and/or Read directory
Effect	permit

- **Security Rule n + 1**

Rule	Value
User	<user-name>
Action	all_ops
Effect	permit, apply-key

Assuming <user-name> is not affiliated with backup/restore operations, <user-name> would view the actual file size which is 4K larger than the size of the user data in the file. The returned file size can result in failure when user-name attempts to read/write

files. By reordering rules n and $n + 1$, `<user-name>` will view the correct size hiding the 4K attribute space in the target file.

For more information, see [Adding Security Rules](#) in the CipherTrust Manager documentation.

Creating a GuardPoint on a CIFS Share Drive on CipherTrust Manager

Creating GuardPoints on a CIFS share drive requires you to add the CIFS share drive to CM as a connector first, and then select it as a network drive.

To add the CIFS share in the CipherTrust Manager Connection Manager, see [Adding a CIFS Connector for CipherTrust Manager](#).

Note

For the host name, use the IP address if the host name does not work.

GuardPoint Naming

When creating GuardPoints in a CIFS share, you must use the UNC name. A Universal Naming Convention (UNC) format name defines the location of files and other resources that exist on a network. UNC provides a format so that each shared resource can be identified with a unique address.

UNC names must conform to the `\\SERVERNAME\SHARENAME` syntax, where `SERVERNAME` is the name of the Provisioning Server and `SHARENAME` is the name of the shared resource.

- UNC names can also include the directory path:

```
\\SERVERNAME\SHARENAME\DIRECTORY\FILENAME
```

- To define a folder that contains a configuration database file in:

```
C:\Program Files\Oracle\customer_data
```

- On the shared server, enter:

```
\\server1\customer_data
```

Selecting the CIFS Share Drive for Guarding

When you create a GuardPoint for CIFS shares, you will have more options when browsing for the file system to guard. You can select to guard a local path, or a network path.

To create the GuardPoint on the CIFS share:

1. In CTE, select **Clients**.
2. Click on the Client to access it.
3. Click **Create GuardPoint**.
4. Select a **policy** and **type**.
5. For Path, click **Browse**.
6. In the Browse Path dialog, select **Network Path**.

Browse Path

Path Type: *

Local

Network Path

User Name: *

Thales-Admin

Password: *


.....

Domain

CPL_Internal

Search Network Path * ⓘ

\\192.8.0.14\cifs_share1


C:
D:

7. Enter the **User Name** and ****Password** for the CIFS share drive.
8. Enter a **domain/IP address** for the CIFS share drive.
9. Enter the **UNC name** for the Network Path.
10. Click **Refresh** to test the connection.
11. Select the directories to guard.
12. Click **Add**.
13. In the Create GuardPoint window, select the **SMB** connection to use.
14. Click **Create**.

Create GuardPoint [X]

Policy: *
Pol1-LDT-CIFS [Select]

Type: *
Auto Directory [v]

Path: *
 Enter/Browse Path Upload CSV
 W10.171.83.35\Gillian_CIFS_Share\Test-GP\' [Browse]

! To apply the same GuardPoint settings to multiple paths, type paths in the Path field or specify them using the Browse button. A maximum of 200 GuardPaths can be specified.

SMB Connection: *
Cifs-10.171.83.1 [v]

Preserve Sparse Region:

Secure Start:

[Cancel] [Create]

Use the GuardPoint Settings on another GuardPoint

1. If you want to guard another GuardPoint on a different path, click **yes** to the question: **Would you like to use these GuardPoint settings on another guard point with a different path?**
2. Select the path for the GuardPoint and click **Create**.

Verifying the GuardPoint

- Before starting the application or accessing the data, you must ensure that the GuardPoints are guarded successfully. Check the status on CM or by running the `voradmin ldt stats` command.
- For Linux, LDT Communication Group member hosts cannot guard or unguard GuardPoints that have single file rekey jobs in progress. Attempts to guard or unguard result in a warning message and the operation is automatically retried until all single file rekey jobs are finished and the state of the GuardPoint can be changed.

Accessing the GuardPoint

If a GuardPoint is applied on a CIFS share on a CTE client, you can only access the data from a CTE client that has CTE installed on it. You cannot access the CTE client from an agent that does not have CTE installed.

Pausing and Resuming LDT per host and per GuardPoint

Rekey operations on a GuardPoint can only be paused or resumed by the primary host for that GuardPoint. You must run the `voradmin` command on the primary host for the GuardPoint. If multiple GuardPoints reside on the same CTE host, and the host is the primary for some of the GuardPoints and the secondary for other GuardPoints, the `voradmin` command is applied only on the GuardPoints for which the CTE host holds the primary role.

The same applies if you pause or resume GuardPoints on a host using CipherTrust Manager. Pausing or resuming LDT on a host from CipherTrust Manager only applies to the GuardPoints for which the host holds the primary role.

Migrating GuardPoints over NFS from or to an LDT Policy

GuardPoints over NFS can be guarded using policies with key rules specifying a mix of CBC or CBC-CS1 keys.

You can migrate to an LDT policy from clear text (no existing policy) or from a standard CTE policy. If the standard policy uses:

- **CBC keys**

The CBC keys are transformed during the initial data transformation when CTE-LDT embeds the LDT metadata in the beginning of each file. CTE-LDT shifts the existing data in the files by 4096 bytes to make room for the LDT metadata.

- **CBC-CS1 keys**

The IV attribute is already embedded in the protected files. CTE-LDT transforms the files in those GuardPoints without shifting the existing data because the required IV attribute already exists.

Migration out of an LDT policy is only *partially* supported because of the shift to the existing data that was done to accommodate the LDT metadata. You can only migrate from a Live Data Transformation policy that uses CBC or CBC_CS1 keys to a different Live Data Transformation policy that uses CBC or CBC_CS1 keys. You cannot remove the Live Data Transformation policy from a guarded directory, and you cannot migrate from a Live Data Transformation policy to a standard CTE policy. The migration support matrix is shown in the following table.

Source Policy Type	Target Policy Type	Supported?
Live Data Transformation using CBC or CBC_CS1 keys	Live Data Transformation using CBC or CBC_CS1 keys	Yes
Live Data Transformation	No policy (unguarded directory)	No
Live Data Transformation	Standard CTE policy	No
Standard Policy using CBC or CBC_CS1 keys	Live Data Transformation using CBC or CBC_CS1 keys	Yes

The only method for migrating an NFS GuardPoint from LDT to clear-text or to a CTE standard policy that uses CBC or CBC_CS1 keys is the following:

1. Perform a full backup of files in the NFS GuardPoint in clear text. Make sure that you disable the security rule for the backup process in the LDT policy, if the security rule skips Apply Key as part of backup operation.
2. Upon completion of full backup, unguard the directory on the key manager and then remove the LDT Private Space directory (`ldtprivspace`) in the NFS GuardPoint using the `voradmin ldt rmltd <GuardPoint>` command.

3. Remove the remaining files inside the NFS GuardPoint directory and restore the full backup of the files in clear-text over the NFS GuardPoint directory. If you are migrating to a standard CTE policy, you can now proceed with re-guarding the NFS GuardPoint directory using the standard policy.

Multiple GuardPoint Pathnames

If you have multiple CTE-LDT hosts guarding and sharing the same CTE-LDT GuardPoint directory with different local pathnames, the pathname that will be associated with CTE-LDT operations is the pathname used on the CTE-LDT host designated with primary status.

When you are entering `voradmin ldt` commands, you must use the primary host's local pathname for the GuardPoint when you specify the GuardPoint parameter. If another host is promoted to the primary status for the GuardPoint, and the GuardPoint directory pathname is different from the pathname on the previous primary host, then CTE-LDT changes the the pathname of the GuardPoint to the pathname of the new primary host.

For example, let's say you have two hosts (`LDT_Host_1` and `LDT_Host_2`) with the same directory in an NFS share mounted on different mount points on each host, and subsequently, you must enter the new primary host's local pathname for the GuardPoint:

- GuardPoint on `LDT_Host_1`: `/nfs-oxf-fs1-host1/gp`
- GuardPoint on `LDT_Host_2`: `/nfs-oxf-fs1-host2/gp`

When the GuardPoint is first added to `LDT_Host_1`, that starts the initial data transformation:

```
secfsd -guard /nfs-oxf-fs1-host1/gp
secfsd: Guardpoint initialization in progress
```

MDS has the GuardPoint configured for rekey at `/nfs-oxf-fs1-host1/gp`.

```
voradmin ldt list all
MDS_1:  type=file, nguards=0, name=/nfs-oxf-fs1-host1/gp/___vorm_mds___
      Guard Table: version 1 nentries 1
      Guard 0:  type=GP, state=REKEYING DIRTY, flags=GP LOCKED, gp=/n
fs-oxf-fs1-host1/gp
          File List: count 4
```


When the GuardPoint is enabled on LDT_Host_2, the host joins the LDT GuardPoint Group for the GuardPoint.

```
secfsd -guard /nfs-oxf-fs1-host2/gp
```

```
secfsd: Path is guarded
```

Disable the GuardPoint on LDT_Host_1 while rekey in progress:

Note

This operation triggers the promotion of LDT_Host_2 to primary status:

```
secfsd -unguard /nfs-oxf-fs1-host1/gp secfsd: Path is not guarded
```

On LDT_Host_2, guard the same directory mounted at the different path:

```
voradmin ldt list all
```

```
secfsd -guard /nfs-oxf-fs1-host2/gp
```

```
secfsd: Path is guarded
```

MDS now has the GuardPoint configured for rekey at /nfs-oxf-fs1-host2/gp on LDT_Host_2.

```
voradmin ldt list all
```

```
MDS_1: type=file, nguards=0, name=/nfs-oxf-fs1-host2/gp/___vorm_mds___
```

```
Guard Table: version 1 nentries 1
```

```
Guard 0: type=GP, state=REKEYING SUSPENDED (qos), flags=GP LOC  
KED, gp=/nfs-oxf-fs1-host2/gp
```

```
File List: count 4
```

Key Rotation Commitment

The primary host drives the execution of the process for checking and committing a new key version across all members of LDT GuardPoint Groups associated with the affected GuardPoints. The primary host commits and launches LDT if all members have also received the same key version from CM. If a member has not received the

new key version, the primary delays launching LDT and continues checking for the availability of the key versions with all of the members of the LDT GuardPoint Group.

The primary host triggers an ALARM on CM if a member of the LDT GuardPoint Group has not received the new key version. The message for this displays as:

```
LDT-NFS-ALERT: Some members of the LDT GuardPoint Group for [GuardPoint Pathname]
have not received the latest key version.
```

After receiving this message, check the status of the connections between CTE hosts and CM to make sure each member is in an active state and in communication with CM, type:

```
voradmin ldt attr get /nfs-oxf-fs1/gp1

LDT stats: version=5, rekey_status=rekeyed
  Number of times rekeyed:          1 time
  Rekey start time:                2022/01/08 09:12:14
  Last rekey completion time:      2022/01/08 09:12:15
  Estimated rekey completion time:  N/A
  Policy key version:              3627
  Pushed Policy key version:       3627
  Policy ID:
    18729
  Data stats:
    total=0.0MB, rekeyed=0.0MB
    truncated=0.0MB, sparse=0.0MB
  File stats:
    total=0, rekeyed=0, failed=0
    passed=0, skipped=0, created=0, removed=0, excluded=0
```

LDT Metadata Management Over NFS/CIFS Shares

CTE manages LDT metadata as extended attributes in local file systems for each file and GuardPoint directory. Because the NFS/CIFS protocol does not support extended attributes, CTE embeds LDT metadata into each file to be encrypted inside the GuardPoint, over NFS/CIFS shares, during the initial data transformation. The size of the LDT metadata is 4096 bytes, so the size of each file in the NFS/CIFS share is

increased by 4096 bytes. The presence of LDT metadata and larger file sizes is not visible to users and applications as long as the GuardPoint remains enabled. The same file stored in a CTE-LDT protected GuardPoint, over NFS/CIFS and a non-protected directory, is identical to other users and applications despite the presence of the embedded LDT attribute and the additional 4096 bytes.

As the Linux example below illustrates, the presence of the attributes embedded during the initial transformation of the file is invisible to users and applications. Note that the size of the file after transformation appears unchanged until the GuardPoint is disabled.

```
ls /nfs-oxf-fs1/gp1

cp /etc/hosts /nfs-oxf-fs1/gp2

ls -l /nfs-oxf-fs1/gp2/hosts
-rw-r--r--. 1 root root 241 Jan  3 16:14 hosts

secfsd -guard /nfs-oxf-fs1/gp2
secfsd: Guard point initialization in progress

voradmin ldt attr get /nfs-oxf-fs1/gp2/hosts
LDT attributes: rekeyed_size=0, rekey_status=none
      Key:      clear_key

ls -l /nfs-oxf-fs1/gp2/hosts
-rw-r--r--. 1 root root 241 Jan  3 16:15 /nfs-oxf-fs1/gp2/hosts

secfsd -unguard /nfs-oxf-fs1/gp2
secfsd: Path is not guarded

ls -l /nfs-oxf-fs1/gp2/hosts
-rw-r--r--. 1 root root 4337 Jan  3 16:15 /nfs-oxf-fs1/gp2/hosts

voradmin ldt attr get /nfs-oxf-fs1/gp2/hosts
LDT attributes: rekeyed_size=4096, rekey_status=none
      Key:      name=LDTNFS_KEY_1, version=1482
```

As noted, CTE also manages the LDT attribute for each GuardPoint directory as an extended attribute. For GuardPoints over NFS/CIFS shares, CTE-LDT stores the LDT metadata for each GuardPoint in the LDT Attribute File associated with the directory. In general, there are multiple metadata files that CTE-LDT manages for GuardPoints over NFS. Those metadata files are stored in the LDT Private Space Directory inside each GuardPoint directory. The directory name of LDT Private Space is `vorm_ldtprivspace`. The directory is created inside the GuardPoint directory at the time of initial transformation. The LDT metadata file for a GuardPoint is also created in the `vorm_ldtprivspace` directory of the GuardPoint at the time of initial transformation. The name of the file is `::vorm:ldtxattr::`, similar to the MDS file which is also protected against user modification or deletion. The size of the LDT metadata file is 4096 bytes. For example:

```
ls -l /nfs-oxf-fs1/gp1/vorm_ldtprivspace/
total 4
-rwxr-xr-x. 1 root root 4096 Jan  4 12:13 __vorm:ldtxattr__
```

Note

The LDT Attribute file is protected, and it can only be manually removed using `voradmin ldt rmldt <GuardPoint Path>`. For details, see [Deleting CTE-LDT Metadata \(Linux\)](#).

Backup and Restore LDT on NFS GuardPoints

Native backup capabilities in an NAS server is the recommended method for backing up and restoring GuardPoints on a NAS share. You may also backup or restore GuardPoints on NAS shares on the host designated as the primary host for an LDT GuardPoint Group, as described earlier.

Backup LDT on NFS GuardPoint

Backups of the NAS servers are remote from an LDT perspective.

It is extremely important that while you are backing up the NAS server:

- LDT is **not** performing data transformation in the GuardPoint

- No client is modifying data in the GuardPoint

This rule applies for:

- Initial data transformation
- Automatic or manual key rotation
- Backups of an entire GuardPoint, or any subset of files within a GuardPoint namespace.

Restore LDT on NFS GuardPoints

All clients must have their GuardPoints disabled prior to **restoring** the backup on the NAS server. GuardPoints is required because LDT over NFS clients are unaware of changes being performed to the files, and LDT metadata for the files, on the NAS server.

Upgrading CTE agent in an LDT Communication Group

Upgrading from 7.2.0. to 7.3.0

- [Upgrading CTE Agents from 7.2.0 to 7.3.0](#)

Upgrading to 7.4.0

You can [Upgrade CTE](#) from previous versions to 7.4.0 with no special considerations.

Warning

For CTE Agents using LDT over CFS/NFS, you **MUST** upgrade to v7.4.0 **BEFORE** you can upgrade to 7.5.0. You cannot upgrade to 7.5.0 from versions previous to 7.4.0.

Upgrading to 7.5.0 and Post 7.5.0

Note

CipherTrust Transparent Encryption 7.5.0, and subsequent versions, have architectural design changes which require all nodes, Windows and Linux, to run CTE 7.5.0 in order for the LDT Communication Group nodes to communicate properly. The architecture design changes introduces [New Terminologies](#). It is critical to understand these new terms.

- [Upgrading CTE Agents in an LDT Communication Group from 7.4.0 to 7.5.0 and post 7.5.0](#)

Upgrading the CTE-LDT Agents in an LDT Communication Group from 7.2.0 to 7.3.0

Note

This is applicable for Linux **and** Windows.

When upgrading from CTE 7.2.0, all nodes in an LDT Communication Group **must be** upgraded together to 7.3.0. Rolling upgrades **do not work** because CTE 7.3.0 introduces security improvements and authentication features that involve the LDT Communication Group. All nodes must contain CTE 7.3.0 in order to be able to communicate.

For example, if an LDT Communication Group contains 10 nodes, then, with 7.2.0, you could upgrade them individually and still expect secfsd communication services to operate as expected. Note that the 10 nodes in the LDT Communication Group may or may not guard the same CIFS/NFS share, but they use the same LDT Communication Group master for communicating. With 7.2.0, if node1 were to be upgraded from 7.2.0.xxx to 7.2.0.yyy, then this does not affect the progress of LDT. Also, node1 is able to use all services correctly because it is admitted to the group by the upgraded communication master. So in this case, a rolling upgrade is possible.

With 7.3.0, Thales is introducing authentication for the LDT communication group nodes. Therefore, only nodes that have the capability to authenticate themselves can be part of the LDT Communication Group. As a result, for an upgrade operation, the entire set of 10 nodes in the LDT Communication Group have to be scheduled for maintenance at the same time. In the maintenance window, you must upgrade all nodes to 7.3.0. After a successful upgrade, the administrator should verify that all GuardPoints are green and working as expected. Any node with 7.2.0 will not be able to authenticate to the LDT Communication Group and, therefore, will not be admitted into the LDT Communication Group.

Procedure

1. Start a maintenance window by removing all nodes, in the LDT Communication Group, from production.
2. Upgrade all nodes to 7.3.0.
3. Verify that all GuardPoints are guarded on all nodes as expected, type:

```
voradmin ldt group check <gp path>
```

4. Verify that the LCG communications master is up and running, in **Windows**, by typing: `voradmin ldt group comm_info`

Note

Linux is adding support for this command in a subsequent version.

5. When finished upgrading, add the nodes back into production.

New Terminology

The architecture design changes introduces new terminologies. It is critical to understand these new terms.

New LDT Communication Group Terms

- **Triad nodes**

The triad nodes are the first three servers in the LDT Communication Group.

- **Subordinate nodes**

The subordinate nodes are the remaining nodes in the LDT Communication Group.

GuardPoint Terms

- **Primary node**

The node that performs LDT rekeying of data.

- **Secondary node**

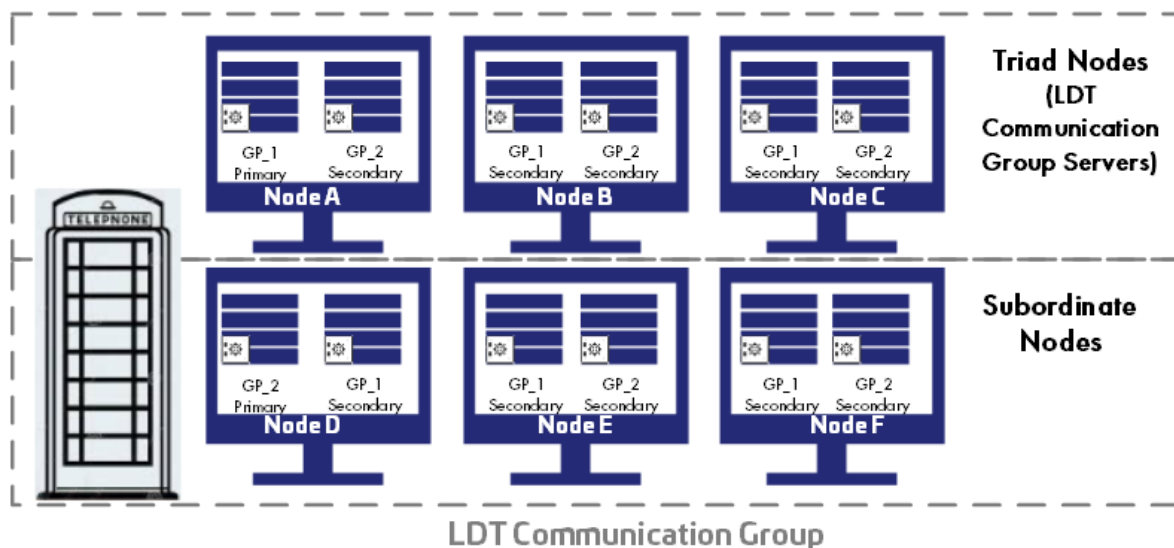
All of the other clients in the LDT GuardPoint Group are assigned as secondary nodes.

Note

They do not participate in rekeying data. Secondary nodes can be promoted to primary nodes if a primary node fails.

- **LDT GuardPoint Group**

An LDT GuardPoint Group contains a collection of LDT-enabled CTE clients that are all on the same network, managed by one CipherTrust Manager, guarding a common NFS/CIFS share and can communicate with each other over that network. An LDT GuardPoint Group enables the primary host of an LDT GuardPoint Group to communicate with the secondary hosts that are members of that LDT GuardPoint Group.



Upgrading CTE agents in an \$ {lcg} from 7.4.0 to 7.5.0 and post 7.5.0

When upgrading an LDT Communication Group, you must stop CTE on all of the nodes to be upgraded before upgrading them.

1. Disable the GuardPoints on the nodes to be upgraded:
 - a. Go to the CipherTrust Manager UI.
 - b. In the GuardPoint window, click the ellipsis on the right side of a GuardPoint and select **disable** to disable the GuardPoint.
 - c. Repeat the steps for all of the GuardPoints on the nodes to be upgraded.
2. Stop the CTE service on all of the nodes to ensure that all of the GuardPoints are unguarded.

For **Windows**:

- a. Go to **Control Panel > Services (local)**.
- b. Select **secfsd**.
- c. Select **Stop the Service**.

For **Linux**, type:

```
/etc/vormetric/secfs stop
```

3. **Upgrade CTE** on all of the nodes.
4. If your setup contains **manual directory** type GuardPoints, then you must run `secfsd -guard <gp>` to guard the GuardPoints again after the upgrade.
5. Verify that the GuardPoints are guarded on all of the nodes. Type:

```
# secfsd -status guard
```

Quality of Service

You can perform QoS administrative functions on the current primary CTE host. If the same CTE host is acting as a primary and secondary for multiple GuardPoints, the LDT schedule and/or Pause/Resume administrative options target those GuardPoints for which the host has assumed the primary role. This restriction is enforced through QoS functions available on CipherTrust Manager and the voradmin command.

Note

- The current pause/resume state of LDT at the host/client level, is contained in the `/var/vormetric/vorm-prs` file. This file is not protected. If it is accidentally removed or changed, it will be re-created and updated to reflect the correct pause/resume status within 30 seconds or less.
- Be sure to pause or resume LDT on the primary host only.

Suspending LDT from the CTE Agent CLI and CM GUI

Promotion of a secondary host, to the primary role, can alter the current pause/resume state of the GuardPoint on the newly promoted host. The new state of the GuardPoint depends on the pause/resume state of the GuardPoint on the primary host prior to promotion. When a GuardPoint is suspended after promotion to primary, run the voradmin command on the GuardPoint to resume rekey. The following table describes the states:

Suspension From	Current Primary - GP Suspended	GP Conditions after Promotion to Primary	New Primary - GP Suspended
CLI	Suspended	N/A	Suspended
CLI	Active	The new primary host is also the primary host for other GuardPoints. Rekey has been suspended on those GuardPoints.	Suspended
CLI	Suspended	The new primary host is also the primary host for other GuardPoints.	Suspended

Suspension From	Current Primary - GP Suspended	Conditions after Promotion to Primary	New Primary - GP Suspended
		Rekey has been suspended on those GuardPoints.	
CM GUI	Suspended	Rekey is not suspended on the new primary host.	Active
CM GUI	Active	Rekey is suspended on the new primary host. Within n seconds after promotion (n < 30 seconds. QOS monitoring interval) the new primary host will suspend rekey on the GuardPoint.	Suspended
CM GUI	Suspended	Rekey is suspended on the new primary host. Within n seconds after promotion (n < 30 seconds. QOS monitoring interval) the new primary host will suspend rekey on the GuardPoint.	Suspended

Alerts

This section lists the alerts for LDT over NFS and describes what to do in each case. LDT generates alerts when issues arise that require attention from the Administrator. Without prompt attention, alerts can delay the rekey process or cause the process to remain incomplete.

Linux Alerts

Guard/Unguard alerts

Label	Alert	Description
I_LDT_NFS_PTS_ACCES S_FAILED	Failed to preserve time stamps: GuardPoint [GuardPoint] objID [objID] error [error]	Primary host sends this alert to notify of the failed attempt to preserve time stamps on the specified inode number in the GuardPoint. The

Label	Alert	Description
		reason for the failure is reported using the error code.
E_LDT_NFS_REKEY_OP_FAILED	Failed to start or stop rekey on GuardPoint [%s] over NAS share, error [error]	Primary host sends this alert if it fails to start or stop rekey on a specified GuardPoint. The error code in the alert provides the error code associated with the failed attempt.

LDT Communications Group alerts

Label	Alert	Description
W_LDT_COMMGRP_MASTER_DOWN	LDT Communication Master is down	Each member of an LDT GuardPoint Group sends this warning reporting that the LDT Communication Group Master associated with the group is down.
I_LDT_COMMGRP_MASTER_UP	LDT Communication Master is up	Each member of LDT GuardPoint Group sends this notice reporting that the LDT Communication Group Master associated with the group is no longer down.
W_LDTGRP_PRIMARY_LEFT	LDT Group Primary host left the group for GuardPoint [%s]	This warning reports that the specified GuardPoint has been disabled on the host designated as the primary host for the GuardPoint. LDT immediately designates another member in the LDT GuardPoint Group as the primary host for the group.
I_LDTGRP_PRIMARY_ELECTED	LDT Group Primary host has been elected for GuardPoint [GuardPoint]	Every member of an LDT GuardPoint Group for a specific GuardPoint sends this notice to report the election of a member of the group as the primary host for the GuardPoint.
I_LDTGRP_PRIMARY_PROMOTED		

Label	Alert	Description
	Host has been promoted to LDT Group Primary role for GuardPoint [GuardPoint]	The host promoted to the primary role for the GuardPoint sends this notice.
W_LDTGRP_PRIMARY_RE FUSED	Host cannot be promoted to LDT Group Primary role for GuardPoint [GuardPoint]	This warning reports that the host sending the alert was elected for promotion to the primary role for the GuardPoint, and the host rejected the request. Read-only access to a GuardPoint or the host not having the latest policy key version are common reasons for a host to reject the promotion.
E_LDTGRP_PRIMARY_EF AILED	Election of this host for LDT Group Primary role has failed for GuardPoint [GuardPoint]	The host elected and accepted the primary role for the GuardPoint sends this error message if the host fails to perform the operations required for promotion to the primary role for the GuardPoint.
E_LDTNFS_FOVER_FOP	File operation recovery failed on new primary host after primary election, GuardPoint [%s] objID [%llu] error [%d] source [%d]	This error message reports that the host promoted to primary status was unable to complete the file operation that was initiated prior to the unexpected failure and departure of the previous primary host from the group. The file inode number is the objID in the message. The error code indicates the reason for the failure. The source identifies the operation that the new primary host failed to complete.
E_LDTNFS_RECOVERY_F AILED	LDT recovery failed on new primary host after primary election, GuardPoint [%s] objID [%llu] error [%d]	This error message reports that the host promoted to primary status was unable to recover incomplete rekey operation on the specified file inode number in the specified GuardPoint after unexpected failure and departure of the previous primary

Label	Alert	Description
		host from the group. The error code indicates the reason for the failure. The error code indicates the reason for the failure.
W_LDTNFS_PROMOTION_TOOLONG	Promotion too long	Promotion process taking too long, causing a delay in file operations, GuardPoint [%s] objID [%llu]. This warning occurs when a request to the primary host is delayed for a long time because a new primary has not been promoted.
E_LDTNFS_FOP_FAILED	File operation failed on secondary host, GuardPoint [%s] objID [%llu] error [%d] source [%d]	This alert from the primary host reports primary host failure to update LDT related metadata on behalf of a member performing a file operation that requires making changes to LDT metadata. File operations performed on secondary hosts to files undergoing rekey, such as truncate or rename, involves the primary host to intervene for updating LDT metadata.
W_LDTNFS_KEY_ROTATE_ENOKEY	Some members of LDT GuardPoint Group for [%s] have not received latest key version	This warning is initiated by the primary host when at least one member of the group for the specified GuardPoint Group has not received the latest policy key version that the primary host is committing. Make sure that all member of the LDT GuardPoint Group are connected to the CM.
W_LDTNFS_LAZY_RK_FAILURE	Lazy rekey operation failed to rekey <file name>, error <error code>	This alert reports failure to rekey the specified file name. The error code provides the reason for the error.

Windows Alerts

Guard/Unguard alerts

Label	Alert	Description
E_LDT_GUARD_FAILED	Failed to enable/apply GuardPoint	Guard operation failed
E_LDT_UNGUARD_FAILED	Failed to disable/delete GuardPoint	Unguard operation failed

Credentials alert

Label	Alert	Description
E_LDT_CIFS_INVALID_CRED	Failed to enumerate GuardPoint <.. \GP0>, Verify the provided CIFS credentials.	CIFS credentials may be expired or may not have permission to access.
E_LDT_CIFS_LOGIN_FAILED	Failed to authenticate a user	Failed to authenticate user using given credential.
[CGA] [ERROR] [4] [CGA3142E]	Not guarding path; Reason: Resource not available	If the agent logs display this message during CIFS GuardPoint creation, you can safely disregard. The GuardPoint will be created after the credentials are authorized. It may take a few minutes.

LDT Communication Group alerts

Label	Alert	Description
E_LDT_LGS_JOINING_FAILED	Failed to join GuardPoint group	CTE client was unable to join the LDTGuardPoint group
E_LDT_LGS_LEAVING_FAILED	Failed to leave GuardPoint group	CTE client was unable to leave the LDT GuardPoint group

Troubleshooting

The majority of the CTE clients must be fully operational for proper LDT operations across all LDT GuardPoint Groups.

If 50% or more CTE clients fail, LDT Communication Group cannot recover from that situation. To fix this issue, you must reboot all of the CTE clients in the entire LDT Communication Group.

The LDT Communication Group requires that the majority of CTE clients are active in order to work properly. If 50% or more CTE clients fail, LDT Communication Group cannot recover from that situation. To fix this issue, you must either:

- Reboot all of the CTE clients in the entire LDT Communication Group.
- In Windows, restart `secfsd` through the **Control Panel > Services** page all of the CTE clients in the entire LDT Communication Group.

Troubleshooting CTE-LDT

- [Monitoring and Statistics](#)
- [Protecting CTE-LDT GuardPoints against Failure in Underlying File Systems \(Linux\)](#)
- [Alerts Playbook](#)
- [Error Messages](#)
- [Warning and Info Messages](#)
- [Recommendations and Considerations](#)

Support Contacts

If you encounter a problem while installing, registering, or operating the product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or [Thales Customer Support](#).

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at [Thales Customer Support](#), is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

Tip

You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the REGISTER link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support@Thales.com.