April 2020

Encrypting IBM Spectrum Virtualize and IBM FlashSystem storage

Overview

Challenge

Encrypt production database without a business outage.

Solution

Move production databases to a staging storage space using VDisk mirroring, and then enable encryption in SAN Volume Controller backend FlashSystem storages.

Encryption is the process of encoding data so that only authorized parties can read it. It improves the physical security of the data and is a requirement for a lot of customers. Encryption of data can be done *at rest* or *in flight*. Data encrypted *at rest* means that the data is encrypted on the end device (disk drives).

IBM® Spectrum® Virtualize systems support hardware encryption for data *at rest* with the IBM Storwize® V7000 Gen2 or Storwize V7000 Gen2+ versions, Storwize 5020, Storwize V5020 (internal only), Storwize V5030 (internal and external), IBM SAN Volume Controller 2145-DH8 or SAN Volume Controller 2145-SV1, and IBM FlashSystem V9000 support encrypting data on internal drives. Software encryption for data *at rest* is also available to encrypt data on external storage controllers with no encryption capability.

This white paper provides a quick configuration guide for enabling, configuring, and using the encryption capabilities for IBM Storwize family systems.

Architecture

Software

- Spectrum Virtualize 8.1.3.5
- IBM FlashSystem 900 1.5.1.2

Hardware

- SAN Volume Controller 2145-DH8 (eight-node cluster)
- IBM FlashSystem 900 AE3 storage

Enabling encryption

To use the encryption capabilities, customers must purchase an encryption license, thereafter, activate the license on the system, and enable encryption. After the storage servers are enabled to use encryption, the subsequent data stored post enabling encryption will be encrypted.

Note: If the storage servers possess non encrypted pools, IBM does provide an additional capability of migrating the data from non-encrypted pools to encrypted pools, after the server has been equipped with encryption.

Enabling the encryption license

The management GUI offers two ways to activate an encryption license.

- Automatic
- Manual

Automatic

Activating the license keys through the automated way would require the notebook being used to be connected to an external network. If you already have a valid license key, click **Settings → Systems → Licensed Functions → Encryption Licenses** and perform the following steps:

1. On the Encryption page, click **Yes** to reflect that you have already purchase a valid license key.
2. Select the control enclosure on which encryption must be enabled. Then click **Actions → Activate License Automatically**.
3. Enter the authorization code that you have received with the licensed function documents and select **Activate**.

Manual

For the manual approach, you need to perform the following steps:

1. On the Encryption page, click **Yes** to reflect that you have already purchase a valid license key.
2. Select the control enclosure on which encryption must be enabled. Then click **Actions → Activate License Manually**.
3. On the Activate License Manually page, retrieve the license keys by completing the following steps:
   a. Go to https://www.ibm.com/storage/dsfa
   b. Select your product type.
   c. Enter the following information:
      - Machine type and model
      - Serial number
      - Machine signature

4. Enter the authorization codes that were sent with your purchase agreement for the licensed function.
5. Copy or download the key.
6. Enter the license key in the space provided.
7. Click **Activate**.

You can also activate the license by performing the following steps using the command-line interface (CLI):

1. Run the `activatefeature -licensekey` *`key`* command in the CLI, where key is the license key to activate a feature. The key consists of a 16-digit hexadecimal character.
2. Run the following command to activate the license with a file path that stores the key:
   `activatefeature -licensekeyfile` *`filepath`*

Methods to configure encryption on IBM SVC and Storwize products

There are two ways to configure encryption. You can either use the centralized key server that simplifies creation and management of the encryption keys or use USB flash drives for storing the encryption keys.

Note: The organizations that follow strict security policies regarding USB flash drives use key servers to manage encryption.

In this implementation of SVC and Storwize products, the test team chose USB flash drives to store the encryption keys and the same is discussed in detail in this paper. Refer to IBM Knowledge Center for details about the IBM Security Key Lifecycle Manager approach.

Using the USB flash drives method provides the following benefits:

- Inexpensive to maintain and use
- No mechanical components required to maintain the read/write operations to the flash drives
- Physical access would be required for rekeying operations
- Convenient to have multiple flash drives for backup operations

Configuring USB encryption

The `lsencryption` command should ensure that the encryption status is set to licensed.

Note: In the SVC GUI workflow for configuring encryption, you are prompted to insert the required number of USB flash drives into the system. When the system detects the USB flash drives, the encryption key is automatically copied to the USB flash drives. It is recommended to have at least one USB flash drive per node.

USB encryption can be configured using either the GUI or the CLI.

Configuring USB encryption using the GUI

Perform the following steps to configure USB encryption using the GUI:

1. Click **Settings → Security → Encryption**.

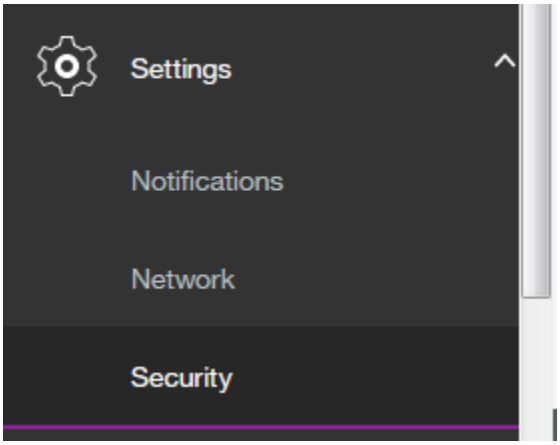

Figure 1. Enabling encryption

Note: IBM FlashSystem 900 does not need encryption license key activation, and it is a trust-based license.

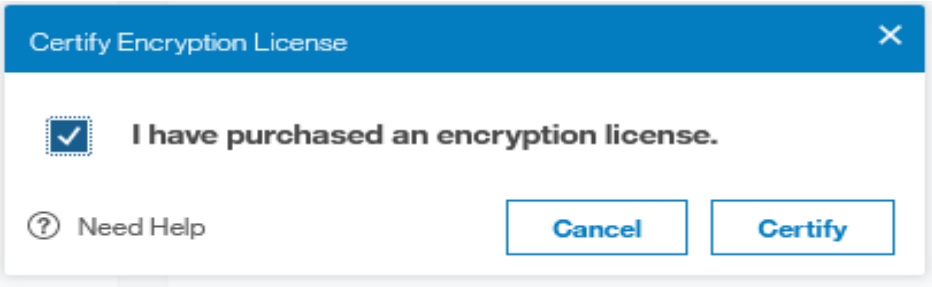

Figure 2. Encryption license purchase acknowledgment

2. Click **Enable Encryption**.
3. Select **USB flash drives**.

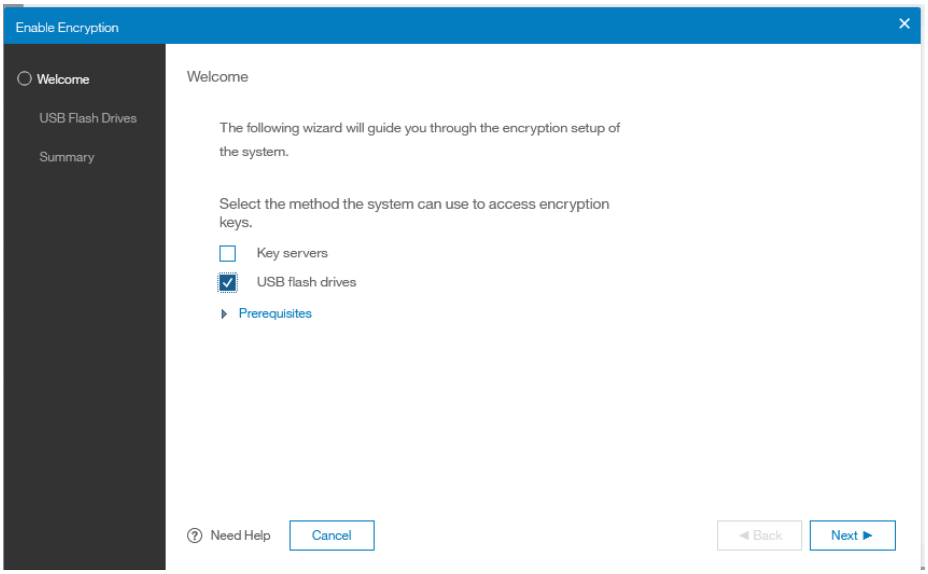

Figure 3. Select USB flash drives

4. In the wizard, you are prompted to insert the required number of USB flash drives into the system. When the system detects the USB flash drives, the encryption key is automatically copied to the USB flash drives. Ensure that you create any required extra copies for backup. You can leave the USB flash drives inserted into the system. However, the area where the system is located must be secure to prevent the USB flash drives from being lost or stolen. If the area where the system is located is not secure, remove all the USB flash drives from the system and store securely.

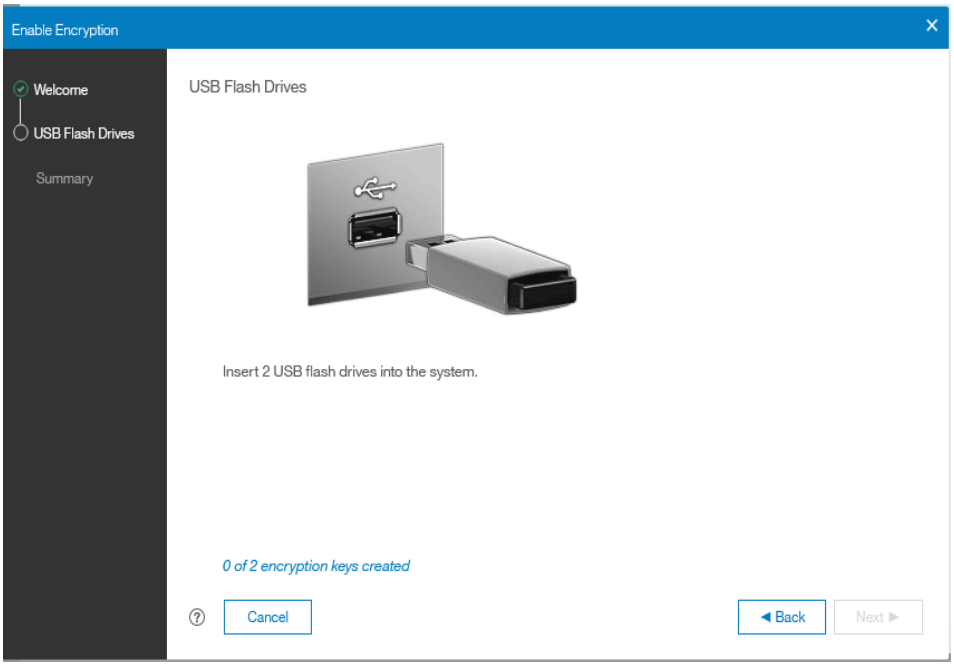

Figure 3. Insert the USB flash drives

5. After configuring the USB flash drives, the encryption keys get copied to the drives. Let it complete and then click **Finish**.

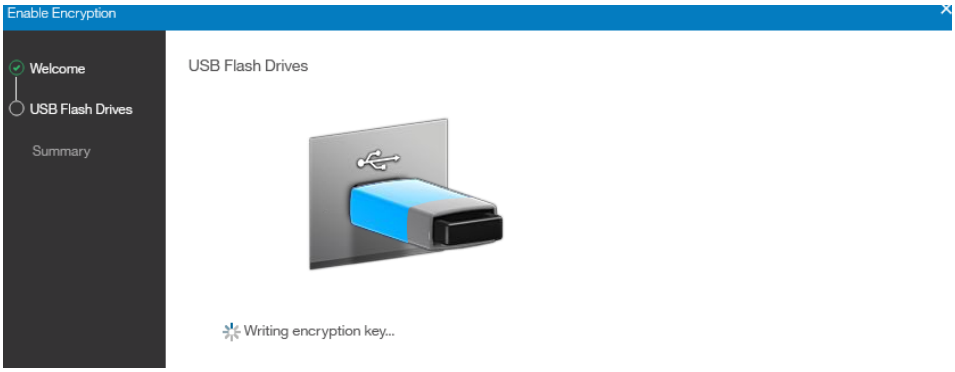

Figure 5. Writing encryption key

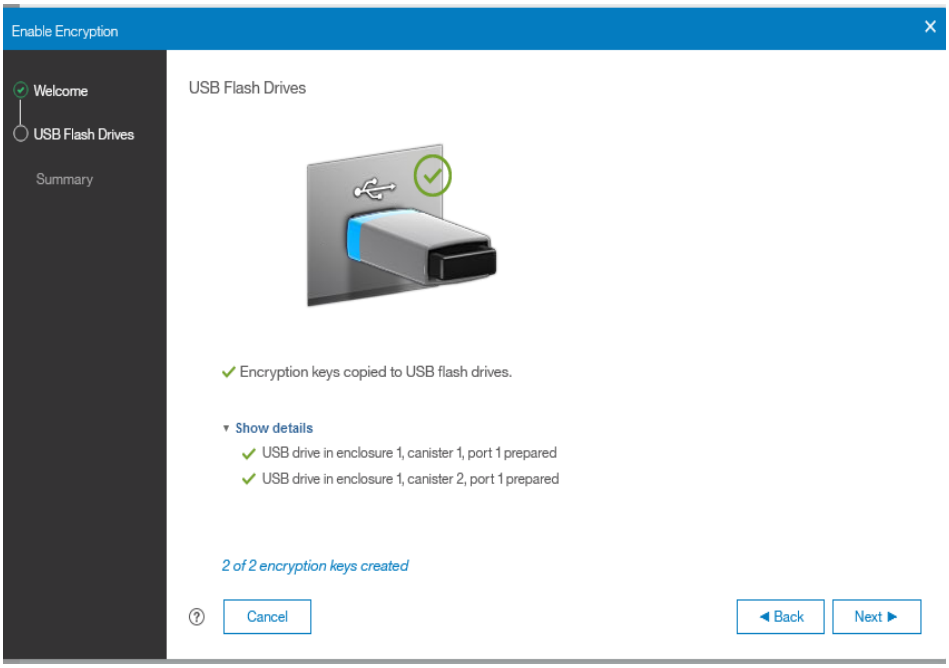

Figure 6. Encryption keys copied to the USB flash drives

6. Click **Finish** to complete the configuration and click **Close** in the subsequent dialog boxes that are displayed.

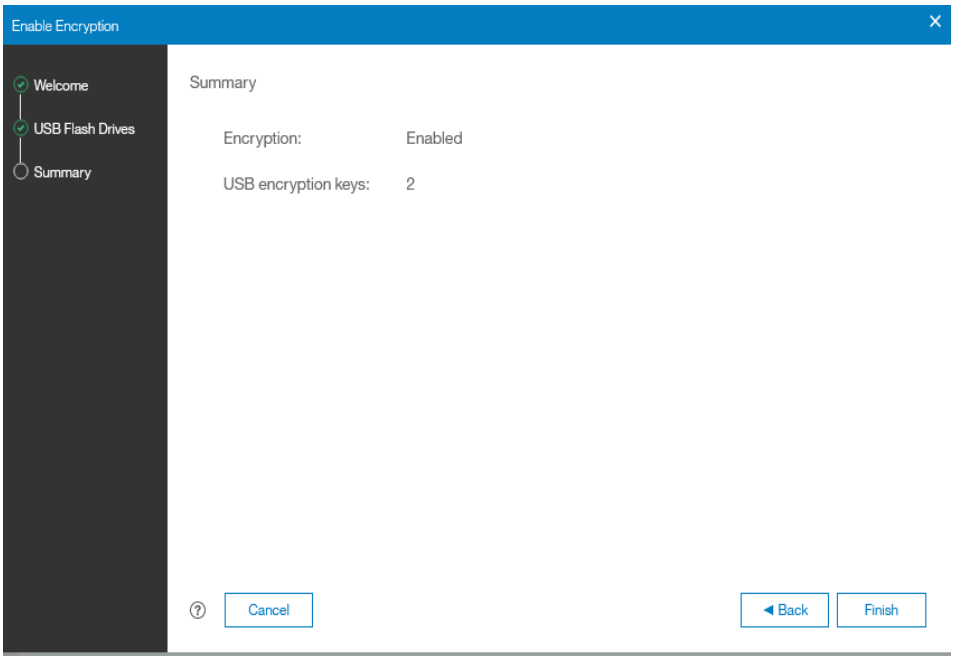

Figure 7. Configuration summary

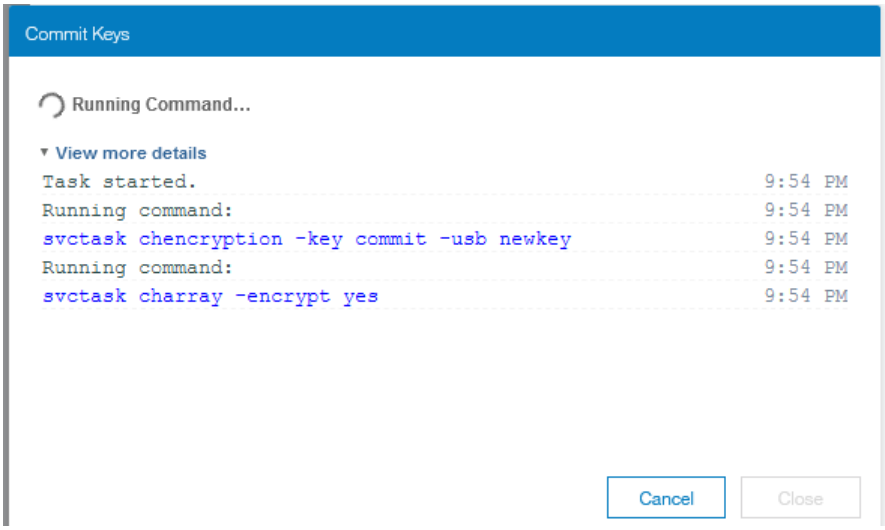

Figure 8. Committing the prepared key as the current key

Note: Plan to create several backup copies either on the flash drives or any other storage media and secure it.

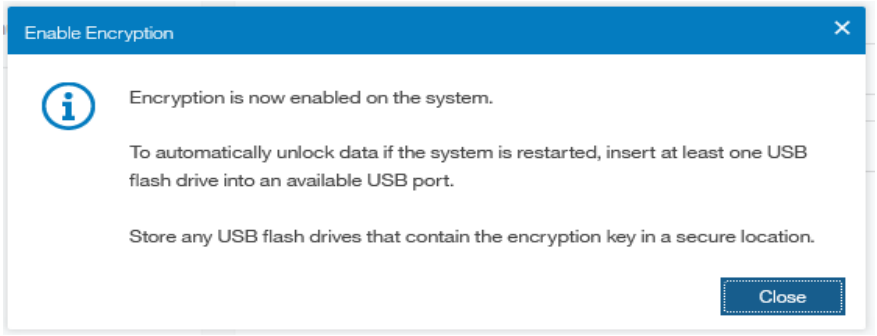

Figure 9. Encryption on the storage server is ready to use

7. After encryption is configured, validate it by clicking **Settings** → **Security** → **Encryption**.

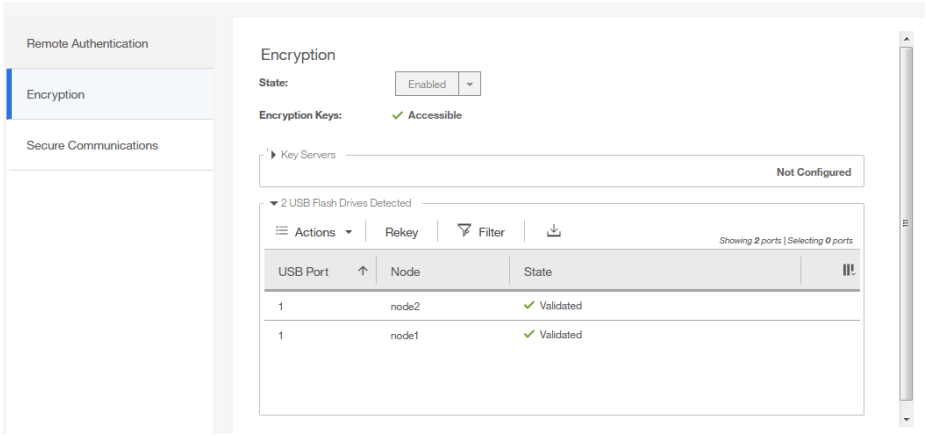

Figure 10. Validate the configuration by clicking the Encryption tab

Configuring USB encryption using the CLI

Perform the following steps to configure USB encryption using the CLI:

1. Enter the following command to enable encryption:
   `chencryption -usb enable`

2. Run `lsportusb` and validate that enough drives are connected to the enclosure and the status is **Active**.

3. Run the following command to create system encryption keys and write those to all attached drives:
   `chencryption -usb newkey -key prepare`

4. Commit the prepared key. Ensure that CLI `lsencryption` reflects the value for `usb_rekey` as `set to prepared` and the number of encryption keys is greater than the minimum number required.
   `chencryption -usb newkey -key commit`

Use case: Encrypting SVC's back-end FlashSystem 900 storages

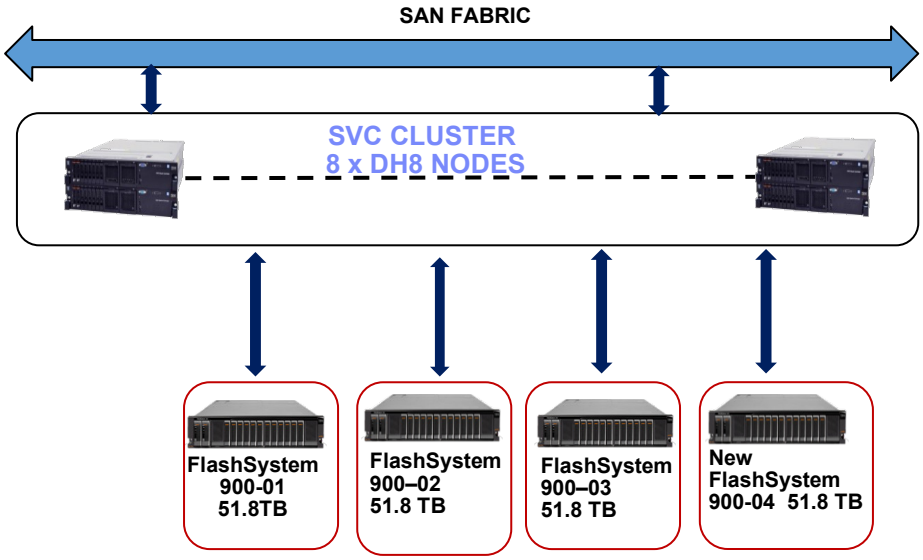

Figure 11. Figure reflecting a use case where a San Volume controller is virtualizes encrypted flash storage servers

The eight node SVC-DH8 cluster architecture hosts all production workloads. IBM FlashSystem 900 storage controllers are virtualized under the IBM SVC-DH8 cluster.

Each flash storage forms a storage pool in SVC that holds mission-critical databases. As part of new regulatory norms, the client needs to ensure that all mission-critical databases are encrypted.

The solution provided encrypts the underlying flash storage controllers using USB keys so that the existing flash storage can be re-initialized with USB keys and the managed disks (MDisks) can be re-created.

In the Figure 11, the additional IBM FlashSystem 900 storage controller (New FlashSystem 900-04) has provided a solution to upgrade the SVC capacity as well as to be used as a staging area to encrypt the existing FlashSystem storage controllers.

The current database volumes migrated from the existing FlashSystem storage pools to the new FlashSystem 900 pool in SVC. Hence the existing flash storages are available to reinitialize and are ready for encryption.

After the flash storage boxes are encrypted, the volumes are copied back to its original pools. This exercise is repeated for all the flash storage boxes in the solution. Because the SVC pool migration was entirely online and a host unaware activity, there were no business impacts.

In this use case, all underlying FlashSystem storage controllers of SVC are required to be encrypted. A typical workflow is shown in the flow chart with an example of a FlashSystem 900 storage system (FS900-01). This approach is repeated for each FlashSystem box encryption enablement.

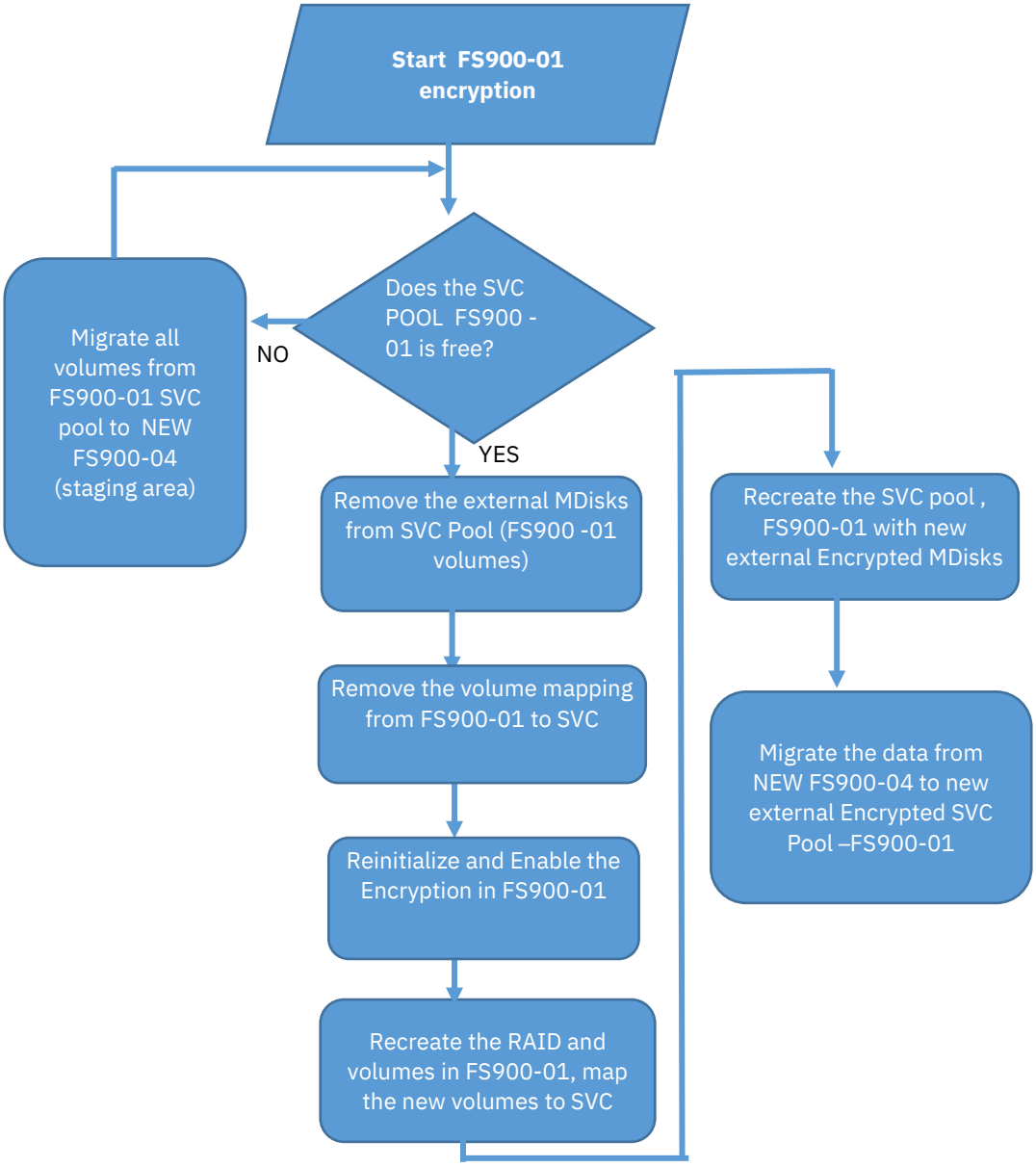

Figure 12. A typical workflow of a sample FlashSystem 900 storage system

Get more information

The following websites provide useful references to supplement the information contained in this paper

IBM Systems on PartnerWorld®
ibm.com/partnerworld/systems

IBM Power Development Platform
ibm.com/partnerworld/pdp

IBM Power Systems Information Center
http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp

IBM Redbooks®
ibm.com/redbooks

About the author

Sumit Mehrotra is a software engineer and Stress Verification Test Lead in the IBM Systems and Storwize team. You can reach Sumit at sumimehr@in.ibm.com

Anuj Chandra is as senior developer and lead in the IBM Systems SVC and Storwize team. You can reach Anuj at anujchan@in.ibm.com.

Akbar Harees is a storage consultant with IBM Systems Lab Services and Training team. You can reach Akbar at akharees@in.ibm.com

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America