EMC DATA-AT-REST ENCRYPTION FOR VNXe: SELF-ENCRYPTING DRIVES
Detailed Overview

Abstract
This white paper introduces EMC® Data-at-Rest Encryption™ for VNXe™, a feature that provides data theft prevention when a drive is stolen or misplaced. This paper provides a detailed description of how to configure, modify, and manage self-encrypting drives on a VNXe storage system.

January 2013
Table of Contents

Executive summary ................................................................. 4
Introduction .................................................................................. 4
  Audience ................................................................................ 5
  Terminology ............................................................................. 5
Concepts ......................................................................................... 6
Managing the Authentication Key .................................................. 9
  Creating the Authentication Key .............................................. 9
  Backing Up the Authentication Key .......................................... 13
  Restoring the Authentication Key ............................................ 15
  Clearing the Authentication Key .............................................. 15
Conclusion ..................................................................................... 17
References .................................................................................... 17
Executive summary
In today’s world of storage, the amount of sensitive data produced daily is growing exponentially, and one of the biggest challenges is the security of this data. To address this issue, the EMC® VNXe™ Series provides Data-at-Rest Encryption, a technology that encrypts idle data stored on a device to protect it from the loss or theft of a drive.

Data-at-Rest Encryption occurs when data stored on a device is encrypted to protect against unauthorized access. All data written to the storage device is encrypted before it is stored, and all data read from the storage device is decrypted as it is read. The VNXe Series achieves this level of security at the hardware level using self-encrypting drives (SEDs).

According to the Advance Encryption Standard (AES), Data-at-Rest Encryption provides customers with the benefit of protection against data theft, as well as the ability to quickly and securely repurpose storage. If a drive loses power, as occurs when it is removed from its owner’s system, access to user data on the drive will be denied when power is restored. After power is restored, the host array will require authentication to access any data stored on the drive. This access can be gained by using a bandmaster password, which is otherwise known as an Authentication Key (AK). Since the authentication key securely resides on both the self-encrypting drive and on the VNXe, it is not possible to access user data when the self-encrypting drive is inserted into another array. This ensures protection against data being accessed and compromised on lost or stolen drives.

Introduction
Imagine that a Storage Administrator manages multiple VNXe arrays in San Francisco and Boston. With the main office in San Francisco, it is difficult to keep a close eye on the drives located across the country. In the event that drives are stolen, the security of the data on them is a major concern. This white paper discusses how the data in this and many other scenarios can easily be protected.

This white paper introduces Data-at-Rest Encryption for the VNXe Series using self-encrypting drives. It explains how the self-encrypting drives function, how the data on the drives become secure, and how the VNXe Series manages each drive.

This paper also describes the implementation of this feature in Unisphere for VNXe and shows how a user can easily manage the authentication key. Finally, it provides instructions for locking/unlocking a self-encrypting drive and clearing an authentication key that was created on a secure array.
Audience
This white paper is intended for EMC customers, partners, and employees who are concerned about data security. It assumes that the reader has general IT experience, including knowledge as a system or network administrator.

Terminology

- **Authentication Key (BandMaster Password)** - This password controls access to the data stored in a locked band. The only way to read/write the data in a band is to know the associated BandMaster password.

- **Bands** – The starting point and length of the Logical Block Addressing (LBA). The LBA ranges on a self-encrypting drive are divided into these bands.

- **Cryptographic Erase** - The act of making data unrecoverable by deleting (or forgetting) the key used to encrypt the information. Cryptographic erase securely erases information without rewriting the data.

- **Data-at-Rest Encryption** – The process of encrypting idle data on a storage device and protecting it against unauthorized access unless valid credentials are provided.

- **Data Encryption Key (DEK)** - A randomly generated 32-byte number that is used by the encryption engine to generate a unique cipher text. This key is internally managed by the drive and is encrypted by the appropriate password or secret that is controlled by the array's Storage Processor.

- **Encryption/Decryption Engine** – Trusted electronics within the self-encrypting drive that use a secret key to convert passing data from clear text to encrypted text and vice versa, without any noticeable delay.

- **Secure Array** – A storage system that accepts only self-encrypting drives and uses Data-at-Rest Encryption to protect data when physical control of a drive is lost.

- **Self-Encrypting Drive (SED)** – A drive that encrypts all data before it is written to the storage medium, and decrypts the same data before it is read.

- **Storage Pool** – A collection of disk drives configured with a particular storage profile. The storage profile defines the type of disks used to provide storage, as well as the type of RAID configured on the disks. The storage pool’s configuration defines the number of disks and quantity of storage associated with the pool.

- **Storage Processor (SP)** – A hardware component that performs VNXe storage operations such as creating, managing, and monitoring storage resources.

- **Unisphere™** – The management interface for creating, managing, and monitoring VNXe storage resources and protection.
Concepts
Today, it is common for organizations to constantly interact with important and sensitive data. A main concern of IT administrators is the security of the drives and the data that resides on them. Data-at-Rest Encryption via self-encrypting drives is available to ensure that data will remain protected when physical control of the drive is lost.

Data-at-Rest Encryption is available on both VNXe3150 and VNXe3300 secure arrays with dual Storage Processors. As soon as the system is unpackaged and powered on, it is ready to start protecting data with no further configuration needed. Encryption is the process of transforming data to make it unreadable to anyone except those possessing specialized knowledge, referred to as the authentication key. A self-encrypting drive encrypts data with the embedded drive electronics before it is stored. This protects the data on the drive against theft and unauthorized attempts to read the data directly by physically deconstructing the drive using methods such as a drive recovery service. Self-encrypted drives are supported in secure arrays only. All drives in the secure array must be self-encrypting. A non-secure array cannot be converted to a secure array and vice versa.

A self-encrypting drive is a hard drive with built-in, trusted electronics that encrypt and decrypt all data to and from the magnetic media on the fly (Figure 1). The drive performs like any other hard drive, and the encryption is transparent to the user. Because this functionality is handled at the drive level, overall system performance is not affected. The trusted drive electronics use an encryption/decryption engine that uses a secret key to convert all passing data from clear text to encrypted text, and vice versa, without any noticeable delay.

![Trusted Drive Electronics](image)

**Figure 1. Trusted Drive Electronics**

The authentication key used by the encryption/decryption engine functions as a password to gain access to the media, both unlocking an associated data band and providing the means to encrypt/decrypt the Data Encryption Key (DEK) stored on the drive. Therefore, a self-encrypting drive really uses two keys to access user data. The
The authentication key functions as a drive’s ownership credentials, proving to the drive that it was inserted into its owner’s VNXe array. The authentication key also grants the VNXe access to the data by decrypting the DEK. The DEK encrypts/decrypts all I/O to and from the drive. It is internally managed by the trusted drive electronics. When a self-encrypting drive is locked, the same authentication key from the owner VNXe is stored on the drive as a hashed authentication key. The hashing is done using the SHA256 algorithm. This process is shown in Figure 2.

**Figure 2. Self-Encrypting Drive**

As seen in Figure 2, the VNXe is responsible for sending both the user data and the authentication key to the self-encrypting drive. It is then up to the trusted drive electronics, including the encryption/decryption engine, to provide access to the storage media by using the hashed authentication key and encrypted DEK stored on the drive. The details of the trusted drive electronics are discussed later in this paper.

The authentication key, also known as the bandmaster password, is created when the first storage pool (user RAID group) is created on the array. The VNXe generates one key per array as soon as the storage is available to protect. This key is applied to every self-encrypting drive within the system. Although the authentication key applies to each drive in the secure array, it is stored only on drives in storage pools, as shown in Figure 3.

**Figure 3. Authentication Key**
As seen in Figure 3, the first six drives, highlighted in blue, are part of a storage pool. Since these drives are in a pool, they are automatically locked. Therefore, they store a hashed copy of the authentication key along with an encrypted DEK. This DEK remains permanently on the drives. The remaining drives in the array remain unlocked until they become part of a storage pool. At that point, they become locked and use the same authentication key that was used for the first pool.

When an authentication key is created on a secure array, not only is a hashed copy of the key stored on the drive, but five redundant copies are stored within the array. The authentication key is triple mirrored on the first three system drives (0,1,2) and securely backed up on the SSD of each Storage Processor. The master copy resides on the system drives. EMC highly recommends that the user back up the authentication key to an external drive as soon as the key is created. Unisphere allows users to easily perform this backup. Remember that if the master copy of the authentication key is missing or is corrupt, the data stored on the system will become inaccessible.

Figure 4 shows a four step break down of the process that is used to encrypt/decrypt data on a drive.

As seen in Figure 4, the self-encrypting drive is separated into two main parts; the trusted drive electronics and the storage media. When I/O is requested from the self-encrypting drive, the system verifies that the drive was inserted into its owner system. As mentioned before, the authentication key acts like a set of credentials that makes sure the drive is in the correct array.
Once the authentication key is sent to the self-encrypting drive, the trusted drive electronics compare the hashed copy of the key on the drive to the copy sent by the secure array. If the two keys do not match, an error is sent back to the array stating that the drive is locked and may have been inserted into the wrong array. If the two keys match, the encryption/decryption engine is granted access to transform the passing data. The engine retrieves the encrypted DEK that is stored on the drive and uses the same authentication key that was sent by the secure array to decrypt the DEK to readable text. The decrypted DEK is loaded into the engine, where it transforms the data. Then, the drive responds to read/write commands from the host. This entire process is transparent to the user and has zero or minimal performance impact on active data.

When a storage pool is first created, the same authentication key is used for all drives in the pool. In order to generate a new authentication key on a secure array, the existing key needs to be destroyed and recreated. A user has the ability to create and destroy the authentication key and to back it up from Unisphere. The following sections highlight these abilities and discuss how the VNXe Series manages self-encrypting drives. For more details on the functionality of all features available in Unisphere, please refer to *EMC Unisphere for VNXe – Next Generation Storage Management* white paper on EMC Online Support.

**Managing the Authentication Key**

**Creating the Authentication Key**
The authentication key is not present when a secure array is first initialized. When the first storage pool is created on the array, the self-encrypting drives request that an authentication key be generated. This authentication key can be used for any drive that is present within the array. Figure 5 shows the message that Unisphere generates when the first storage pool is created.

![Figure 5. Creating an Authentication Key](image)
The user has the ability to check the current status of the authentication key on the secure array at any time. In Unisphere, the Self-Encrypting Drive Key Management section on the Management Settings page displays this status. Figure 6 shows the message that appears if the authentication key was not yet generated.

**Figure 6. Authentication Key Not Generated**

Once the first storage pool has been created, the message in Figure 7 is displayed.

**Figure 7. Authentication Key Generated**

When the authentication key is generated, the system prompts the user to immediately back up the key in case it becomes corrupt or lost. This key is used to lock a drive. If a drive loses power, as would be the case if it were removed from the original system of which it is a part, its recorded data is locked against unauthorized access as soon as power is reapplied. With power reestablished, the host system needs to prove to the drive that it is the owner by providing the drive with the appropriate authentication key.

A drive becomes locked when the authentication key and DEK are present on it. Both keys are created on a drive once that drive is within a pool. Therefore any drive within an array that is not part of a storage pool contains neither the authentication key nor the DEK, and remains unlocked.

If a storage pool needs to be expanded, the drives added to the pool automatically become locked and use the same authentication key. This is the same case as if a faulted drive needs to be replaced. Simply remove the faulted drive and replace it with an unlocked self-encrypting drive. Once inserted, the secure array recognizes it as a new drive and resynchronizes it to obtain the user data and authentication key. The use of hot spares is discussed later.
If a user moves a locked drive to another pool within the same array, the drive is functional, but the system performs a resynchronization. If a locked drive is moved to another array, the other system immediately notifies the user that the newly inserted drive is locked and may have been placed in the wrong array. This alert is shown in Figure 8.

![Figure 8. Locked Drive Alert](image)

If a storage pool is deleted when a drive from the pool is missing, the missing drive will remain locked until it is reverted to factory default. Reverting a drive cryptographically erases the user data on the drive, which removes access to it. It also disables authentication. This is done by deleting the DEK used to decrypt the data. To revert a self-encrypting drive to its factory default, run the svc_key_restore service script. For information about the svc_key_restore service script, go to the EMC Online Support website, access the VNXe Product page, and search for VNXe Service Commands Technical Notes. For additional information and help, contact your service provider.
In a secure array, all drives must be self-encrypting drives, hence, selective encryption is not supported. If a non-self-encrypting drive is inserted into a secure array, the system generates the alert shown in Figure 9.

![Figure 9. Non-SED Alert](image)

Because the secure array does not support the drive, it is highlighted as faulted on the System Health page in Unisphere. Remove the drive and replace it with a self-encrypting drive. If a supported self-encrypting drive is inserted into the array, the array can operate normally, and the System Health page shows that the drive as a SED (Figure 10).

![Figure 10. Self-Encrypting Drive Component Description](image)

The behavior of a hot spare is similar to other drives configured on the system. When a drive is added to a hot spare pool, it belongs to a separate hot spare storage group. This operation triggers the drive to lock by creating the authentication key on the drive.
**Backing Up the Authentication Key**

A user can trigger a backup operation either through Unisphere or by using the command-line interface (CLI). It is strongly advised to back up the authentication key in order to restore it in the event of a corruption or loss. It is important to note that without the authentication key, all user data is lost.

Once the user initiates a backup operation, a user-supplied password is requested. This password protects the new instance of the authentication key that is created. If a restore is needed, the system uses this password to unlock and retrieve the backup of the key.

After the first storage pool is created, the system generates an alert stating that the authentication key was generated and should immediately be backed up to an external device. This alert bubble is shown in Figure 5. To back up the authentication key through Unisphere, access the Management Settings Page. The Back Up Authentication Key button is located in the Self-Encrypting Drive Key Management section on this page, as shown in Figure 11.

![Self-Encrypting Drive Key Management](image)

**Figure 11. Back Up Authentication Key Button**

When the Back Up Authentication Key button is clicked, Unisphere prompts the user for a password that will be used to protect the copy of the authentication key on an external device. This prompt is shown in Figure 12.
In the event that a user wants to back up the key to several external devices, the backup key can be generated as many times as needed, and the same password can be used each time. The password must meet the following criteria:

- Be at least 8 characters in length
- Include a symbol character
- Include a lower case letter
- Include an upper case letter

Once the password requirements are met, create the backup by clicking the Generate Backup Key button. After generating the backup key, click OK. This prompts the user to save the .key file on a local device. The .key file will only be recognized by this exact array and cannot be restored on a different array.

To back up the authentication key using VNXe Unisphere CLI, run the following command on the primary Storage Processor:

```
uemcli –d <address> -u <user_name> -p <password> -download –f SedKeyFile dek –password <password>
```

After this command is run, a SedKeyFile is generated on the host on which the command was run. This file will be used to restore the authentication key.
Restoring the Authentication Key
If the authentication key of an array becomes corrupt, or if restoring the key is the only recovery option during a failure scenario, the user can restore the key. The restoration of the authentication key is a process that can only be done by running a script as the service user on either Storage Processor via the CLI. Both Storage Processors must be put into Service Mode to execute this command. Place the backup of the key on a USB thumb drive and insert it into the back of one of the Storage Processors. Use the following command on the same Storage Processor to restore the authentication key:

```
service@spa spa:~> svc_key_restore --sed --restore <path/name>
```

In the above command, `<path/name>` refers to the path and name of the backup file that was created via the user interface or CLI. Once this command executes, the system displays a message stating that the restore was completed successfully as shown in Figure 13.

![Successful Restore Message](image)

**Figure 13. Successful Restore Message**

When the key is successfully restored, the system can be unlocked by booting back to its normal mode.

Clearing the Authentication Key
To clear the authentication key from the entire array, the user must remove every storage pool. Once the last pool is removed, the system clears the key and displays an alert stating that the key is no longer present, as shown in Figure 14.

---

1 To point to a USB pen drive, use `/mnt/pendrive/1/<file name>`
After the system displays the alert stating that the authentication key was cleared, the user can again see the status of the key by accessing the Management Settings page. This page notes that the key has not yet been generated and that a storage pool must be created to generate the key (Figure 6). Once the user creates another storage pool on the array, the authentication key is regenerated. This authentication key is different than the key that was previously on the array.

Since the authentication key is used to lock each drive in a storage pool, a drive becomes unlocked when it is removed from the pool. When a drive is unlocked, it can be moved freely from secure array to secure array. A self-encrypting drive that is unlocked functions normally in other secure arrays, but it is not supported in non-secure arrays. If an unlocked self-encrypting drive is placed into another secure array and the slot is part of an active storage pool, the drive resynchronizes with that pool and automatically obtains the new authentication key and DEK of the new array. This drive is now locked with the new keys and will be unlocked when it is removed from the new storage pool.

When moving a drive from a storage pool to another slot in the same array that is not in a storage pool, the authentication key remains on that drive. The key is only removed from the drive when the drive is removed from the storage pool. The drive will still function normally in the array. However, if the user tries to move that drive to another SED array, it is reported as locked, since the key is still present. To get rid of the key, the drive must either be added to a storage pool in its owner system and then removed from that pool (removing the authentication key) or reverted back to the factory default.
Conclusion
Data security is an important concept to keep in mind while designing an organization’s IT environment. Using Data-at-Rest Encryption ensures the protection of information when physical control of a drive is lost. Administrators located across the country can have a peace of mind knowing that their data is safe in the event that the drives are stolen.

This white paper described how self-encrypting drives function in a VNXe and how the authentication key and drive encryption key work together to encrypt/decrypt user data. To access user data from a self-encrypting drive, the correct key must be supplied by the VNXe. If not, access is denied. The authentication key is created when the first storage pool is created and destroyed when the last pool is removed. EMC highly recommends backing up the key in case of corruption. To provide extra security, a new key is generated each time the first storage pool is created. This protection is transparent to the user, and no performance impact is experienced.

References
The following can be found on EMC Online Support (https://support.emc.com) › VNXe Product Page:
- EMC VNXe Storage Systems – A Detailed Review
- EMC Unisphere for VNXe - Next Generation Storage Management – A Detailed Review