DATA CENTER

Encryption Solution Design and Deployment Considerations

This *Encryption Solution Design and Deployment Considerations* reference guide is designed to help customers and partners architect and deploy Brocade encryption solutions to maximize system performance, minimize administrative overhead, and mitigate the possibility of operational disruptions.

This guide was compiled with the help of a working group of subject matter experts from Brocade Headquarters and the Brocade field organization.

BROCADE

DEDICATION

This Encryption Solution Design and Deployment Considerations reference guide is dedicated to the memory of Peter Carucci—a wonderful person, father, and husband, who left us much too soon. Peter was an avid supporter of the Brocade encryption solutions and was instrumental in their success. He is sorely missed.

CONTENTS

INTRODUCTION

The Brocade Encryption Solution Design and Deployment Considerations reference guide is a result of extensive experience deploying Brocade® encryption solutions in a wide range of configurations and customer environments. This guide is designed to help customers and partners architect and design the Brocade encryption solutions to maximize system performance, minimize administrative overhead, and mitigate the possibility of operational disruptions.

This guide is designed for a wide range of audiences:

- SAN designers/architects who design and build encryption solutions
- Professional services consultants who deploy encryption solutions
- SAN/security managers who manage encryption solutions on a daily basis

Brocade has been offering data encryption functionality since it was first released in September 2008. With hundreds of deployments in various configurations and storage platforms, a wealth of experience has been accumulated since data encryption was introduced.

Document Scope

This document discusses a variety of business and technical considerations for designing, building, and managing the Brocade Storage Area Network (SAN)-based encryption solutions.

This document is not designed as a detailed how-to manual on deploying Brocade encryption solutions, but rather as a higher-level overview with some technical detail that will help ensure a successful implementation of these solutions. It is highly recommended that all customers engage the services of a trained Brocade Professional Services Consultant when first deploying the Brocade encryption solutions. Experienced SAN users will quickly realize that these solutions are quite different from a standard Fibre Channel (FC) deployment and that they require specialized knowledge and training. Furthermore, the consequences of an error during installation can be quite severe, including lost access to data and possible data corruption. Nevertheless, once initially configured, the Brocade encryption solutions are relatively simple to manage, and common operations—such as key management—are entirely transparent.

For additional information, refer to the appropriate version of the Brocade Fabric OS® (FOS) Release Notes, Encryption Interoperability Matrix, and Fabric OS Encryption Administrator's Guide for your Brocade FOS release, all available on www.brocade.com.

ESSENTIAL ELEMENTS OF CRYPTOGRAPHY AND SECURITY

The word "cryptography" is derived from the Greek words "kryptos," which means hidden, and "graphia," which means writing—so it is the art of hidden writing. Stated more completely, cryptography is the art, science, skill, or process of communicating with or deciphering messages written in code. Scholars speculate about the first use of cryptography, but one fact is indisputable. The need to exchange or store sensitive information in a manner that only the parties involved can understand has been around for a very long time—certainly for several centuries.

Symmetric vs. Asymmetric Cryptography

One of the enduring problems in cryptography is the distribution of keys. How do you distribute a secret key and then minimize or eliminate the risk of compromise if the key is intercepted? This problem is compounded when the key used to encrypt the message is the same as the one used to decrypt it, as in the case of data-at-rest encryption.

Symmetric Keys

Symmetric cryptography uses the same key or a secret key to encrypt and decrypt messages. An example is the Caesar cipher. Since the same key is used for both encryption and decryption, anyone in possession of the key can decrypt the encoded message using that key. Distributing the keys to authorized persons poses a particular challenge. Extreme measures are sometimes necessary to ensure a secure key exchange. If the key is stolen or intercepted during the transfer process, the code is deemed broken, and the encrypted message is no longer considered to be secure. Examples of well-known symmetric key algorithms are Data Encryption Standard (DES), 3DES (pronounced "triple DEZ"), and Advanced Encryption Standard (AES).

Asymmetric Keys

Asymmetric cryptography addresses the key exchange issue by using two different keys—one to encrypt and another to decrypt. Exchanging keys in times of war on the battlefield certainly offered its challenges, but the Internet and e-commerce present even greater challenges. How can you conduct millions of transactions per day at wire speeds across the world and ensure that each transaction is authenticated?

Asymmetric cryptography is also called public key cryptography, since it makes use of keys that are known publicly. A public key exchange system works on the principle of encrypting a message using a combination of a public key and a private key. Each party has its own public and private keys, which are different but mathematically related. Examples of familiar asymmetric key algorithms are used with Public Key Infrastructure (PKI) and RSA (which stands for Rivest, Shamir, and Adelman, the inventors).

There are several ways of implementing public key exchanges. Below is a high-level example of how asymmetric keys work.

Suppose that Jim sends Maria a message that only Maria is able to read. Both Jim and Maria have a private key that only they know about. They also have a public key that is available on a public server containing the public key repository. Jim queries the repository to obtain Maria's public key and uses it with his own private key to encrypt the message. The message is sent to Maria, and she then retrieves Jim's public key. Using the combination of Jim's public key and her private key, she can then decrypt the message and read it.

Bob is a hacker, and he intercepts the message between Jim and Maria. Since Bob does not know either Maria or Jim's private key, he is unable to decrypt the message using just Maria and Jim's public keys. This example is illustrated in Figure 1.

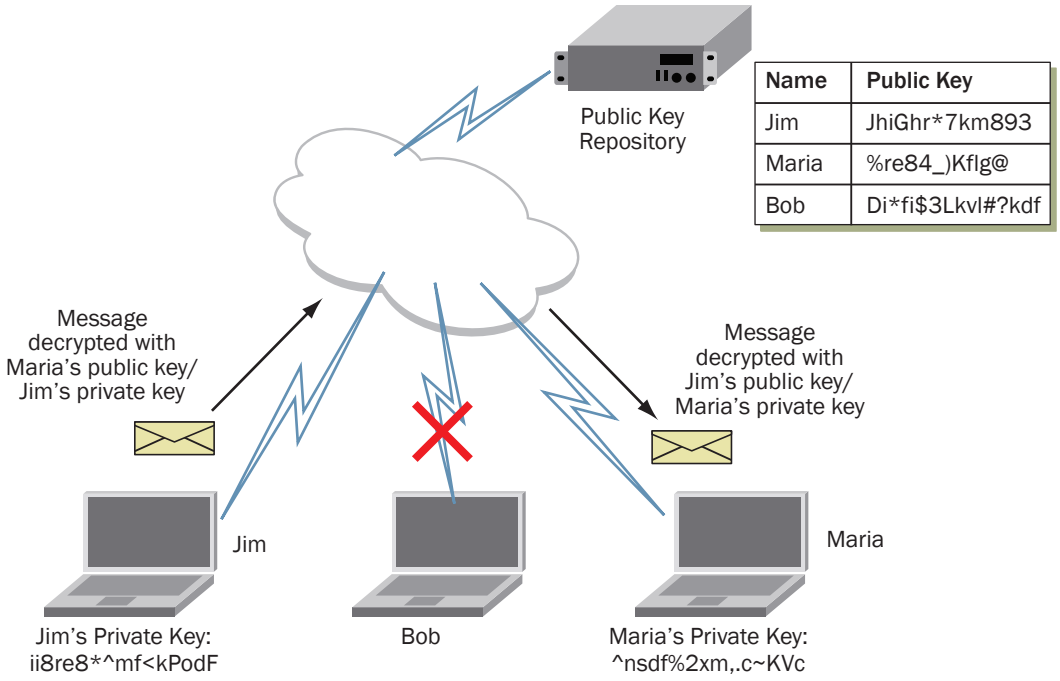

Figure 1. Simplified public key exchange.

Key Management

The decision to encrypt information residing on disk or tape creates a long-term commitment and a dependence on the encryption keys. After being created, keys need to be backed up and managed. Keys can be lost, stolen, or destroyed intentionally, or they can expire after a predetermined period of time. All of these are security vulnerabilities.

Loss of encryption keys is comparable to losing data. Unlike data in flight, the keys for data at rest must be available for as long as the data needs to be read. In the case of patient health records, information may need to be retained for seven years after the death of a patient, which could amount to several decades. Keys can also be stolen or compromised, in which case the information must be re-encrypted or rekeyed using a different key to ensure the confidentiality of the information.

Media such as disk and tape also have a limited shelf life, and they go through evolution cycles to an eventually incompatible format. Encrypting data-at-rest requires management of the encryption key for the life of the data. Encryption keys are usually managed by a comprehensive key management system, because keys need to be managed for an extended period of time. A key management system is used to manage the lifecycle of keys. Encryption key information needs to be refreshed as the media expires, and the data has to be re-encrypted using the same key or a new key.

Finally, encryption keys need to be backed up in a secure manner to avoid being compromised in the process. Keys can be backed up to a key vault as part of a comprehensive key management solution used to establish policies and manage the keys throughout their lifecycle. For redundancy, a typical key vault is implemented with two or more units to prevent single points of failure and mitigate the impact of a catastrophe. If the primary key vault becomes unavailable, the secondary or clustered key vault can accept or provide keys to the encryption device.

Key management solutions are implemented using two basic methodologies to exchange the keys between the encryption device and the key management solution: trusted and opaque.

Trusted Key Exchange

Trusted key managers have the ability to securely obtain cleartext keys. To protect the keys during the transfer, a trusted relationship must be established between the two devices. For example, the device performing the encryption must be able to store the encryption key in the key vault. The encryption device and key vault must authenticate each other to ensure that both are authorized to exchange keys. When each device is authenticated and authorized, then the trusted relationship is established. An example of a trusted key exchange is shown in Figure 2.

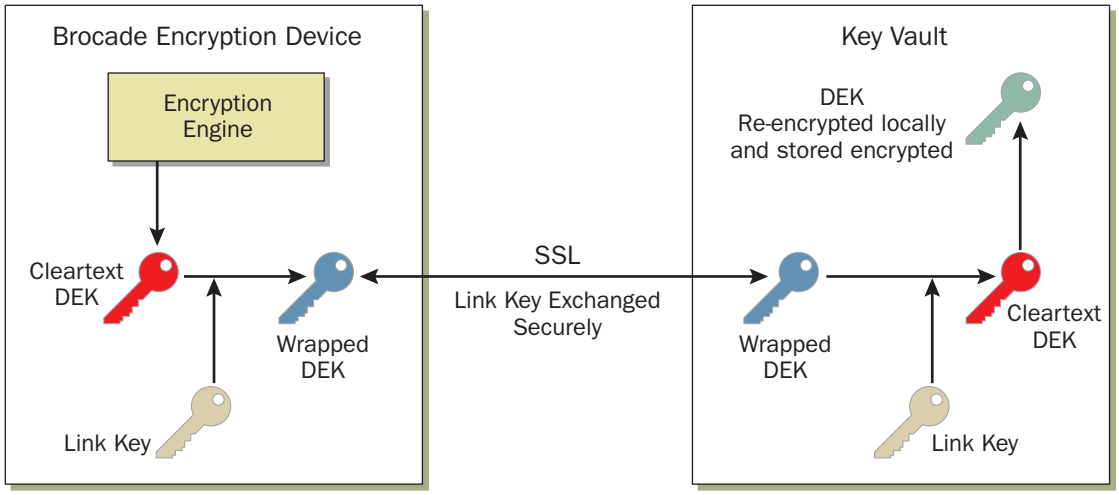

Figure 2. Trusted key exchange.

In the example in Figure 2, a link key (orange key) is used to securely exchange keys between the encryption device and the key vault. The Data Encryption Key (DEK—shown in red) created by the encryption engine is then wrapped (encrypted) using the link key, resulting in a wrapped key (blue key). The wrapped key is sent across a secure channel to the key vault, where it is unwrapped using the link key, resulting in the unwrapped or cleartext key (red key).

To prevent key exchanges from being sniffed or intercepted during transmission between encryption devices and key vaults, most vendors use secure channels for the key exchange (usually based on Secure Sockets Layer [SSL]), or they wrap the key using a symmetric key before sending it over the channel. Many variations exist for the key exchange process. For example, the NetApp Lifetime Key Management (LKM) and the SafeNet KeySecure (in LKM-compatible SSKM mode) use a secure channel (SSL) and wrap (encrypt) the key before sending it across the secure channel.

Opaque Key Exchange

With opaque key managers, the key vault never has access to cleartext keys; the keys are always received in a wrapped or encrypted form. One of the advantages of the opaque key management solution is that it does not require a hardened chassis and can be implemented using a software-only solution in a typical server. Figure 3 illustrates the simplified opaque key exchange process.

One of the primary distinctions between an opaque key exchange and a trusted one is that the DEK is wrapped prior to sending it to the key vault, where it is stored “as is.” With a trusted key exchange, the wrapped key is unwrapped at the key vault and then rewrapped using a different key encryption key. An opaque key vault does not contain information on how the DEK was initially encrypted.

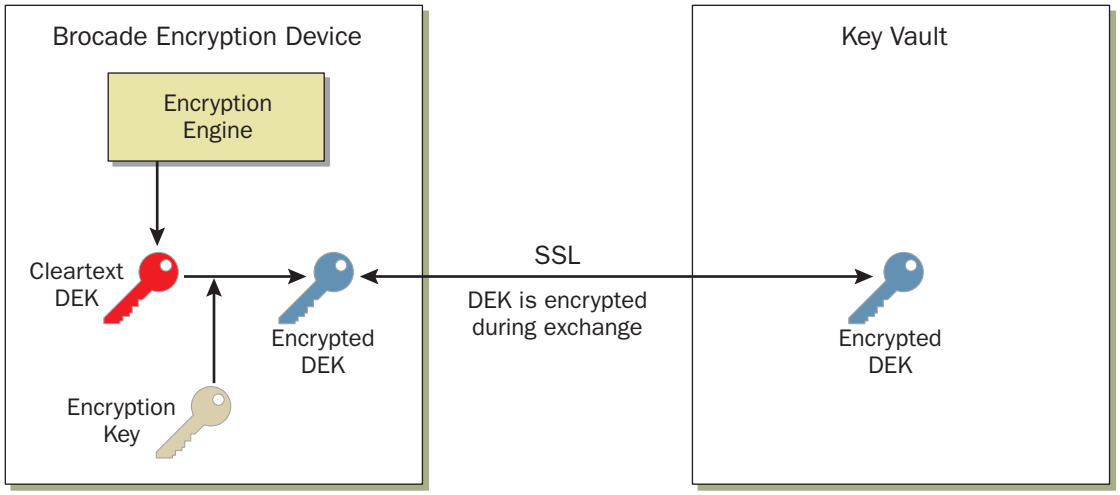

Figure 3. Opaque key exchange.

The RSA Data Protection Manager (DPM), HP Enterprise Secure Key Manager (ESKM), IBM Tivoli Key Lifecycle Manager (TKLM), and Thales keyAuthority are examples of key managers that work in conjunction with Brocade encryption devices as opaque key management solutions.

Cryptographic Algorithms

A cryptographic algorithm or cipher is the actual procedure used to manipulate a readable message and render it unreadable. The readable message that is input to a cipher is called plaintext, and its output is called ciphertext.

Early ciphers encouraged security through obscurity. Proprietary algorithms were kept secret for fear of their being discovered and subsequently broken. With certain exceptions, notably military-grade applications, this approach has been replaced by the use of open algorithms that withstand public scrutiny. Proprietary encryption algorithms are generally not considered secure, since they do not benefit from being scrutinized by either the cryptographic community at large or the general public. These algorithms are usually analyzed by a group of elite professional cryptographers, whose focus on only one perspective can result in an overlooked flaw in the security of the algorithm.

An open algorithm, on the other hand, has this advantage: At some point, thousands of individuals attempted to break it. If thousands of people from different professions and viewpoints are unable to break the code, then the algorithm certainly can be considered secure. When someone eventually breaks the code, it becomes public knowledge, and the useful life of that algorithm is ended.

The complex process of designing a cryptographic algorithm should take into consideration these factors, to ensure its efficient use practical commercial applications:

- **Speed of encryption:** A highly complex and completely unbreakable algorithm has no practical commercial use if it also requires excessive amounts of processing power to compute, which drastically affects performance.
- **Memory usage:** Algorithms that use too much memory to perform their computations and manipulations may require memory components that are too large to physically fit in certain portable devices. This may restrict their practical application.
- **Range of applications:** The ability to implement the algorithm in a wide range of devices—from supecomputers and disk arrays to Smart Cards and radio-frequency identification (RFID) devices—can affect its value
- **Cost:** If the cost to implement the cryptosystem is too high, then it may not find commercial relevance. Military and intelligence applications sometimes warrant a high cost in exchange for stronger cryptographic capabilities.

- **Openness:** In support of Kerckhoffs' principle (stated by Auguste Kerckhoffs in the 19th century), a cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

There are three basic categories of ciphers: block ciphers, stream ciphers, and hashing algorithms.

Block Ciphers

Block ciphers are used to encrypt data as an entire block, as opposed to one bit at a time. An entire block of data is processed at the same time by the block cipher. A plaintext message is broken down into fixed-length blocks and passed to the block cipher as plaintext. Each plaintext block is encrypted with the key to create a ciphertext block that is the same size as the input plaintext block. The decryption process takes the ciphertext message and breaks it down into fixed-size blocks. Each ciphertext block is decrypted using the key to produce a plaintext block that is the same size as the input ciphertext block, as shown in Figure 4.

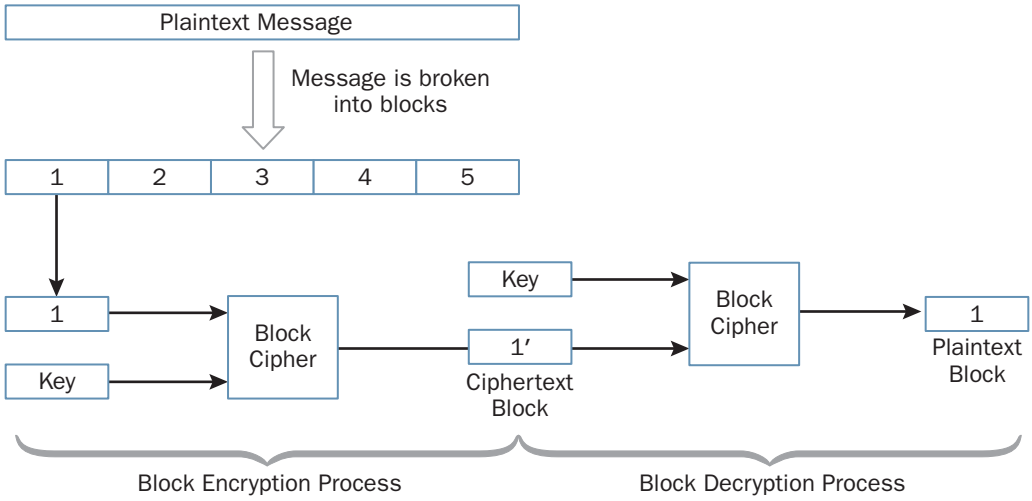

Figure 4. Block cipher encryption/decryption.

Stream Ciphers

Stream ciphers process plaintext one bit at a time, as shown in Figure 5. Generally, stream ciphers are considered less secure, since there is a higher risk of having repeating patterns. For this reason, block ciphers are more commonly used. Block ciphers can, however, be used on streaming data when they are operating in a streaming mode of operation, such as the counter (CTR) mode discussed later in this section.

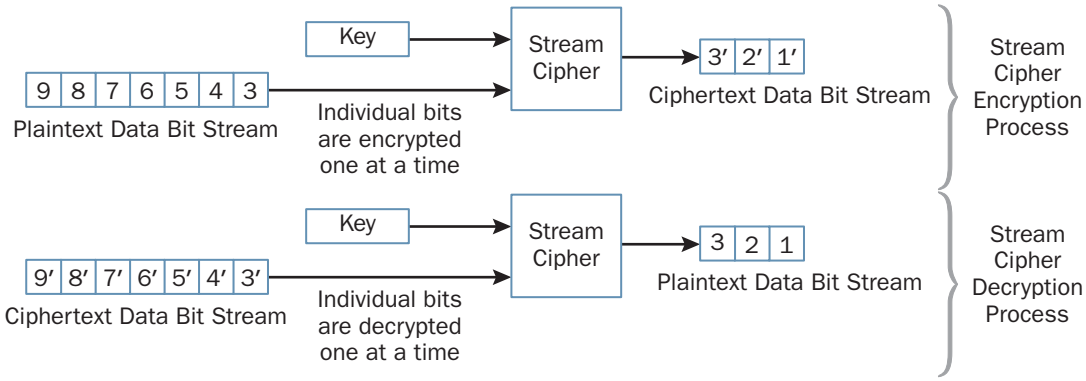

Figure 5. Stream cipher encryption/decryption.

Both block and stream ciphers address the data confidentiality issue by rendering the data unreadable without the key. Hashing algorithms, on the other hand, address the integrity issue by providing a means to verify that data has not been modified.

Digital Signatures

A digital signature, shown in Figure 6, is exactly what it says: it is the equivalent of a person's paper signature but is used for digital transactions. Digital signatures cannot be repudiated later.

A digital signature is created as follows:

1. A message is created.
2. The message is passed through an algorithm to generate a hash value.
3. The hash value is encrypted using a private key from some public/private key authority.
4. The resulting encrypted hash is the digital signature.

The validation process at the other end is as follows:

1. The message is passed through the same hashing algorithm.
2. The digital signature is decrypted using the public key of the sender.
3. The resulting decrypted hash is compared with the newly calculated hash.
4. If the hash values match, then the message is deemed valid.

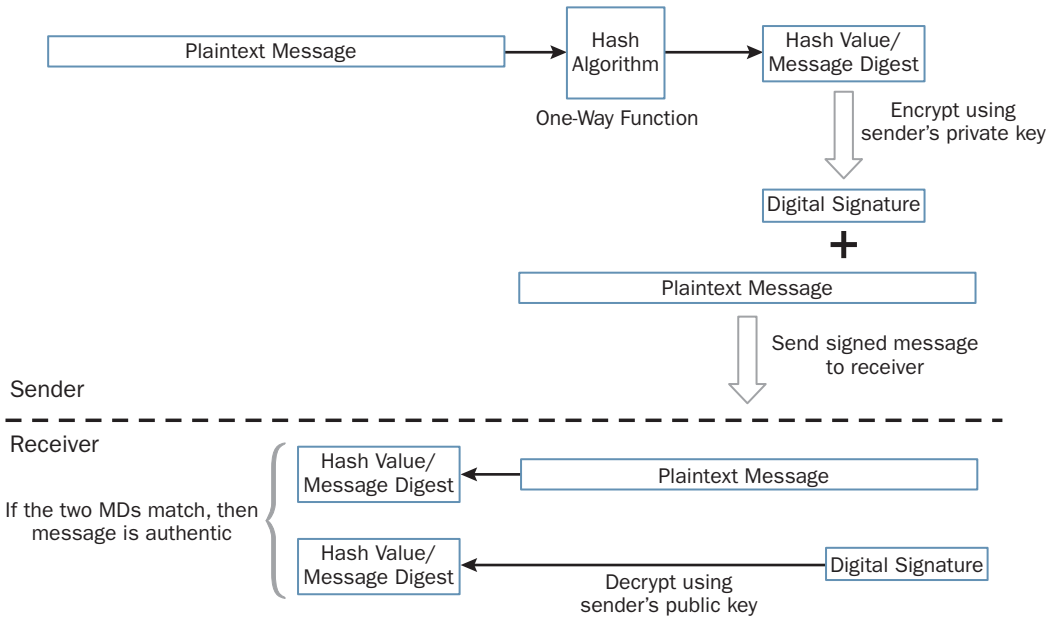

Figure 6. Digital signature.

Digital signatures provide non-repudiation and integrity to prevent someone from claiming that they did not perform an action or approve a transaction, and to confirm that the message has not been modified. Digital signatures are extensively used when making financial transactions over the Internet, to verify the authenticity of the person making the transaction and also to protect the vendor from clients claiming they never agreed to purchase an item.

Modes of Operation

A cryptographic algorithm can be applied in different ways depending on the type of data and specific requirements of its application. For example, some data is fixed length and must remain exactly the same size after it is encrypted, as is the case with block data written to disks. In other contexts, such as tape backup applications, the data is streaming to the device very rapidly on flexible media. Instead of creating a different cryptographic algorithm for each application and type of data, the same algorithm is used in different ways to accommodate the requirements. Furthermore, encrypting data bit by bit as it is transported serially through a wire requires another method of encrypting data. These methods are called modes of operation.

The following describes the modes of operation used by Brocade encryption solutions:

- **Electronic Codebook (ECB).** ECB divides the message into equal-size blocks that are encrypted separately. ECB is not very good for hiding patterns, since identical plaintext blocks encrypt to identical ciphertext blocks. ECB is used by Brocade encryption solutions to encrypt block and streaming data on disk and tape media in the DataFort-compatibility mode. This mode of operation is used in Brocade encryption solutions only when it is deployed in DataFort compatibility mode in support of existing DataFort environments.
- **Galois Counter Mode (GCM).** GCM is a similar mode of operation to Counter mode, with the addition of an authentication component called the Galois mode. Authentication is usually a computing-intensive process, which is not acceptable for streaming data. Authentication is also necessary to prevent certain types of attack on a data stream. The Galois mode was developed to authenticate messages at very high speeds with minimal performance impact on the data throughput. GCM is used as the native Brocade encryption mode of operation for encrypting block data on tape media.
- **XEX-based Tweaked Codebook with Stealing (XTS).** This mode of operation was designed for data formats that are not evenly divisible by a given block size, as is the case for disk drives with sectors not evenly divisible by their block size. XTS is used as the native Brocade encryption mode of operation for encrypting block data on disk media.

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) was developed by the National Institute of Standards and Technology (NIST) to replace the DES through a competitive process, in which 15 competitors submitted proposed algorithms. The Rijndael algorithm, proposed by Vincent Rijmen and Joan Daemen, two Belgian engineers, was selected as the new encryption standard in 2000. The AES is defined in the Federal Information Processing Standards (FIPS) publication 197. The Rijndael algorithm is a symmetric key block cipher that supports keys with 128 bits, 192 bits, and 256 bits (AES-128, AES-192, and AES-256 respectively). It was rapidly adopted by the industry. Most commercial applications for encryption of data-at-rest use AES-256.

The AES standard is the first to use an open cipher that is available to anyone, which distinguishes it from its predecessor, DES. Although there was some controversy around DES, which was co-developed by the National Security Agency (NSA), as to whether the NSA had created a back door into the algorithm, the open nature of the AES standard has all but eliminated this possibility.

Digital Certificates

A digital certificate is sometimes confused with a digital signature, but they are very different. A digital certificate is the equivalent of an ID card and is issued to an individual by a trusted Certification Authority (CA). It is composed of the owner's name, a serial number, an expiration date, a copy of the owner's public key, and the digital signature of the CA. Some digital certificates use the standardized X.509 format, defined in RFC 2459.

As of Brocade FOS v4.2, Brocade switches came preloaded with a digital certificate. Digital certificates are no longer preloaded (since the release of Brocade FOS v5.1), but one can still be obtained if desired. This digital certificate was used to authenticate switches that were joining a secured fabric using the Switch Connection Control (SCC) policy.

Federal Information Processing Standards (FIPS)

IT security product consumers may not necessarily have the expertise, knowledge, or resources necessary to fully evaluate products—that is, to determine whether the security of a product is appropriate and meets their requirements. Assertions from the vendors and developers of these products may not provide the highest level of confidence to the consumer. To increase this level of confidence, a consumer can hire an independent organization to evaluate products for them, or they can simply use a pre-established standard that vendors must comply with.

When U.S. Federal and private sector organizations make purchasing decisions for security products that perform a cryptographic function, they must evaluate the proposed products from each vendor. This is sometimes accomplished by creating an evaluation matrix comparing the different product features. A compliant/non-compliant system may be used, while others may prefer a weighted point system that gives more importance to some functionalities than others. Since this matrix can become quite large and complex when multiple vendors respond to a tender, a standard was created to establish base security criteria levels for all vendors.

The National Institute of Science and Technology (NIST), reporting to the U.S. Department of Commerce, created Publication 140-2 on May 25, 2001 (also known as the Security Requirements for Cryptographic Modules), to simplify the acquisition process. FIPS 140-2 was developed primarily for U.S. Federal organizations, and it provides standard evaluation criteria for cryptographic modules used in certain security products. It is sometimes used by private sector organizations in North America but is seldom used in other parts of the world. The FIPS 140-2 standard applies specifically to the cryptographic modules used in security products. A cryptographic module consists of the hardware, software, and/or firmware used to implement security functions (including encryption algorithms and key generation). It is contained within a cryptographic boundary that establishes its physical boundaries.

Each organization has different security requirements and requires different degrees of security, hence FIPS 140-2 defines four security levels (see below). The lowest security level is 1, and each subsequent level builds on the previous levels.

The actual certification of the cryptographic module is performed by an independent lab, which validates the product to ensure it meets the criteria required for the security level required by the vendor. Once the tests are completed, the results are submitted to NIST and, upon their approval, the product is officially posted on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/inprocess.html.

Security Level 1

Security Level 1 provides the lowest level of security and defines production-grade equipment with no physical security. Nearly any product using a cryptographic module qualifies for this level of certification. An example of a Security Level 1 certified product is an ordinary laptop with a software-based encryption module.

Security Level 2

Security Level 2 enhances Security Level 1 with the tamper evidence requirement. Tamper evidence is implemented using special coatings, seals, or pick-resistant locks for removable covers and doors. If a protective cover or door is tampered with to allow physical access to critical security parameters or plaintext keys stored in the cryptographic module, the coatings or seals are broken and permanently modified.

Additionally, role-based authentication must be used to authenticate an operator with a specific role that allows that operator to perform certain tasks, such as deleting keys.

Security Level 3

Security Level 3 builds upon Security Level 2 with the addition of tamper-resistant mechanisms to prevent someone from gaining access to the Critical Security Parameters (CSP) stored in the cryptographic module. This may include tamper detection and response systems, which could, for example, zeroize the keys stored in the local cache when the cover or door is opened.

Security Level 3 must also include identity-based authentication mechanisms to authenticate a specific individual and verify that the person is authorized to perform the specified task.

Security Level 3 also requires that plaintext CSPs are exchanged using different ports than those used for other purposes (such as management interfaces). This enforces the principle of separation of duties, which allows different individuals to have authority over different types of functions, thus preventing one person having control over the entire process.

Security Level 4

Security Level 4 provides the highest level of security and builds upon Security Level 3. The physical security mechanisms at this level must provide a complete envelope of protection around the cryptographic module. All unauthorized attempts to physically access the cryptographic module must be detected and responded to by zeroizing all plaintext CSPs. The cryptographic module must also be protected against extreme environmental conditions that exceed the normal operating ranges for voltage and temperature.

Only the most demanding environments require products certified to Security Level 4—such as combat zones and highly secure facilities that use equipment containing highly sensitive information. Under these exacting conditions, the equipment must still be able to zeroize the CSPs. For this reason, some people refer to Security Level 4 as a "science experiment," since the testing process is extremely rigorous, lengthy, and expensive, and few products are certified to this level.

Encryption Methods Used With Brocade Encryption Solutions

Encrypting data on a storage medium involves encrypting the data with one key and decrypting it using the same key, hence symmetric key cryptography is used when encrypting data-at-rest. With Brocade encryption solutions, these keys are referred to as Data Encryption Keys, or DEKs. There are other types of keys used with Brocade encryption solutions. Brocade encryption devices use AES-256—the strongest standard encryption algorithm available today—to encrypt data-at-rest.

The DEKs are stored locally in the cache of the encryption devices, but they are also stored and, essentially, backed up to a key management solution or key vault. When authenticating with key vaults, asymmetric keys are used to authenticate between the Brocade encryption device and the key vault, using digital certificates.

BROCADE SOLUTION OVERVIEW

This section provides an overview of the Brocade encryption solutions, as well as some of the internal architecture. Key management is also reviewed as it applies to these solutions.

Brocade Encryption Solutions Overview

Brocade encryption solutions are available in two form factors that share the same internal hardware. The solutions are available as a standalone switch, the Brocade Encryption Switch, and as a blade for the Brocade DCX® 8510 Backbone and the Brocade DCX family of Backbone products—the Brocade FS8-18 Encryption Blade. The term "encryption device" used throughout this section refers to either the encryption switch or the encryption blade. A Brocade encryption solution includes the Brocade encryption device, along with all other components required for a production environment, such as the key vault.

The Brocade encryption device features the following:

- Up to 96 Gbps processing bandwidth for disk encryption
- Up to 48 Gbps processing bandwidth for tape encryption and compression
- Encryption using the industry standard AES-256 algorithm
- Hardware compression of tape data
- Disk encryption latency of 15–20 microseconds

- Tape encryption and compression latency of 30–40 microseconds
- Brocade internally-developed encryption ASIC technology
- FC switching connectivity based on the Brocade 8 Gbps Condor 2 ASIC
- Dual Ethernet ports for High Availability (HA) synchronization and heartbeats
- FIPS 140-2 Level 3 validated cryptographic boundary cover
- Smart Card reader used as a System Card (ignition key)

The System Card feature is included with all encryption devices but is not enabled by default. The ignition key is a Smart Card inserted into a Smart Card reader to enable the cryptographic capabilities of the switch. Without it, the Brocade Encryption Switch is limited in its operation as a regular 8 Gbps Layer 2 FC switch.

If the ignition key feature is used, then it is also imperative to store the Smart Card in a safe location after the cryptographic functions of the switch are enabled. The Smart Card must be reinserted in the reader (see Figure 7 and Figure 9 for the Smart Card Reader location) each time the switch is powered up (after a power shutdown) to enable the cryptographic capabilities of the switch. A system reboot does not require the use of the ignition key to re-initialize the encryption functionality.

Brocade Encryption Switch

The Brocade Encryption Switch is the standalone version of the hardware encryption device for data-at-rest. It offers the following features:

- 32 × 8 Gbps FC ports
- Three redundant fan modules
- Two redundant power supplies
- USB port
- RJ-45 GbE management port
- Two redundant RJ-45 GbE ports for intercluster communication
- FIPS 140-2 Level 3 compliant cryptographic boundary cover
- Smart Card reader used as a System Card (system key)

Figures 7 and 8 illustrate the Brocade Encryption Switch and its components.

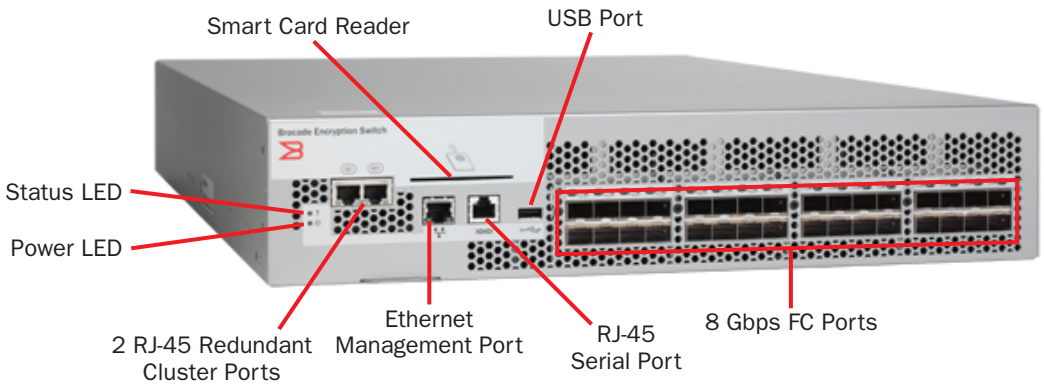

Figure 7. Front view of the Brocade Encryption Switch.

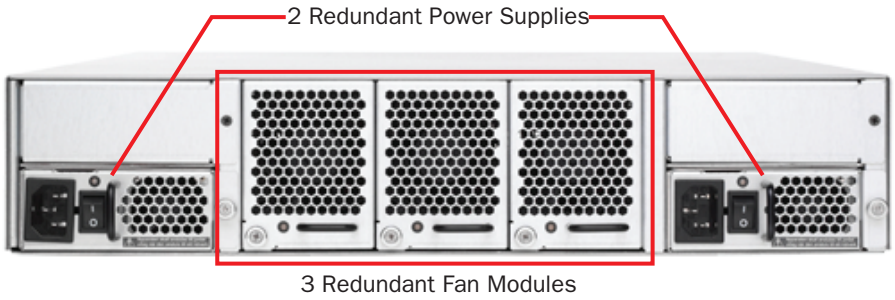

Figure 8. Rear view of the Brocade Encryption Switch.

The Brocade Encryption Switch is available in an entry-level version for disk encryption. Some environments may not require the full 96 Gbps of bandwidth for disk encryption or have the budget for this type of solution. This entry-level product was created offering up to 48 Gbps of disk encryption processing capability at a lower price point. The entry-level version is identical to the full-featured version, with the exception of encryption processing bandwidth; all 32 FC ports are still enabled and can be used to connect hosts and storage devices. Later, if the 48 Gbps encryption-processing capability is exceeded, a simple license upgrade to the full 96 Gbps version can be installed. Note that as the encryption processing for tape encryption is limited to 48 Gbps due to the precompression of the data, the use of the upgrade license may not provide any tape performance benefit.

Note also that hosts and storage devices involved in the encryption process do not need to be connected directly into the Brocade Encryption Switch. In fact, you can connect the Brocade Encryption Switch to an existing fabric using only Inter-Switch Links (ISLs), and the encryption process will still work, regardless of where the hosts and storage devices are located within the fabric.

Brocade FS8-18 Encryption Blade

The Brocade FS8-18 Encryption Blade is the embedded version of the Brocade Encryption Switch for the Brocade DCX 8510/DCX/DCX-4S Backbone. The Brocade FS8-18 has the same functionality and performance characteristics as the Brocade Encryption Switch:

- 16 × 8 Gbps FC ports
- USB port
- RJ-45 GbE management port
- Two redundant RJ-45 GbE ports for intercluster communication
- FIPS 140-2 Level 3 validated cryptographic boundary cover
- Smart Card reader for the System Card (system key)
- Up to four Brocade FS8-18 blades supported in one Brocade DCX-class chassis

Figures 9 and 10 illustrate the Brocade FS8-18 blade and its components.

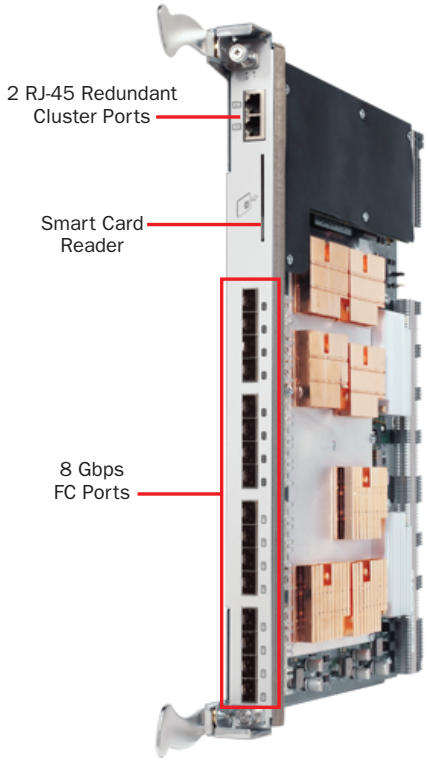

Figure 9. Profile view of the Brocade FS8-18 Encryption Blade.

The FIPS 140-2 Level 3 validation posed several challenges for the Brocade FS8-18. The typical Brocade enterprise-class platform blade has all of its ASICs exposed on the card. To prevent tampering with the components of the blade involved in the cryptographic (crypto) process, it was necessary to build a physical crypto security boundary protecting all the memory, the true random number generator, encryption, and Condor-2 ASICs. This boundary was created using a cover over these components, which in turn posed a new challenge: cooling. The cover cannot have vents, which could allow intruders to access the internal components with specialized tools, so copper heat sinks were placed on the cover to dissipate the heat, as shown in Figure 10.

As with the Brocade Encryption Switch, the Brocade FS8-18 Encryption Blade is also available in an entry-level version for disk encryption. The entry-level version of the blade applies to the entire Brocade DCX family chassis; a performance upgrade license upgrades the entire chassis, not individual blades. The Brocade DCX family chassis can support from one to four Brocade FS8-18 blades per chassis. With the entry-level version, the entire chassis is limited to 48 Gbps of disk encryption processing bandwidth per blade. The entry-level version affects only the encryption processing capability; all 16 FC ports are still enabled and can be used to connect hosts and storage devices. Later, if the 48 Gbps encryption processing requirement is exceeded, you can either add new Brocade FS8-18 blades, or you can upgrade all the encryption blades in the chassis with a simple chassis-level license upgrade to a full 96 Gbps of disk processing capability.

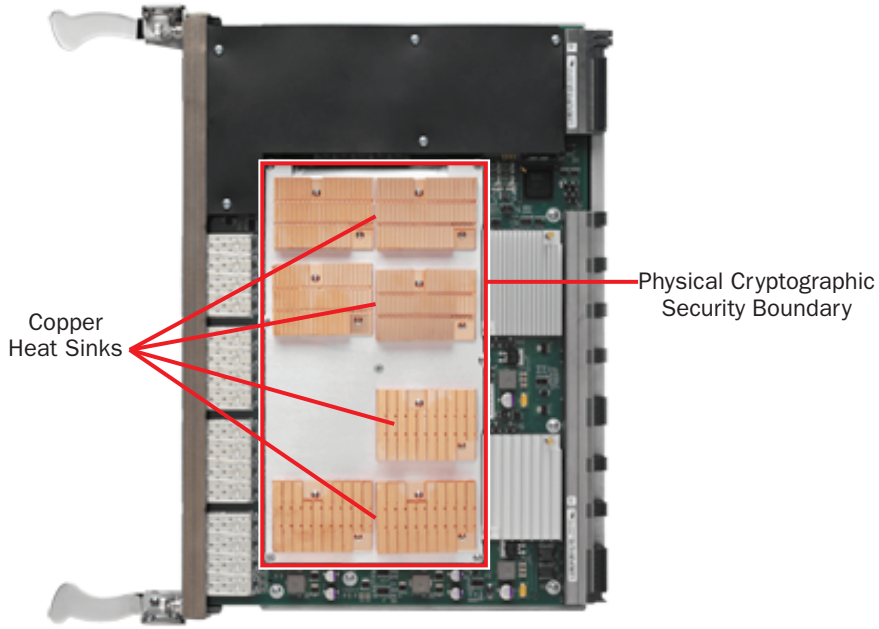

Figure 10. Side view of the Brocade FS8-18 Encryption Blade.

One advantage the Brocade FS8-18 blade has over the encryption switch is that all of the encryption traffic passes over the backplane, so you do not need to be concerned with ISLs. Additionally, it is not necessary to connect the hosts or storage devices involved in the encryption process into one of the 16 FC ports on the blade. In fact, frame redirection technology ensures that the encryption takes place, even though there are no hosts or storage devices directly connected into the blade.

Brocade Encryption Features

This section describes in detail the unique features and functionality of the Brocade encryption solutions.

Brocade Encryption Process

The Brocade encryption solution uses the industry standard AES-256 encryption algorithm implemented in hardware:

- Disk encryption is performed using the XTS mode of encryption, which is optimized for fixed-block data.
- Tape encryption is performed using the GCM mode of encryption, which adjusts for variable length and streaming data.

Compression is an important component of a data-at-rest encryption solution for tape. Once data is encrypted, it is no longer compressible. Compression works on the principle of searching for patterns and optimizing them. Strong encryption takes data and removes all patterns by randomizing the data. Once the data is randomized and all patterns are removed, then the compression algorithm has no patterns to optimize. If encrypted data is sent directly to a tape drive, then the native compression capabilities of that tape drive no longer operate. Hence, it is important to compress the data first and then send it to the tape drive to prevent an unnecessary increase in the number of tape media used for backups.

The compression algorithm used in Brocade encryption solutions are based on a variant of the standard gzip algorithm. The compression ratio obtained using this compression algorithm varies, like any other compression algorithm, depending on the type of data and how compressible it is. Data with a considerable amount of white space compresses quite well, while highly randomized data—such as video and precompressed files—may not compress at all. For most applications, a 2:1 compression ratio is achieved.

CryptoTarget Containers

A CTC (CryptoTarget Container) is created for each storage target port hosted on a Brocade encryption device and is used to define the application of encryption to a storage media. A CTC can be composed of only one storage port target, but it can have multiple initiators or hosts associated with it. A CTC can also have several Logical Unit Numbers (LUNs) behind the storage port in the CTC. It is up to the administrator to specify which LUNs should be encrypted and which ones should not be encrypted. It is important to note that frame redirection works at the World Wide Name (WWN) level and not at the LUN level. When a host has been added to a CTC, all frames for that host are redirected, regardless of whether a particular LUN is being encrypted; all traffic is redirected through the encryption device. Thus, it is generally preferable to mark every LUN to be encrypted for a given host, as there is no performance advantage or reduction in bandwidth consumption by leaving some LUNs in cleartext. Furthermore, once a storage port has been assigned to a CTC, it cannot exist or be defined in another CTC. Essentially, this forces all traffic to be redirected to a specific storage port and to go through the same encryption device.

NOTE: *You can still make the storage port accessible (with appropriate zoning) for other hosts, in case encryption is not required for one host and its associated LUNs. In this case, the hosts and associated LUNs are not added to the CTC.*

First-Time Encryption and Rekeying

The First-Time Encryption (FTE) process on a Brocade encryption solution is performed in-place on the same LUN. The process involves reading the first logical block on the LUN (which is in cleartext), encrypting it, and then writing it back in its original location as ciphertext. Subsequent blocks are encrypted in the same manner sequentially, until all the blocks in the LUN have been encrypted. You can perform this process offline or online, depending on your organization's business requirements.

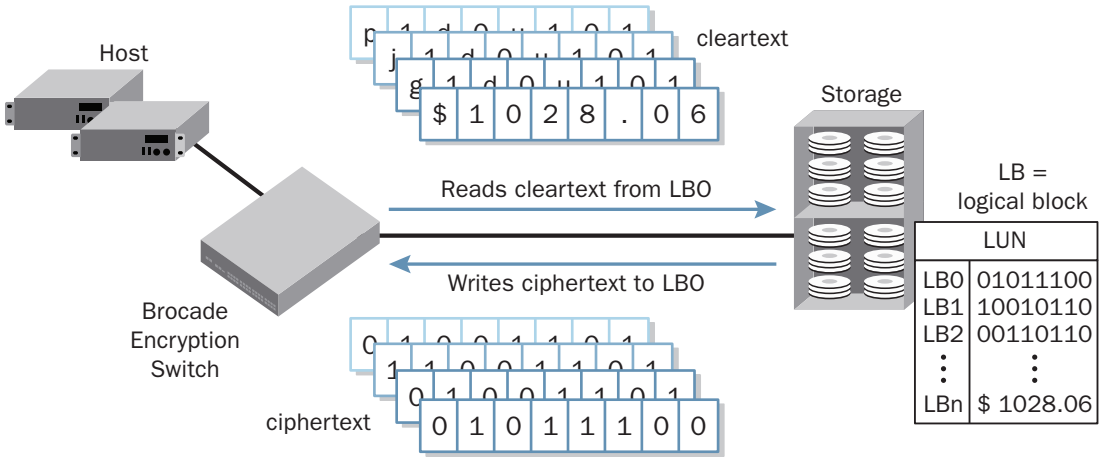

Figure 11. First-time encryption operation illustrating the FTE process.

The next encryption process is the rekey operation, in which a LUN is re-encrypted using a different key. There are two basic scenarios that may force a rekey operation: a compromised key or a security policy requirement. If a key is deleted or stolen, it is compromised, and the data encrypted using this key can no longer be considered secure. The security or risk management department may also implement a policy requiring that all keys must be refreshed on a specified schedule, such as every 36 months. This is often done due to a concern that keys were compromised without the department's knowledge. Thus the desire is to err on the side of caution by forcing a rekey of all encrypted data after a defined period of time. Rekeying can be performed automatically by setting an expiration date for a key while it is being created on the Brocade encryption device.

In-place rekeying is not possible for tape, since a tape drive is a streaming device, and the media itself is flexible. Rekeying data on a tape involves copying it to a new tape and encrypting it with a different key as the data is copied. With disk media, the process is much simpler, since the LUN with the compromised key can be rekeyed in-place and online if necessary.

During the rekey operation, the LUN actually has two keys assigned to it. Once the rekey process is completed, the original key is simply discarded. As with a first-time encryption, you can perform the rekey operation online or offline.

BEST PRACTICE:

- Make sure there are at least two encryption devices within the same encryption group that have access to the LUN involved in an FTE/rekey operation. In other words, you should have a DEK cluster before performing an FTE/ rekey operation.
- Perform FTE/rekey during periods of low application I/O to avoid excessive impact on application performance and an increase in FTE/rekey duration.
- Whenever possible, avoid performing a backup of the LUN while performing an FTE/rekey operation.
- Whenever possible, avoid data replication when performing an FTE/rekey operation.

Clustering and Availability

One of the principal tenets of security is maintaining availability. Needless to say, downtime can be expensive, and you must take precautions to prevent a loss of availability of information. This is particularly true for encryption solutions, since there is a complete dependence on the encryption keys to recover encrypted information. Compounding this problem is the importance of the applications that require encryption. Any loss of availability to information that is so important it needs to be encrypted is likely to be disastrous for its owners. You must take extensive precautions to protect the keys and to maintain the availability of the encryption solution.

As with any IT solution, there are many ways to ensure availability. Choosing the best method to maintain availability depends on the value of the information (and impact of a loss of availability), the risk and probability of disruption, and the cost of implementing high availability. As with all aspects of IT, the issue is often about getting the best value for your investment.

Clustering is commonly used to ensure protection against hardware failure. There are two types of clusters for Brocade encryption solutions, which can be used independently or simultaneously. The HA cluster provides hardware redundancy for the encryption devices. The DEK cluster allows two or more encryption devices to share the same keys and, therefore, to access the same LUN(s).

HA Cluster

The HA cluster is an active-passive clustering configuration in which one encryption device is a warm standby for the other encryption device it is paired with. Only two encryption devices can form an HA cluster, and they must exist within the same fabric. Heartbeats are exchanged between the encryption devices using redundant Gigabit Ethernet ports through an out-of-band dedicated network to let the other know it is still “alive.” This same dedicated network is used to synchronize key state information between the units to allow one to take over for the other when its HA pair has failed and no longer appears in the nameserver. Unlike the DEK cluster described below, the HA cluster does not result in a path failover following a failed encryption device.

Since the HA cluster uses an active-passive configuration per CTC, it is more efficient to balance the load across both encryption devices instead of having the entire load on one unit, with the other unit being inactive unless the active unit fails. It is possible for each encryption device to be active simultaneously and carry its own encryption load. In this case, each unit is active with its load and passive while waiting for the other unit to fail over. If one encryption device fails, it is important to consider the available bandwidth on the other cluster member and its impact on application performance.

For example, say that Encryption Device A in the cluster is currently pushing 52 Gbps of traffic and Encryption Device B is pushing 61 Gbps. If Encryption Device B fails, Encryption Device A takes over the CTCs. Since Encryption Device A is already pushing 52 Gbps and now has an additional 61 Gbps, for an aggregate of 113 Gbps of traffic, this exceeds the 96 Gbps capability of the encryption device. At this point, there will be more I/O going through Encryption Device A than it can handle, and a performance bottleneck will occur, resulting in a downgraded performance of the production environment.

Encryption Group

An Encryption Group is a group of encryption nodes that share the same key management configuration and CTC configuration.

DEK Cluster

The DEK cluster by definition shares the same data encryption keys as all other encryption devices within a cluster management group. An encryption group contains up to four encryption devices that share the same DEKs. For each encryption group, you must designate one encryption device as the group leader. The group leader is responsible for functions such as the distribution of the configuration to the other members of the group, authenticating with the key vaults, and configuring CTCs.

It is important to note that the DEK cluster offers good redundancy, since the loss of one encryption device does not necessarily result in a loss of production, given that disk solutions are implemented using dual paths. With a dual path, there is always an alternative path for the data to take to get to the LUN. For this reason, the HA cluster is very seldom used, other than for the most stringent application requirements and environments where downtime cannot be tolerated and intrafabric redundancy is required.

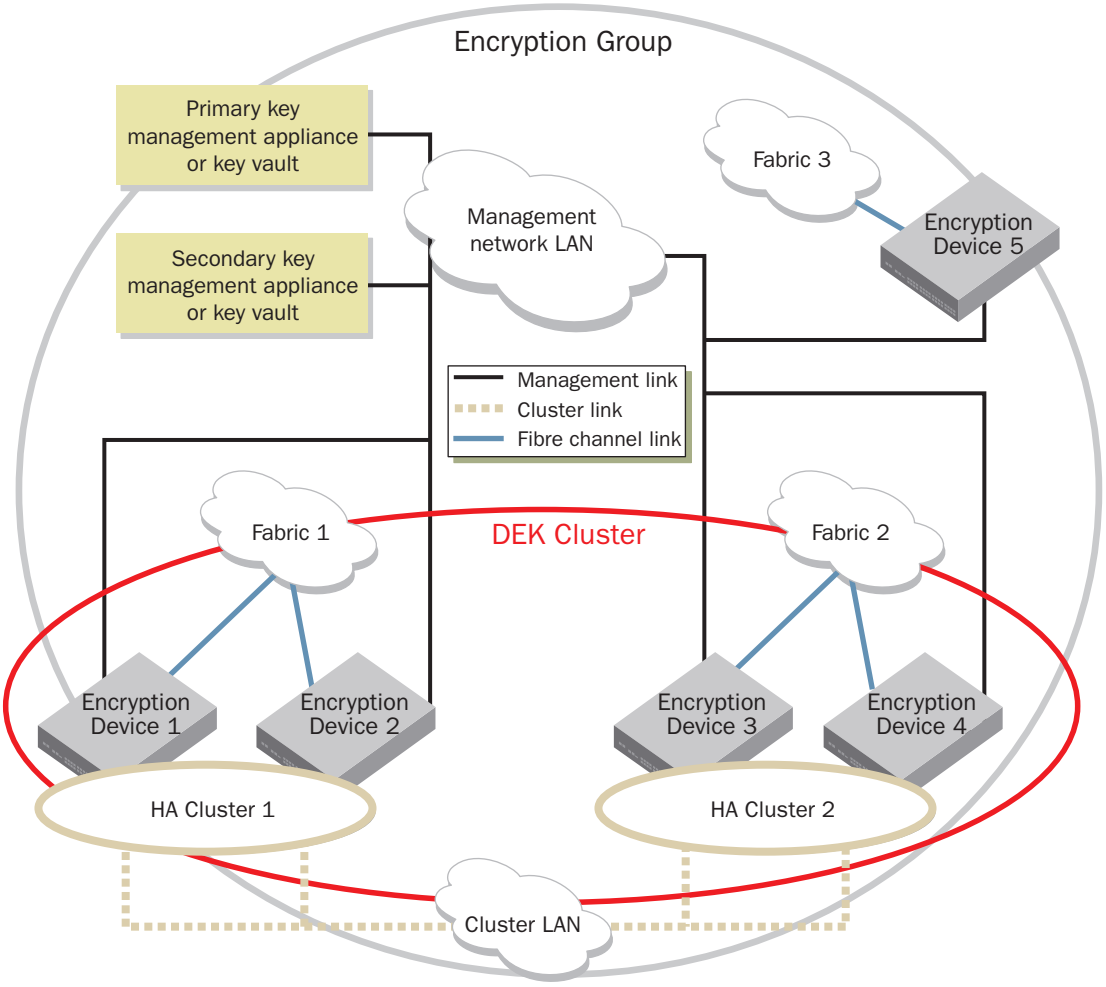

Figure 12. HA and DEK clusters.

Key Management Solutions

As discussed previously, once data has been encrypted on storage media, the keys become critical, and you must take specific measures to protect them. Keys need to be backed up, as they can be lost, stolen, or destroyed intentionally or can expire after a predetermined period of time. This is why you need a key management solution or key vault. Brocade made the decision early on to leverage existing and proven partner solutions for key management, supporting key management systems from several industry-leading vendors. Refer to the Brocade Encryption Interoperability Matrix on www.brocade.com for a complete list of supported key management systems.

Brocade supports the development of the OASIS KMIP (Key Management Interoperability Protocol) standard and offers native KMIP 1.0/1.1 client support in Brocade FOS v7.1. Refer to the Brocade Encryption Interoperability Matrix on www.brocade.com for the most current list of key management systems that support OASIS KMIP.

Brocade encryption devices generate the actual data encryption key and store it locally in the cache. The DEK is used to encrypt data using the AES-256 encryption algorithm. Before any data encryption begins, the key must be backed up to a key vault or key manager and then placed in the local cache before it can be used. Subsequently, the DEK is exchanged with the other members in the Encryption Group.

When a new LUN, tape media, or LUN with existing cleartext data is encrypted, the Brocade encryption device generates a DEK. This key is then backed up to the primary key vault and secondary key vaults if they exist. Once the primary key vault has successfully stored the DEK, it confirms this to the encryption device. The DEK is then synchronized with all of the other members in its encryption group, as shown in Figure 13. Only after all of this has occurred is the new key used for the encryption process.

It is important to note that a loss of connectivity between the encryption device and the key management appliance/server does not affect production I/O. Such a loss of connectivity results only in the inability to create new LUNs for encryption; the production I/O continues uninterrupted, since the DEKs are cached in the local encryption device and remain accessible for the encryption/decryption process.

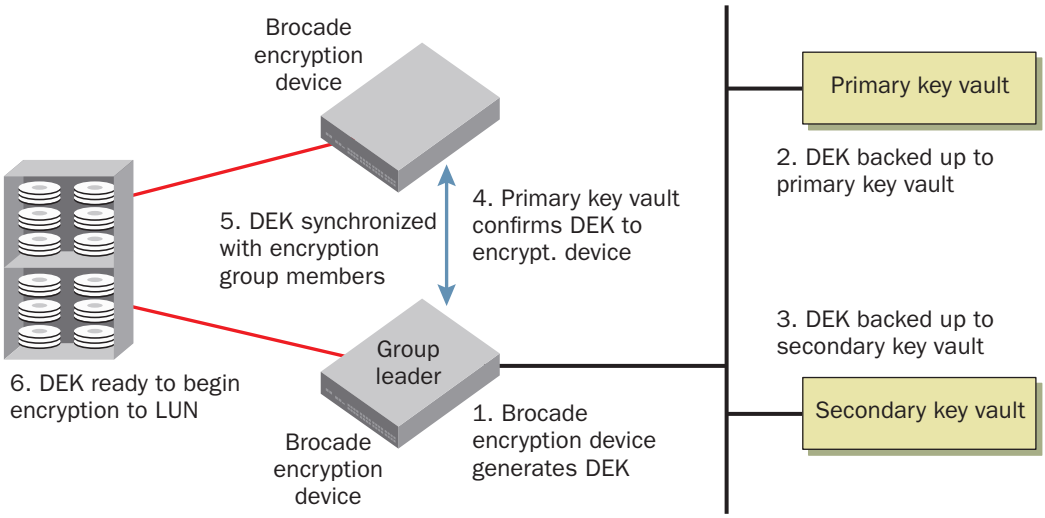

Figure 13. DEK synchronization.

Redundant Key Vaults

You can also configure key vaults in a clustered configuration to provide redundancy. Each key management solution vendor offers different features and functionality around clustering, but all of them provide some form of clustering capability. Although clustering the key vault is an optional feature, it is certainly recommended as a best practice. Ideally, a key vault should be located in at least two physically separate locations to provide the maximum redundancy. Specific details are provided in a later section on deploying these solutions.

Encrypting with Backup Applications

Although only the payload in the frame is encrypted, special adaptations are needed for each backup software vendor. There are two basic elements in a backup solution that an encryption solution must take into consideration.

The first element is how the backup application writes its metadata to the tape media. This is important for determining where to place the key information on the media for later data recovery. Obviously, the actual unencrypted key is not stored on the tape media itself (this would be like sliding a spare house key under the front doormat). In fact, only an index referring to the key is written to the tape media as part of the tape header written by the backup application.

The second element is how each backup application handles tape pools. Keys can be assigned either on a per-tape media basis or on a per-pool basis. As a best practice, you should assign one key per physical tape media to reduce the rekey overhead if a key gets compromised. Nevertheless, for some special situations, it may be useful to use one key per pool. For example, if you plan to send a set of tapes to a third party, perhaps for auditing purposes, you can use a single key for the entire tape set, to simplify the reading of the tapes at the other end.

Brocade encryption solutions support a variety of popular backup software applications. Refer to the appropriate version of the Brocade Encryption Interoperability Matrix, available on www.brocade.com, for a complete list of supported backup software solutions.

Brocade Encryption Solution Internals

The Brocade encryption device is a state-of-the-art hardware product built to integrate seamlessly into an existing SAN infrastructure and integrate with the market leaders of encryption key management. Both the encryption switch and the encryption blade share essentially the same hardware components and offer the same functionality, just in a different form factor. The encryption blade does not have a USB port, serial port, or management Ethernet port, and the switch does not have a backplane, but the rest of the setup is basically the same. Figures 14 and 15 illustrate the simplified internal architecture of the Brocade Encryption Switch and Brocade FS8-18 Encryption Blade, respectively.

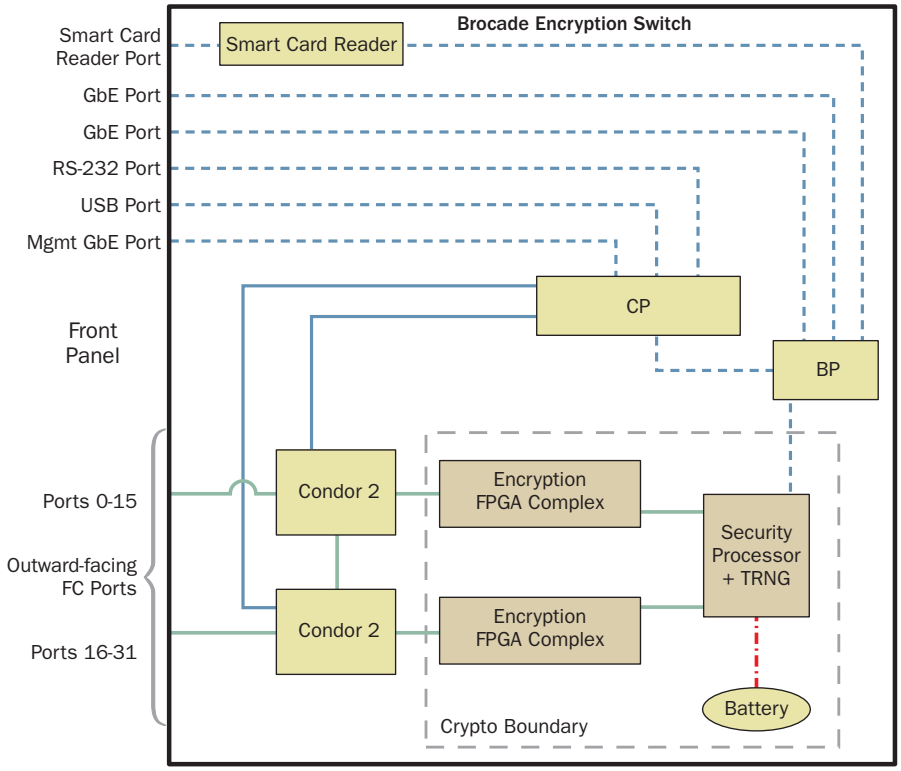

Figure 14. Brocade Encryption Switch internal architecture.

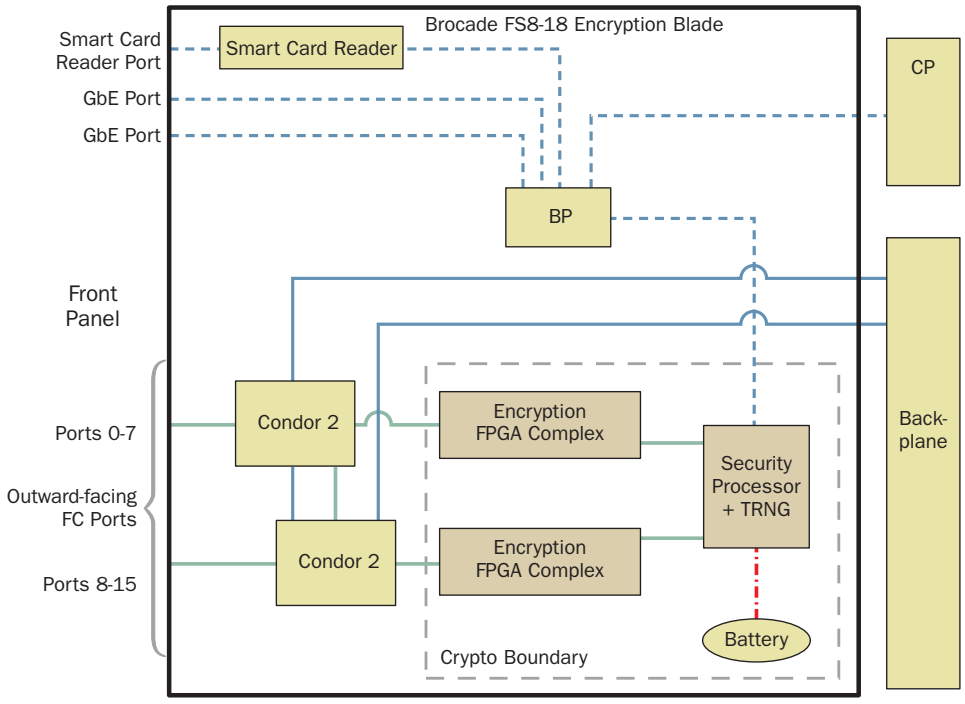

Figure 15. Brocade FS8-18 Encryption Blade internal architecture.

The components described in the following three sections are enclosed within a physical crypto boundary. The security boundary is designed to comply with the FIPS 140-2 standard at Level 3 to isolate all hardware components involved in the processing of cleartext keys. The encryption switch cover is the physical crypto boundary for the Brocade Encryption Switch, and the encryption blade has a special cover that covers the necessary hardware on the card.

Encryption FPGA Complex
The FPGA (Field Programmable Gate Array) complex is composed of several FPGAs and other hardware components. The principal encryption component is the FPGA. An FPGA is a programmable hardware device that contains instructions to perform specific functions. The advantage of the FPGA is that it is programmable, and the instructions can be changed at any time. A new feature or enhancement can be made without requiring a hardware upgrade.

The FPGA complex is where the actual encryption and compression are performed in the encryption device, in addition to a few other functions.

Security Processor + TRNG
The Security Processor provides data security functions such as generating and processing symmetric keys (the DEK) based on the True Random Number Generator (TRNG).

The TRNG is the hardware component used to generate the random number from which the DEK is generated. A TRNG uses physical phenomena such as transient noise to truly randomize the random number generation process. The TRNG used in this solution meets the FIPS validation requirements.

Battery
A lithium-ion battery is used when there is no power to the encryption device. This battery has a life span of approximately ten years after power has been removed from the encryption device. It is used primarily to sustain the FIPS 140-2 Level 3 tamper response mechanism, which zeroizes the keys stored in the local cache once tampering has been detected.

The remaining components are found outside the security boundary.

Control Processor (CP)
The Control Processor performs various control and coordination functions such as authentication processes.

Blade Processor (BP)
The Blade Processor acts as a bridge between the Security Processor and the Control Processor, as well as with the Smart Card reader and GbE ports.

Condor 2 ASIC
The Condor 2 ASIC features 40 8 Gbps ports and is the heart of the FC Layer 2 switching. Each encryption device has two Condor 2 ASICs.

Metadata
Recovering encrypted data from a given storage medium requires knowledge of what Data Encryption Key (DEK) to use to decrypt the data. Since it would be a security violation to actually write the key itself on the media, a Key ID is written instead. This Key ID is then mapped back to the actual DEK in the encryption device. For disk media, the Key ID is derived from the volume serial number of the LUN. The metadata is stored in a small number of bytes at the beginning of the first 16 blocks of data of the LUN. Since production data most likely resides in these first 16 blocks, the production data is first compressed to make space for the metadata. If for some reason the data in these blocks is not compressible, it is possible the metadata cannot be written to the LUN. This is not generally an issue, given that the Key ID is derived from the LUN volume serial number, thus it can be recreated. Most operating systems also leave the first 128 KB for volume information, making it even more likely that the first 16 blocks can be compressed.

For storage environments using data replication, the replicated LUN is a different LUN than the original LUN and thus has a different volume serial number. In the unlikely event that the key ID cannot be written to the original LUN, and the encrypted LUN is replicated to a different LUN, Brocade has implemented a technical element to handle this situation. When a new LUN is added for encryption, the encryption switch reduces the LUN size advertised to the host by a small number of bytes and places the key ID for the LUN in the unadvertised bytes at the end of the LUN. This specific compression technique has been implemented specifically for EMC, but it is also under consideration to be extended to other vendors' storage arrays.

PREPURCHASE VALUATION

The prepurchase evaluation phase for a Brocade encryption solution is likely the most critical phase of the entire lifecycle of the solution. It is imperative to properly qualify and size the encryption solution capabilities, then to ensure that the customer fully understands the implications of encrypting their data. An encryption solution can be quite complex, with many pieces interacting tightly. This requires the involvement of multiple disciplines to ensure appropriate purchasing decisions, solution design, installation and configuration, and ongoing management of the solution. Due to the implications of implementing a complex solution in a production environment and the large number of stakeholders involved in the decision phase of such projects, the acquisition cycle for an encryption solution can be relatively long, compared with simpler standard FC solutions.

Typically, the acquisition cycle for an encryption solution is divided into two distinct phases. The first phase involves making the decision to actually encrypt data-at-rest. The second phase, once it has been decided to encrypt, is to decide which solution to acquire that meets an organization's encryption requirements. The first phase generally involves the security, management, compliance, and other teams within an organization. Once Phase One is complete, the project is passed on to the technical teams (storage, network, application, and security) to discover the available solutions that meet customer requirements and to select the one that is the most cost-effective.

Why Encrypt Data-at-Rest?

One of the first questions to answer before proceeding with an encryption solution is: Why do you need to encrypt your data-at-rest? Reasons vary depending on the customer's particular requirements. Typically, the primary driving factor to acquire an encryption solution is to meet an industry or government compliance requirement. The second leading reason is to fulfill fiduciary obligations. This might mean simply protecting the company's goodwill or "brand" or, more importantly, ensuring that no data breaches occur (with resulting negative publicity). In some cases, customers see an open window of opportunity—such as constructing a new data center—to anticipate future regulations.

Comparative Overview of Encryption Solutions

The first step before acquiring an encryption solution is deciding which solution to deploy. There are several solutions available to customers, each with its own advantages and disadvantages. The key is to choose the ideal solution for the customer's specific requirements. Customers generally acquire an encryption solution to meet compliance requirements or out of due diligence. Typically, you can consider data contained within a restricted access data center to be safe without the need for further protection using encryption. There are, however, exceptions. For instance, some customers have specific confidentiality requirements within a secure data center—such as in a multihosted or multitenant environment where data from several customers, possibly competitors, may be physically co-located within the same data center.

You can implement encryption solutions in different places within a SAN, and with different tools. Encryption solutions generally fall into two broad categories: software-based and hardware-based. Software-based encryption solutions are typically less expensive than their hardware-based counterparts, but they may come with a 30 to 100 percent performance impact. Hardware-based encryption solutions have minimal or no performance impact on the production environment, but they generally come at a premium cost. Encryption solutions can be further subdivided by where they are implemented: on the host, as an appliance, on the storage device, or in a switch.

Host-based encryption is always implemented using software that must be installed on the host, which implies a loss of performance. Typically, performance is decreased by about 30 percent with these solutions, and sometimes more. There are no Host-Based Adapters (HBAs) with encryption capabilities available on the market. Since physical access to the equipment and use of specialized equipment is required to sniff data between the host and the fabric, it is also understood that there are much simpler attacks that can be performed, and at lower risk to the attacker. For these reasons, it is difficult to justify the supplemental cost of encrypting data between the host and fabric. Scalability is another important issue with host-based encryption. If only a few servers require encryption, then the cost can actually be quite reasonable. However, the costs increase proportionally with the number of servers requiring encryption. Furthermore, no host-based solutions offer the capability of encrypting existing data or rekeying previously encrypted data without having to first migrate the data to a different physical volume.

Application-based encryption is an excellent method of encrypting data, since it is truly end-to-end. That is, the data is encrypted directly by the application to its storage and remains encrypted until it is presented to the user in the application. The downside of this solution is that it is software-based and has a negative performance impact on the order of about 30 percent. Furthermore, this solution requires modification of the source code to add the encryption, with the necessary quality testing cycles once completed, which is performed as a consulting engagement and at great cost. This is generally only used for one specific application and does not scale very well to several applications. Furthermore, once the data is encrypted at the application level, the data is no longer compressible and cannot take advantage of other devices that compress and deduplicate data.

Backup solutions can also be grouped into this category. Most enterprise backup application vendors offer an encryption module to encrypt data to the backup media. The encryption process is built into the backup application software, but this method utilizes processing cycles on the backup server, resulting in a negative performance impact that increases the backup window. Reported performance impact varies based on the solution and whether compression is enabled or not, but typical performance impact falls anywhere between 50 and 100 percent.

Appliance-based encryption solutions use specialized hardware purposely designed to encrypt data. These solutions do not impact performance and have been used extensively in the past for this reason. The appliance needs to be introduced into the data path, which requires disconnecting devices from the production environment and reconnecting them into the appliance. Additionally, appliances require one port per storage device. For example, ten tape drives or ten connections to a disk array require ten ports on the appliance. The cost of this solution increases in proportion to the number of devices that require encryption. These solutions are rarely seen on the market, since all the major vendors have end-of-lifed their encryption appliances, with the exception of a few smaller vendors.

Storage hardware-based encryption includes tape drives and disk arrays. Tape drive encryption is a good solution for organizations that only need to encrypt smaller amounts of tape data. Some tape drives, such as LTO-4, LTO-5, IBM Jaguar, and Oracle T10000 drives, have built-in encryption capabilities. Some require an additional license to enable the encryption capability, while others do not. Since this type of encryption is a hardware-based solution, there is no noticeable performance impact using this solution to encrypt tape backups. On the other hand, if an organization also needs to encrypt disk data, it must purchase a different solution with a different key management solution to achieve this, resulting in two separate encryption solutions to manage.

Some disk arrays also offer built-in encryption capabilities. The HDS USP-V (and VSP-V) offers an optional encryption feature using specialized encryption controllers, with one encryption key per chassis. IBM and NetApp use the Seagate encrypting disk drives, with the encryption key hardcoded into the disk drive itself. Again, these solutions are hardware-based, resulting in no noticeable performance impact. However, not all disk array vendors offer encryption on their disk arrays and, within any specific vendor, not all models of disk arrays have built-in encryption capabilities. Usually, encryption is offered only on the high-end enterprise disk array model, while entry-level models do not offer encryption. Similar to the tape drive encryption, disk array encryption addresses only disk encryption; if an organization needs to encrypt tape data as well, this requires a second encryption solution.

The optional encryption feature usually comes at a hefty price, which needs to be factored into the total cost of the solution. In some cases, a performance impact is noticed when using the array-based encryption solution—another question that must be asked when evaluating such a solution.

Furthermore, none of the disk array vendors offer a viable method to perform a first-time encryption or rekey of existing data. These operations require a physical migration of data from one physical volume to a different physical volume.

Fabric-based encryption can be performed within a FC switch. There are only two vendors currently offering this solution, but with very different implementations. Cisco offers an encryption solution based on their application platform: the 9222i (and their blade equivalent). Essentially, Cisco has created encryption software to run on their generic application platform, an FC switch designed to run applications similar to the Brocade 7600 Application Platform. Since this is a software-based solution, there is a noticeable performance impact roughly on the order of 30 percent. With tape encryption, and the high compression ratio claimed by Cisco, some environments might possibly tolerate this performance impact, but for disk encryption it is unlikely that customers will be willing to accept any performance degradation of their applications. Additionally, the inability of Cisco to perform online and in-place first-time encryption and rekey operations places them at a significant disadvantage compared with Brocade encryption solutions.

The Brocade encryption solutions are purposely-designed hardware-based encryption solutions that use new hardware to eliminate any performance impact on the production environment once encryption is enabled. The Frame Redirection capabilities of the Brocade FC switches provide tremendous flexibility and scalability by eliminating the need to tie down a specific port on the encryption switch or blade to a specific storage port. In fact, you can have over 1000 storage ports being encrypted through a 32-port encryption switch. Another important feature of the Brocade Encryption Switch is the ability to perform a first-time encryption of data in-place without any disruption to the production environment. Once a LUN on a disk array is configured for encryption, the Brocade Encryption Switch reads the first logical block, encrypts it, and writes it back on the LUN exactly where it was located, while the LUN is still on production. Sequentially, The Brocade Encryption Switch then goes through the remaining logical blocks on the LUN until it has encrypted the entire LUN. There is no need to migrate the data from one LUN to another, as with all other disk array-based solutions. Similarly, a data rekey is accomplished by reading the logical blocks on the LUN, decrypting them, re-encrypting them using a different key, and writing the block back to the LUN exactly where it was. This is all performed online while production data is being written to the LUN; thus there is no need to migrate data as is the case with any other disk array-based encryption solution.

Table 1 summarizes some of the key characteristics of the various encryption solutions.

Table 1. Comparison of Encryption Methods

Encryption Method	Encryption Type	Cost	Taoe	Disk
Host-based	Software	$	No	Yes
Application	Software	$$$	Yes	Yes
Backup software	Software	$	Yes	No
Appliance	Hardware	$$	Yes	Yes
Tape drive	Hardware	$$	Yes	No
Disk array	Hardware	$$	No	Yes
Cisco FC switch	Software	$$	Yes	Yes
Brocade FC switch	Hardware	$$	Yes	Yes

Considerations for Export of Cryptographic Products

Until recently, cryptographic algorithms and materials were considered to be munitions, and as such fell under specific export regulations as dictated by each country. Although cryptographic material is no longer considered to be munitions, it is still subject to export regulations in the U.S.

In the U.S., the export of cryptographic material is controlled by the Department of Commerce Bureau of Industry and Security (BIS). Some countries, known as "rogue states," are strictly forbidden to export cryptographic material. For the most part, laws around exporting cryptographic material outside of these countries have been relaxed, but there still are some restrictions. It is best to verify with the BIS before exporting any cryptographic material.

Other countries also have restrictions on exporting or importing cryptographic materials. For example, France has a current import restriction on 128-bit keys, which are subject to special permission.

Qualifying the Solution

Given the complexity of encryption solutions and the diversity of equipment in production environments, it is critical to properly qualify the solution. Qualifying a Brocade encryption solution is accomplished by first completing the System Assurance form (an Excel spreadsheet), which can be obtained from the local Brocade Systems Engineer (SE). The account SE then verifies that the equipment listed in this form appears on the latest Brocade Interoperability Matrix for the encryption solution. If all equipment is supported, the Encryption System Assurance (ESA) form is submitted to the technical review team for final verification, and the Brocade professional services team is engaged to develop a statement of work for the services. Given the specialized expertise the encryption solution requires and the impact of a configuration error on a production environment, it is essential to use Brocade (or qualified OEM) professional services.

Some specific vendor products may not appear on the Encryption Interoperability Matrix, but, after review, the decision may be made to support a particular product as a Category 2 support. Category 2 support refers to equipment listed on the ESA form that is not on the Brocade Encryption Interoperability Matrix for the applicable Brocade FOS release, but that is similar to items that have been tested and are currently supported. In this case, Brocade support may be willing to support the items in question with confidence in their interoperability; however, if the items are not on site, the environment cannot be replicated for troubleshooting. If an issue does occur that requires troubleshooting on the hardware, the customer's assistance may be necessary, as well as access to their equipment to identify the cause of the issue and validate a fix through onsite testing. Items that require developing and testing of feature enhancements for full support may be prioritized based on urgency and availability of resources. A customer accepting deployment of a Brocade encryption solution, where some of the components in their environment are supported as a Category 2, need to accept that the resolution of support issues involving the Category 2-supported product may on some occasions take longer to troubleshoot.

Sizing the Solution

As part of the exercise of completing the System Assurance form, the sizing of the encryption solution needs to be established. The ideal way to size such a solution is by performing a detailed dataflow analysis to fully understand the average and peak bandwidth requirements. In reality, however, very few customers have this information on hand, and it is generally too costly and time consuming to perform this exercise. Alternative methods are generally used in lieu of a detailed dataflow analysis.

Sizing a solution for tapes is relatively easy and is performed by using the System Quality Assurance form tab for tape sizing. The model and number of tape drives is simply entered in the appropriate fields, and the spreadsheet calculates the bandwidth requirements.

The general rule of thumb for disk encryption is to think of an encryption device (switch or blade) as a "pool" of encryption processing capability that can be drawn upon by any device from anywhere. To simplify this process of estimation, assumptions have to be made regarding equal distribution of I/O across the different servers (which is seldom the case in the real world). Again, this is only an estimation process that is used in lieu of a sophisticated, lengthy, and expensive dataflow analysis.

For disk storage:

1. First, calculate the number of connections to the storage arrays.
2. Next, multiply this by the bandwidth of these connections. This provides the maximum bandwidth to the storage arrays.
3. Count the number of servers in the entire fabric.
4. Count the number of servers that require encryption.
5. Calculate the percentage of servers that require encryption.
6. Calculate the estimated required storage bandwidth.

NOTE: *Often this type of estimated calculation is fairly conservative, since most storage arrays cannot continuously deliver data equal to the bandwidth on the host ports (due to a limited number of disks or performance of the array controllers).*

From this, you can now estimate the number of encryption devices that are required for a production environment.

1. Calculate the number of encryption devices required per fabric to meet encryption requirements.
2. Count the total number of fabrics that will have encryption devices.
3. Multiply the number of fabrics by the total number of encryption devices per fabric.

Example 1:
There are 200 servers in a production fabric, of which 50 servers will require encryption. There are eight 8 Gbps connections between the fabric and the storage arrays. Encryption will be performed only at the primary data center in a dual-fabric configuration. Here is the estimated bandwidth requirement calculation:

1. Eight connections to storage arrays
2. 8 × 8 Gbps = 64 Gbps
3. 200 servers in production environment
4. 50 servers will require encryption
5. 50 / 200 × 100 = 25 percent of servers requiring encryption
6. 25 percent of 64 Gbps bandwidth to arrays = 16 Gbps

The estimated encryption bandwidth required for this environment is 16 Gbps. Given that the entry-level configuration of an encryption device is 48 Gbps, this solution will require only one encryption device per fabric.

1. 1 entry-level encryption device required
2. One data center with 2 fabrics = 2 fabrics
3. 2 fabrics × 1 entry-level encryption device = 2 entry-level encryption devices

Final Solution Size: 2 entry-level encryption devices.

Example 2:
There are 400 servers in a production fabric, of which 2000 servers will require encryption. There are 16 connections at 8 Gbps between the fabric and the storage arrays. There are two data centers, one primary data center and one for Disaster Recovery (DR), with two fabrics in each data center. Here is the estimated bandwidth requirement calculation:

1. 16 connections to storage arrays
2. 16 × 8 Gbps = 128 Gbps
3. 400 servers in production environment
4. 200 servers will require encryption
5. 200 / 400 × 100 = 50 percent of servers require encryption
6. 50 percent of 128 Gbps bandwidth to arrays = 64 Gbps

The estimated encryption bandwidth required for this environment is 64 Gbps. Given that the entry-level configuration of an encryption device is 48 Gbps, this solution will require more bandwidth. If the disk encryption performance upgrade license for the encryption device is purchased, the total available bandwidth on the encryption device will be 96 Gbps. In this environment, one encryption device with the disk encryption performance upgrade license per fabric will be sufficient.

1. 1 encryption device with disk encryption performance upgrade license is required
2. Two data centers with 2 fabrics each = 4 fabrics
3. 4 fabrics × 1 encryption device with performance upgrade = 4 encryption devices with performance upgrade license

Final Solution Size: 4 encryption devices with the disk encryption performance upgrade license.

Example 3:
There are 600 servers in a production fabric, of which 450 servers will require encryption. There are 20 connections at 8 Gbps between the fabric and the storage arrays. Only the servers in the primary datacenter with a dual-fabric configuration will require encryption. Here is the estimated bandwidth requirement calculation:

1. 20 connections to storage arrays
2. 20 × 8 Gbps = 160 Gbps
3. 600 servers in production environment
4. 450 servers will require encryption
5. 450 / 600 × 100 = 75 percent of servers require encryption
6. 75 percent of 160 Gbps bandwidth to arrays = 120 Gbps

The estimated encryption bandwidth required for this environment is 120 Gbps. Given that the entry-level configuration of an encryption device is 48 Gbps, this solution will require more bandwidth. If the disk encryption performance upgrade license for the encryption device is purchased, the total available bandwidth on the encryption device will be 96 Gbps. This still is not sufficient to meet the encryption requirements for this environment. This environment will require, at a minimum, at least one encryption device with the disk encryption performance upgrade license (96 Gbps) and one entry-level encryption device (48 Gbps), for a total of 144 Gbps of encryption capability—which is sufficient to meet the 120 Gbps estimated encryption requirement.

1. One encryption device with the disk encryption performance upgrade license and one entry-level encryption device is required
2. One data center with 2 fabrics each = 2 fabrics
3. 2 fabrics × 1 encryption device with the disk encryption performance upgrade = 2 encryption devices with performance upgrade and 2 entry-level encryption devices

Final Solution Size: 2 encryption devices with the disk encryption performance upgrade license and 2 entry-level encryption devices.

Encryption Switch vs. Encryption Blade?

For the encryption solution, the same rules apply to a generic Fibre Channel design with no encryption with regard to the decision of using blades or switches. When implementing this solution with encryption switches, ISLs are required to connect it to the production fabric. How many ISLs are required depends on the total amount of required encryption bandwidth. The advantage of using encryption blades is that you do not need to be concerned with ISLs, since all of the encryption is performed via the backplane. However, the encryption blades are available only for the Brocade DCX 8510 and the DCX family of Backbone products. If you have a Brocade 48000 Director, for example, then you need to obtain the Brocade Encryption Switch or upgrade to a Brocade DCX-class Backbone. Some architects prefer not to use up a premium Brocade DCX 8510/DCX/DCX-4S slot with only a 16-port blade, in favor of a higher density 32-port, 48-port, or 64-port blade. In this case, a Brocade Encryption Switch can be connected to FC ports in a higher density FC blade using ISLs. A less common, but viable, option is to pair an encryption blade with a Brocade Encryption Switch in an HA cluster.

High Availability

As discussed previously, there are two clustering configurations available with Brocade encryption solutions: the DEK clusters and the HA clusters. The DEK (data encryption key) cluster is used when you need to synchronize the keys between two encryption devices that may or may not be within the same fabric but that need to share the same keys. The HA (high availability) cluster is used when two encryption devices within the same fabric are in an active-passive failover configuration, where one encryption device can take over the load of the other following a hardware failure. Essentially, a heartbeat is exchanged between the two encryption devices. When an encryption device in an HA cluster loses the heartbeat from its HA cluster paired device, it fails over and takes over the load from the downed encryption device.

For the bladed version, the HA cluster can be implemented only when the encryption blade pairs reside in physically separate chassis in the same fabric. That is, one encryption blade must be in one Brocade DCX chassis, and the paired blade must be in a different Brocade DCX chassis. However, both Brocade DCX Backbones must still remain within the same fabric.

In reality, although the DEK cluster is quite common, the HA cluster is seldom deployed, due to the additional cost and the limited additional protection it offers. In a typical dual-fabric disk environment, one encryption device is deployed per fabric and offers a redundant configuration. In the event of a failure of an encryption device in one fabric, the multipathing software on a host simply redirects all traffic to the other path that is still functioning. With an HA cluster, a failure of an encryption device in one fabric does not result in a multipathing software failover, since all traffic directed to the failed encryption device is redirected to its HA cluster pair in the same fabric. The HA cluster is used only in the most demanding environments that are pushing the fabrics heavily and where a loss of one fabric will result in a decreased overall performance of the applications. Some environments, such as outsourced environments, may incur penalties if a guaranteed level of service is not maintained. Thus, after a cost analysis, it is found to be less expensive to implement an HA configuration (at almost twice the cost for the double encryption infrastructure) than to pay the stated penalties for a decreased level of service.

When using a single fabric for tape backup encryption, HA clustering can be one method to ensure that backup Service Level Agreements (SLAs) are met in the case of a Brocade Encryption Switch or blade outage.

Cost Considerations

The overall cost of the encryption solution may become an important factor in more cost-sensitive environments. Brocade created the entry-level version of the encryption devices specifically to make the price of an initial deployment more attractive to these more cost-sensitive customers. Quite often, customers have an initial encryption requirement for a smaller number of applications. Over time, once the solution has been in production for some time, other servers and applications are eventually encrypted and use up more encryption processing power on the encryption devices. Once the entry-level throughput (48 Gbps) of the encryption devices is exceeded, a simple license upgrade with the disk encryption performance upgrade license will increase the processing power of the encryption device to the full 96 Gbps, to provide twice the original throughput.

Some production environments may use an asymmetrical configuration. For example, a two-site environment, with a main production data center and a secondary DR data center, may have one full bandwidth encryption device per fabric at the main data center but only one entry-level encryption device per fabric at the DR site (since not all encrypted data may be replicated across sites). Some customers with a cold DR site have even implemented this solution with a single encryption device at the DR site, since the disk arrays at that site have only a single fabric. In this case, the customer was willing to accept a possible performance degradation and additional vulnerability of having a single fabric, in order to maintain costs at a reasonable level. This configuration is not generally recommended, but it demonstrates the flexibility of this solution based on specific customer requirements.

Solution Interoperability

Since the Brocade encryption solutions were first deployed, the number of successful installations and diversity of products within these installations has increased considerably. The encryption solution interoperability matrix is now quite extensive and includes most major storage vendor products on the market, as well as many legacy storage products. The Brocade Encryption Interoperability Matrix is available on www.brocade.com. A Proof-of-Concept (POC) may be performed under certain circumstances; all POCs must be arranged with the Brocade security product management team.

DESIGN AND ARCHITECTURE CONSIDERATIONS

Proper design of the encryption solution involves considering the performance, management, availability, and cost characteristics required by each individual production environment. Generally, improvements on the first three characteristics have an impact on the fourth one—the cost of the total solution—and the encryption solution architect must weigh the importance of each of these characteristics to maintain costs at a reasonable level.

The Brocade Encryption Switch, like any other security product, does not come fully configured out of the box. It must be configured properly and be part of a well-designed architecture with the appropriate operational procedures, to ensure continuous and secure operation. This section outlines some best practices for the design and implementation of Brocade encryption solutions.

Encryption is only one component of a comprehensive SAN security program. An organization may have the best encryption solution possible, but if it is installed on a SAN with security holes, then the entire solution may be vulnerable. In security, a system is only as strong as its weakest link, which is usually the place attackers will target first.

Availability Considerations

As with any IT solution, there are many ways to ensure availability. Choosing the best method to maintain availability depends on the value of the information (and the impact of a loss of availability), the risk and probability of disruption, and the cost of implementing high availability.

Clustering Encryption Devices

Clustering is a commonly used method to ensure protection against hardware failure. As explained in a previous section, there are two types of clusters for Brocade encryption solutions, which can be used independently or simultaneously. The High Availability (HA) cluster provides hardware redundancy for the encryption devices. The Data Encryption Key (DEK) cluster allows two or more encryption devices to share the same keys across fabrics.

For tape encryption using a single fabric, since tape drives are single-attached devices (actively attached devices), a single encryption device might be sufficient. However, some organizations consider the backup application as mission-critical or high priority, due to a Service Level Agreement that must be respected. If this is the scenario, a business case can be made to justify the use of a second encryption device within the same fabric to form an HA cluster.

For disk encryption using a dual-fabric configuration, the minimum requirement is for one encryption device per fabric. In the event of a failure of one encryption device, the MPIO (Multi-Pathing I/O) software on the host

automatically fails over the traffic to the remaining path. This may result in degraded performance in some heavily used systems, which may or may not be acceptable. If this is not acceptable, then a second encryption device must be added to each fabric to form two HA clusters.

For redundancy, it is a common practice to implement more than one path from the disk storage device to the fabric. If more than one path exists in the same fabric from a host to a LUN, then it is important to use Brocade FOS v6.3 or later when performing a first-time-encryption or a rekey operation. Multipath rekeying operations through a single encryption engine are not supported prior to Brocade FOS v6.3.

Dual Sites

Production environments with dual sites, such as a primary production site and a secondary disaster recovery (DR) site, have specific requirements that vary depending on what data needs to be encrypted. Since a security solution is only as strong as its weakest link, it is essential to encrypt the data at all locations if data is replicated. For example, if data is encrypted at the primary data center, but the replicated data at the DR site is not encrypted, a knowledgeable attacker will simply target the cleartext copy of the data at the DR and bypass the encrypted data at the primary site.

It is not recommended, and in many cases not likely supported, to place the encryption devices at two different physical locations in the same encryption group. The distance between the two sites can result in additional latency, and a faulty or noisy connection between the two sites may result in a loss of IP connectivity between the encryption devices at the different sites. This is problematic, since the creation of new LUNs requires synchronization between the two sites and may result in a degraded encryption group.

When configuring two sites with different encryption groups, but with shared keys, it is important to make sure the master key is the same for both encryption groups, to allow for decryption of the DEK created by the encryption devices in other encryption group.

BEST PRACTICE:

- Do not place encryption devices at physically different sites or locations in the same encryption group.

Redundant Key Vaults

Key vaults can also be configured in a clustered configuration to provide redundancy. Each key management solution vendor offers different features and functionality around clustering, but most of them provide some form of clustering capability. Although clustering the key vault is an optional feature, it is certainly recommended as a best practice. Ideally, a key vault should be located in at least two separate locations to provide the maximum redundancy. Furthermore, the keys cached in an encryption device in one location should be backed up to a key management device in a different location. For example, if the primary data center where the encryption device resides experiences a major catastrophe, and the entire site is destroyed, the keys should be available on the key management device at a different location (perhaps the DR site) to recover the keys needed to recover encrypted data.

Performance Considerations

As explained earlier, the latency of the Brocade encryption devices is negligible compared to the time it takes to complete an I/O operation. However, a complex fabric may have multiple ISLs and offer many paths between the various devices within the fabric. As discussed earlier, the frame redirection feature can automatically redirect frames to the encryption device, regardless of where it is located in the fabric. However, certain locations for the encryption devices offer the best performance.

The basic concept of locality applies to the encryption solution as well. Locality simply states that a host and its storage devices should be located as close to one another as possible given a specific architecture. For example, the highest locality occurs when a host and its storage device are connected to the same switch in a fabric or the same blade in a director or backbone. Essentially, SAN placement of the encryption devices should be done as close as possible between the host and its storage devices.

To avoid forcing traffic to pass through ISLs, a backbone can be used to consolidate multiple switches. The Brocade FS8-18 Encryption Blade in a Brocade DCX 8510, DCX, or DCX-4S does not require ISLs to perform the encryption, and all the traffic to be encrypted passes through the backplane.

Although it is not necessary to connect devices (hosts, disk arrays, or tape drives) involved in the encryption process directly into the encryption device, there may be some cases where it makes sense to do so. The general rule here is similar to designing regular FC fabrics without encryption, and it involves locality. Locality simply means to place devices that talk to each other as close to each other as possible to reduce the distance or number of paths/hops between them. In some cases, you may consider connecting a host running a very I/O-intensive application on the same switch to which its storage is connected. Similarly, a very I/O-intensive application that is encrypted can be connected directly to the encryption device as well as its storage device. The concern here is not so much the latency or delay with the distance and hops but rather simply to avoid potential bottlenecks resulting from heavily used or saturated paths and to avoid contention for these paths with other applications.

Another performance question that comes up occasionally is whether it makes a difference if devices are connected on specific ports on the encryption switch or blade. For example, should the devices or ISLs involved in the encryption process be spread out evenly across quads (4-port clusters on switch or blade), or can you use up all four quads anywhere on the switch? Due to the internal architecture, it does not matter where the devices or ISLs are connected. They can all be connected starting from the first physical port and added sequentially to the other adjacent ports, or they can be spaced out, if that is preferred. The choice of physical ports has no impact on the performance of the encryption solution.

Deduplication and Compression with Encryption

Encryption is an application that does not co-exist with deduplication and compression applications. In the case of compression, this application looks for patterns in data, white space, and so on that are optimized by the compression algorithm, resulting in less data traveling through a link or stored on a storage medium. Encryption, on the other hand, randomizes data and essentially removes all patterns or converts white space into a random combination of 1s and 0s, making it impossible for the compression algorithm to optimize this data.

This inability to compress data should be considered in advance for organizations that use compression between two data centers to reduce the amount of data, and subsequently the data communications charges, with their provider, which are exchanged between the two data centers. To illustrate this, consider the example of an organization replicating data between two data centers over a high-speed IP connection using an FCIP tunnel and using a compression appliance at both ends to reduce the total amount of data being sent between the sites. If the data at the primary data center is encrypted, then encrypted data is replicated and sent across the FCIP tunnel to the other data center. In this case, since the data being replicated is encrypted, it is not compressible, and the amount of data transferred between the two sites is not optimized. For this reason, when a Brocade encryption solution compresses data destined to a tape drive, it first compresses the data and then encrypts it—in that order. Furthermore, if you simply encrypt the data and send it to a tape drive, the native compression capabilities of the tape drive are no longer able to compress the data, since it has been randomized through encryption and no longer contains patterns or white space that can be optimized. In the case of disk-based data that is replicated across two data centers, any compression that is implemented between the two sites is no longer effective when replicating volumes containing encrypted data.

Deduplication applications work in a similar fashion as compression, in that the deduplication engines search for identical files or perhaps identical blocks on storage devices and use pointers, which take up much less space than the actual repeated data. With encryption, two identical files or blocks of data look completely different once they have been encrypted, which prevents the deduplication algorithm from optimizing this data. The only way deduplication applications can remain functional with encryption is when the data passes through the deduplication engine first, gets optimized, then is sent through the backend and encrypted at that point.

In reality, though, it is not common for an organization to encrypt all of their production data. In fact, only the sensitive data that must be protected to meet specific compliance requirements or that is particularly important or sensitive to the organization gets encrypted. Subsequently, if an organization already uses or plans to use a

deduplication device in their production environment, only the data that is encrypted will no longer benefit from the deduplication. However, the remainder of the data in the production environment is unaffected. The same argument can be made with compression: Since not all production data will be compressed, the impact on the production environment may be minimized.

Cost Considerations

As mentioned previously, the cost of the total solution increases as performance, availability, and simplicity of management is increased. Generally, most production environments have different requirements, and most do not require the highest requirements. More often, the minimum requirements suffice. At a minimum, a single encryption device per fabric and a single key manager is required. Any additional hardware or software increases the costs of the total solution. For a disk environment with dual paths, at least two encryption devices (one per fabric) are required. For a tape environment where all tape drives hang off of one fabric, a single encryption device suffices. For a tape environment where tape drives are distributed over two fabrics, two encryption devices (one per fabric) are required. However, in the latter scenario, if not all backups will be encrypted, then it is possible to assign only the tape drives on one fabric for encryption, while the tape drives on the other fabric are used for cleartext backups. In this case, only one encryption device is required in the fabric that is used for encryption. In some cases, as mentioned in a previous section, customers have acquired a single fabric solution for a cold DR site. In this case, only a single encryption device is required for this configuration.

Since most production environments do not encrypt every single LUN in their environment, but rather they select the ones that contain more sensitive data, the 96 Gbps bandwidth for an encryption may greatly exceed the initial encryption bandwidth requirements. For this reason, Brocade has created an entry-level version of the encryption device to make it more accessible for price-sensitive environments. The entry-level encryption device works just the same as the full-bandwidth version, and every physical FC port on the device remains functional. However, the available encryption bandwidth has been reduced to 48 Gbps. If later on, as more devices or more data are being encrypted, the 48 Gbps limitation is reached and more bandwidth is required, a disk encryption performance upgrade license may be purchased and installed to obtain the full 96 Gbps bandwidth. It is important to note that the disk encryption performance upgrade license applies only to disk encryption, since the maximum encryption bandwidth for tape encryption cannot exceed 48 Gbps to begin with. If a production environment requires concurrent tape and disk encryption, a disk encryption performance upgrade license might be beneficial. However, the user will not be able to reach the throughput that could be achieved in a disk-only environment.

To reduce costs, some suggest deploying a single encryption device in a dual fabric configuration. Clearly, you cannot have one path in a dual fabric configuration that is encrypted and the other path in cleartext, since both paths lead to the same storage and ultimately the same LUN. To address this issue, the same people suggest cross-connecting the encryption device to the other fabric. Although this is technically possible, it is not recommended. In fact, connecting the encryption device from one fabric with a switch in another fabric results in one merged single fabric and the subsequent loss of redundant, independent dual paths.

Other Considerations

Virtual Host Considerations

Virtual hosts have become ubiquitous in enterprise-level data centers. They offer more effective usage of available server resources by sharing the hardware with multiple applications that run independently on virtual servers configured on the physical server. In this configuration, virtual hosts are assigned a unique virtual WWN that is different from that of the physical server. In the case of the encryption solution, since it is based on frame redirection technology that operates on source and target WWNs, the redirection process does not make the distinction between a physical or virtual WWN, and it has no impact on virtual servers.

The main considerations involve how the storage is presented to the Virtual Machine (VM). VMware offers two methods of presenting storage to the VM:

- VMFS—Virtual Machine File Systems via data store
- RDM—Raw device using Raw Device Mapping

In the case of VMFS (Figure 16), the storage presented to the virtual server is a file on the data store. This means that the LUN represented in the data store is created on the storage device, where it is subsequently presented to the ESX server. Once the ESX server and the storage ports are defined in the CTC, the LUN can be encrypted. At that moment, every VM sees storage that is part of that data store, and all data presented to the virtual server is encrypted. Although the virtual server sees the storage as though it is on a physical disk, it resides on the data store.

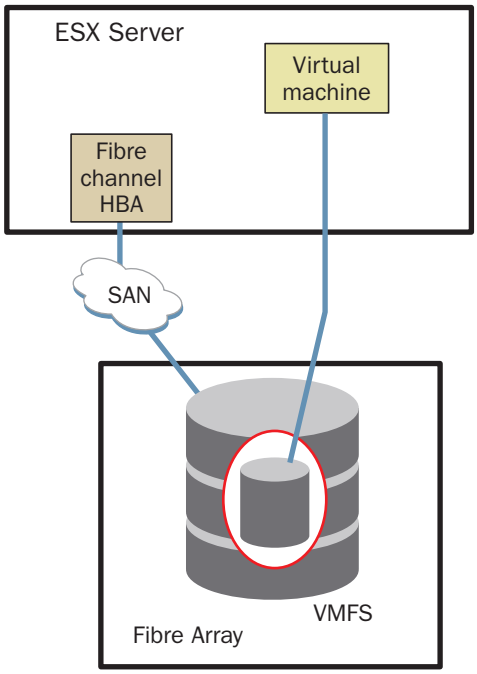

Figure 16. VMFS representation of data store.

For RDM (Figure 17), there is a direct relationship between the VM and the storage presented to the server. The VM is mapped to a special file located on a VMFS system, where pointers to the raw device can be found.. In effect, the CTCs need to be created in every fabric containing the storage ports and need to use the port World Wide Name (pWWN) of the ESX server HBA. When LUN masking is performed on the storage device, the LUN is presented to the pWWN of the physical ESX server HBA pWWN. The data store LUNs become visible only to the ESX server and the VMs. The VM simply exists as another file on the data store, where it receives virtual storage from the data store.

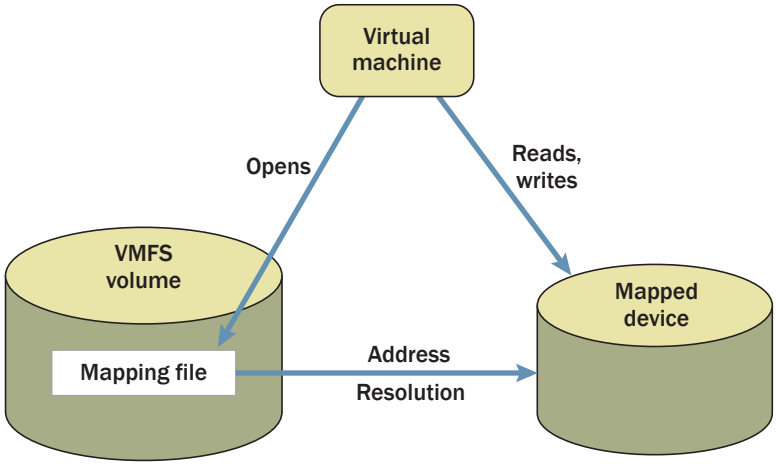

Figure 17. RDM representation of disk data.

Figure 18 illustrates a VMware ESX server with multiple VMs with a dual-path configuration to storage using independent fabrics.

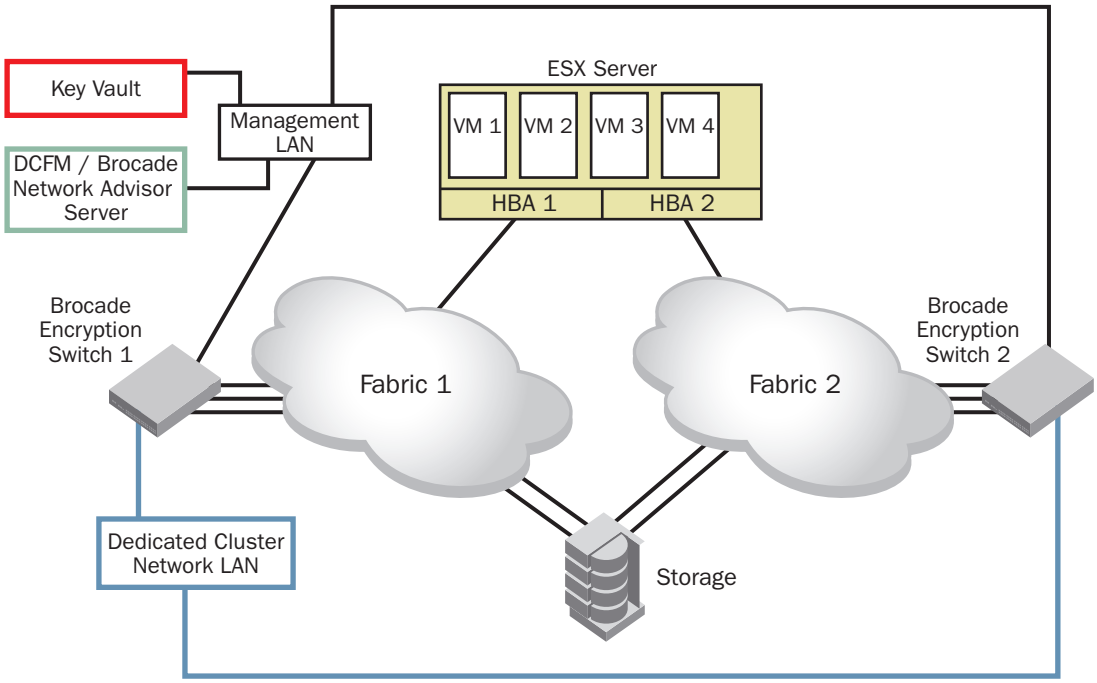

Figure 18. Dual-path connection between ESX server and storage.

Key Management

Key Expiration

Part of managing keys is determining how long a key should exist. Many organizations never expire a key, while others require expiration every six months. There is no general rule as to the frequency of key expiration; it depends entirely on business requirements and tolerance to the risk of a 256-bit key going stale. Since an online rekey operation can affect application performance, and an offline rekey requires downtime, most organizations would rather not perform a rekey too often. Generally, it is considered safe to expire 256-bit keys somewhere between every two to four years.

Key per Media vs. Key per Pool

For tape encryption, a single DEK can be assigned to one tape media or to an entire pool of tapes. Best practice is to have one DEK per tape media. In the event that the DEK is compromised, it is much simpler to create a new backup encrypted with a new DEK for one tape, as opposed to recreating an entire pool of tapes with a new DEK.

Certificates

Key managers and the Brocade encryption device must authenticate with each other before being able to exchange key information between them. This is done using a digital certificate or Key Authentication Certificate (KAC), and each key management solution vendor has its individual methods to perform this operation. In some instances, when a POC is performed, a temporary certificate may be used during the POC. However, it is important to replace this certificate with a permanent certificate when deploying the solution for production. Temporary certificates usually expire after a specified amount of time—usually about 365 days.

NetApp, specifically, has issued some certificates that expire after three years. In this case the certificates must be reissued, and a reauthentication between the encryption device and the key manager must be performed.

As a preventative measure, you can verify the expiration date for a KAC on a Brocade encryption device by using the following commands as the root user.

Switch_A:root> cryptocfg –show -nodecerts

Certificate for KAC, label: , format: PEM, file: /etc/fabos/certs/mace/pluto160.pem

Certificate for Primary Key Vault, label: F5RKM, format: PEM, file: /etc/fabos/certs/mace/cacert.pem

Switch_A:root> openssl x509 –in /etc/fabos/certs/mace/pluto160.pem –noout -dates

Not Before: Dec4 18:03:14 2009 GMT
Not After : Dec 4 18:03:14 2012 GMT

The output from the command above indicates that the certificate on this switch will expire on December 4, 2012.

BEST PRACTICE:

- Use a Certificate Authority (CA), so you will be reminded before a certificate expires. Otherwise, use a self-generated openSSL certificate.

Key Replication

The different vendors of key management solutions offer different methods to replicate the keys on the key manager (such as NetApp LKM and EMC DPM), while some (such as Thales TEKA/TEMS and IBM TKLM) do not have built-in replication services and require a manual procedure to do so.

BEST PRACTICE:

- Ensure you perform periodic backups of your DEKs if inter-site replication has not been automated.
- Periodically verify that the replication process is working for replicated environments.

Administrative Security Measures, Policies, and Procedures

Some often-overlooked components of the encryption solution are the administrative security measures, policies, and procedures surrounding the solution. The saying is often overused that security is as strong as the weakest link, but it holds quite true with this solution. For example, encrypting data helps maintain the confidentiality of the data on the media. However, if all system administrators have the ability to destroy keys, they can essentially destroy the data. It is important to look not only at the technical security measures around security but to also consider the human effect on security. Best practices could include the use of Role-Based Access Control (RBAC) and authentication methods to limit and restrict access.

Key Management Considerations

From the perspective of the Brocade encryption solutions, the key management solution really behaves like a key vault. Since all keys are created by the encryption device and can be destroyed or expired by the encryption device, the key management functions are performed by the encryption device—whereas the key management solution is utilized only as a vault to securely store DEKs. Specific key management solution vendors also offer specific features, such as the ability to replicate keys between more than one appliance. Some make use of master keys, while others do not. It is beyond the scope of this document to outline the different features and functionalities of the different key management solutions. The onus rests upon the customer to evaluate the different key management vendor solutions and understand the features and caveats associated with each one.

Key management solutions may also come in a software version, which is installed on a generic server, or as a self-contained hardware appliance. The software versions of key management solutions may be less expensive than their hardware appliance counterparts, which results in a lower total cost of the encryption solution. However, the software version of key management solutions cannot meet FIPS validation requirements. Having said that, not all key management appliances are FIPS validated, and the level of validation may vary from one vendor to another.

A few general key management guidelines and best practices are as follows:

- Ensure physical and logical security of the key management appliances/servers.
- Ensure that at least one key management appliance/server is located in a physically different location as the primary copy of the production data.
- Use the key management vendor's approved method to safely back up the keys.

DEPLOYMENT CONSIDERATIONS

This section discusses the best practices recommended by Brocade when deploying the Brocade encryption solutions as an initial install. Once configured and operational, the management of these solutions is described in the next section on management best practices.

Virtual Fabrics

Virtual fabrics provide the ability to partition a physical FC switch into multiple logical switches to isolate devices from others. There are several caveats with the virtual fabric feature and the Brocade encryption devices. The Brocade Encryption Switch does not support the virtual fabrics feature, but it can be connected to other switches that use this feature. The Brocade FS8-18 Encryption Blade does support virtual fabrics, but it can only be a part of the default switch partition in the Brocade DCX 8510/DCX/DCX-4S Backbone. Other virtual fabrics can still use the encryption service from the default switch, using FC routing to redirect data from the virtual fabric to the encryption blade and back into the virtual fabric. However, the administration of the service across virtual fabrics is still shared by a common group of administrators.

Management Interface Considerations

Managing and configuring Brocade encryption solutions can be performed either with the Brocade FOS Command Line Interface (CLI) or Brocade Network Advisor. As a best practice, it is highly recommended you use Brocade Network Advisor. The CLI requires many more steps to perform operations that can be done with one click in Brocade Network Advisor, and having to type many commands increases the risk of typing errors. Also, the Brocade Network Advisor interface provides wizards that guide you through the configuration process, to further reduce the risk of errors introduced as a result of improper sequencing of commands.

The management interfaces should never be accessed using unsecure protocols, such as Telnet for the CLI or HTTP for Brocade Network Advisor. Use secure protocols, such as SSH instead of Telnet and HTTPS instead of HTTP, and block or disable their equivalent unsecure services.

For additional protection, the System Card or system key feature should be implemented and a Smart Card required, to enable the encryption capability of the switch. This prevents someone who steals both the switch and the disk media from being able to decrypt the data on the storage media. Of course, it is also important to store the System Card in a secure location away from the encryption switch and storage media. Furthermore, it is recommended to backup the System Card and store the backups in a safe location.

Since the management network is used for encryption node communication with the key management system, it is recommended to separate the network (logically or physically) from other networks to ensure reliability of the management network and decrease potential impact on the encryption solution.

Quorum Authentication

To prevent one single person from having complete control over sensitive operations on the encryption device, the quorum authentication feature may be enabled on the management interface. Quorum authentication forces one or more persons to authenticate or “approve” the use of certain commands on the encryption devices using Smart Cards assigned to authorized persons. Certain sensitive operations can be forced to use a quorum authentication mechanism to allow them to be performed. As an optional feature, you can set quorum authentication for the management of the Brocade encryption devices using an external Smart Card reader installed on the management server. Once installed, the quorum authentication can be configured to require a quorum to perform specific operations. The quorum can be set using a minimum of one card and a maximum of ten cards.

The following is a list of the specific operations that can be controlled under quorum authentication:

- Master key generation, backup, and restore operations
- Control of activation of encryption engines
- Control of user access to the management application security administrator roles
- Replacement of authentication card certificates
- Enabling and disabling the use of system cards
- Changing the quorum size for authentication cards
- Establishing a trusted link with the key manager
- Decommissioning LUNs

Using Authentication Cards

When a quorum of authentication cards is registered for use, an **Authenticate** dialog box is displayed to grant access to the following:

- The Encryption Group Properties dialog box Link Keys tab.
- The Encryption Group Properties dialog box Security tab, which provides access to the following:
  - **Master Key Actions**, which includes **Backup Master Key**, **Restore Master Key**, and **Create Master Key**
  - The **System Cards** radio buttons used to specify whether or not a system card is **Required** or **Not Required**
  - The **Authentication Card Quorum Size** selector
  - The **Register from Card Reader** and **Register from Archive** buttons
- The **Master Key Backup** dialog box
- The **Master Key Restore** dialog box
- The **Decommission LUNs** dialog box

Role-Based Access Control

Brocade encryption solutions can be managed using specific roles assigned to restrict the operations that individual users and administrators can perform. There are two separate sets of RBACs available to manage the encryption solutions: one set for the CLI and another for Brocade Network Advisor/Brocade Data Center Fabric Manager (DCFM).

Table 2. Encryption user privileges

Encryption Method	Encryption Type
Storage Encryption Configuration	• Launch the Encryption Center dialog box. • View switch, group, or engine properties. • View the Encryption Group Properties Security tab. • View encryption targets, hosts, and LUNs. • View the LUN-centric view. • View all rekey sessions. • Add/remove paths and edit LUN configuration on the LUN-centric view. • Rebalance encryption engines. • Clear tape LUN statistics. • Create a new encryption group or add a switch to an existing encryption group. • Edit group engine properties (except for the Security tab). • Add targets. • Select encryption targets and LUNs to be encrypted or edit LUN encryption settings. • Edit encryption target hosts configuration. • Show tape LUN statistics.
Storage Encryption Key Operations	• Launch the Encryption Center dialog box. • View switch, group, or engine properties. • View the Encryption Group Properties Security tab. • View encryption targets, hosts, and LUNs. • View the LUN-centric view. • View all rekey sessions. • Initiate manual rekeying of all disk LUNs. • Initiate refresh DEK. • Enable and disable an encryption engine. • Zeroize an encryption engine. • Restore a master key. • Edit key vault credentials. • Show tape LUN statistics.
Storage Encryption Security	• Launch the Encryption center dialog box. • View switch, group, or engine properties. • View Encryption Group Properties Security tab. • View the LUN-centric view. • View all rekey sessions. • View encryption targets, hosts, and LUNs. • Create a master key. • Back up a master key. • Edit the Smart Card. • View and modify settings on the Encryption Group Properties Security tab (quorum size, authentication card list, and system card requirement). • Establish link keys for LKM and SafeNet (in SSKM mode) key managers. • Show tape LUN statistics.

NOTE: *For up-to-date information on RBACs, refer to the latest Encryption Administrator's Guide.*

Disk Storage Considerations

Data can be encrypted on disk storage at the LUN level. One single key is used to encrypt the data on a LUN, except during a rekey operation, which requires two keys. LUNs on a disk array are discovered through the standard SCSI LUN Discovery process.

Thin-Provisioning

Thin-provisioning is used to optimize the use of disk space by assigning only smaller chunks of disk space for data, while reserving additional extents of disk space for future expansion as required. Thin-provisioning can coexist with encryption, but there are some caveats.

Brocade supports encrypting certain thinly-provisioned arrays with Brocade FOS v7.1 and later. For these arrays, logical LUNs do not expand to the fully provisioned disk capacity, and only written blocks are subject to First-Time Encryption (FTE) or rekey operations. Refer to the Brocade FOS release notes, Encryption Interoperability Matrix, and Administrator's Guide available on www.brocade.com for additional thin provisioning support information. Note that it is important to make sure that the OEM vendor that owns the support contract for the encryption solution also supports encrypted thin-provisioning. Thin-provisioning is complex and should be carefully verified with all parties concerned as to whether it will be supported in that particular environment.

For arrays not supported with thin provisioning, a thin-provisioned production LUN with existing data may be encrypted using Brocade encryption solutions. However, the current disk space assigned to the LUN, as well as the reserved extents, will be encrypted during the first-time encryption process. This will result in essentially nullifying the advantages of thin-provisioning. In reality, of course, since most organizations encrypt only a portion of their data, this will only affect the encrypted LUNs; the remaining LUNs will still benefit from thin-provisioning. The requirement to encrypt sensitive data usually outweighs the desire to temporarily under-provision physical disk space.

On the other hand, if you create a new thin-provisioned LUN with no data and configure this LUN for encryption before writing data to it, the thin-provisioning will be effective. In this case, since there is no need to pre-encrypt or convert existing data on the LUN, new writes to the LUN will be encrypted as they are written to the LUN.

Another caveat with thin-provisioning is when a LUN is encrypted using the -newLUN option when used with an EMC DPM key management appliance. This option allows the metadata to be written in the last three blocks of a LUN instead of the first 16 blocks of a LUN. This is generally used when the first 16 blocks of a LUN are not compressible, and it is not possible to write the metadata. Instead, the metadata is written in the last three blocks of the LUN, and the size is presented to the host as its true size minus three blocks. In the case of thin-provisioning, if an original LUN with the metadata written at the end is subsequently expanded, the metadata is no longer at the end of the LUN, and the Brocade encryption device is no longer able to retrieve the metadata. Different storage vendors with thin-provisioning offers may also behave differently than what is described here.

Another important limitation of encrypting thinly-provisioned disks is the loss of the rekey capability. A rekey operation results in encrypting the entire LUN, including the unused but reserved extents.

Remote Disk Replication

Remote disk replication generally involves copying data from one LUN to another LUN over distance, in order to make a second copy available in the event of a lack of accessibility to the original LUN. Typically, remote data replication is implemented between a primary data center and a secondary or DR site. The procedure below was originally designed for EMC Symmetrix Remote Data Facility (SRDF) replication (as discussed in the next section), but the same procedure can be applied. The -newLUN option is not necessary for replicated LUNs not using SRDF, but it is nevertheless a good practice.

Local disk replication (within the same array or within the same data center), sometimes called snapshots, mirrors, or clones, can also be performed with the encryption solution. Essentially, local disk replication should not pose any problems with Brocade encryption solutions, but you should always verify this with Brocade before proceeding.

When the metadata and key ID are written, the primary metadata on blocks 1 to 16 is compressed and encrypted. However, there are scenarios where these blocks are not compressible, and the metadata is not written to the media. If blocks 1 to 16 are not compressible on a local source device, and metadata is not written, obtaining the correct DEK for the remote target device becomes problematic. This problem is avoided by reserving the last three blocks of the LUN for a copy of the metadata. These blocks are not exposed to the host initiator. When a host reads the capacity of the LUN, the size reported is always three blocks fewer than the actual size. The behavior is enforced by setting the -newLUN policy parameter option on the cryptocfg –add -LUN command. This option allows the metadata to be written in the last three blocks of a LUN in addition to the first 16 blocks of a LUN. Note the following when using the -newLUN option:

- The -newLUN option is used only if an EMC DPM key vault is configured for the encryption group. The possibility of using this option is being investigated for other key vault platforms as well.
- The -newLUN option can be used only if replication is enabled for the encryption group using the cryptocfg –set -repl command.
- Both LUNs that form an SRDF pair must be added to their containers using the -newLUN policy parameter option.
- At any site, all paths to a given SRDF device must be configured with the -newLUN option.
- All LUNs configured with the -newLUN option will report 3 blocks fewer than the actual size when the host performs READ CAPACITY 10/READ CAPACITY 16.
- If a LUN is added with the -newLUN option and with encryption enabled, it will always have valid metadata, even if blocks 1–16 of the LUN are not compressible.
- LUNs configured as cleartext must also be added with the -newLUN option if they are part of an SRDF pair. This is to handle scenarios where the LUN policy is changed to “encrypted” at a later point in time and to verify formation of DEK clusters and LUN accessibility prior to enabling encryption for the LUN. When cleartext LUNs are configured with the -newLUN option, no metadata is written to the last 3 blocks, but it will still report 3 blocks fewer than the actual size when the host performs READ CAPACITY 10/READ CAPACITY 16.
- If the local LUN contains host data, configuring it with the -newLUN option causes the data on the last 3 blocks of the LUN to be lost. Before using the -newLUN option, you must migrate the contents of the LUN to another LUN that is larger by at least 3 blocks. The new larger LUN can then used when creating the SRDF pair. The remote LUN of the SRDF pair must be of the same size. The original smaller LUN with user data can be decommissioned.
- When the -newLUN is specified, the -keyID, -keylifespan, and -enable_rekey operands cannot be used.

Disk Replication with SRDF

EMC Symmetrix Remote Data Facility (SRDF) products provide data replication between Symmetrix storage arrays through a Storage Area Network (SAN) or IP network. Logical pairs of LUNs (storage) or groups of LUNs replicate data from each other synchronously or asynchronously. SRDF also allows established LUN pairs to be split, so that separate hosts can access the same data independently for backup and other purposes and then can resynchronize.

Symmetrix arrays running SRDF are frequently found in large, enterprise-class environments, such as those targeted by high-performance Brocade data-at-rest encryption products. Brocade Engineering has tested EMC SRDF with Brocade encryption products and supports environments running SRDF. To ensure reliable operation, Brocade provides specific deployment and operational guidelines. Failure to follow these guidelines may result in potential interruptions to data recovery or data corruption.

Refer to Figure 19 for the following guidelines:

- During rekey or First-Time Encryption (FTE) of a source LUN (R1) at Site A, a replication or snapshot operation must be stopped or paused. Host access at the source LUN R1 (Site A) and remote LUN R2 (Site B) is still allowed at both sites during the rekeying operation.

NOTE: *To ensure that SRDF operation has stopped prior to initiating rekeying, do not use Auto-Rekey in TimeFinder or SRDF environments. Not configuring Auto-Rekey prevents an unanticipated Auto-Rekey operation from initiating. Instead, perform data rekey operations in TimeFinder or SRDF environments within a scheduled maintenance window, when the rekeying option can be manually selected.*

- After the rekeying or FTE is complete for the source LUN R1, host access for the remote LUN R2 at Site B must be prevented until the source LUN R1 is synchronized with remote/destination LUN R2. After the rekeying of source LUN R1 is complete, the replication or snapshot operation can be resumed from source LUN R1 to remote/destination LUN R2.
- Once all the data from source LUN R1 is replicated to the remote LUN R2, then the remote LUN R2 must be removed from the Crypto-Target Container (CTC) on the Brocade encryption device and added back with "lunstate=encrypted." This enables a refresh of the new DEK for the LUN following the rekeying process. Resume the host access to LUN R2 at Site B when the LUN goes to the "Encryption Enabled" state.

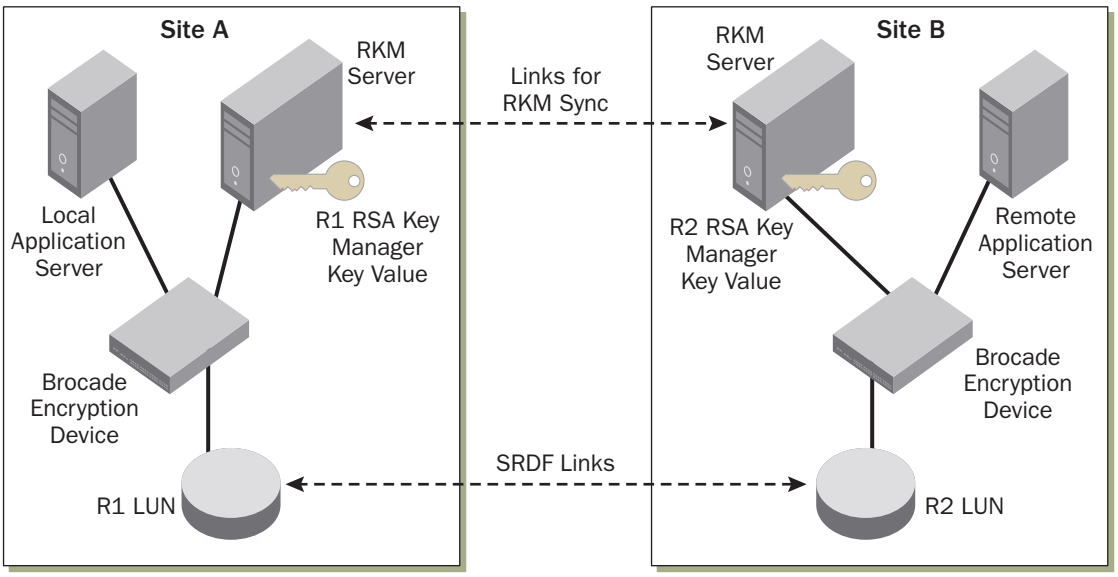

Figure 19. Local and remote site configuration.

For additional information on this topic, please refer to the Tech Note entitled, "Data Recovery Options for Brocade Encryption in EMC Replication Environments." This document is available on the Brocade partner website or through your local Brocade SE.

Multiple Paths to a LUN

When there is more than one path to a LUN, which is almost always for most production environments, it is necessary to configure the CTC for all paths. This is the primary reason why it is required to use Brocade Network Advisor Professional Plus or Enterprise versions to configure Brocade encryption solutions. Brocade Network Advisor Professional only provides a view of a single fabric at one time, which may lead to the omission of adding the additional path to a LUN in the CTC. With Brocade Network Advisor Professional Plus and Enterprise versions, multiple fabrics can be viewed simultaneously, allowing the user to view the multiple paths to a given LUN. However, as a best practice it is recommended to use the Brocade Network Advisor Enterprise version.

BEST PRACTICE:

- When configuring LUN encryption with multiple paths to the LUN, it is useful to use "Storage Groups" in the GUI. This enables the administrator to see all paths to a given LUN under one heading, as opposed to remembering to locate all the paths separately.

Clustering Applications

Clustering applications allow the grouping of several hosts that can take over the load of a failed host within the cluster. Brocade encryption solutions support a variety of popular clustering applications. Refer to the Brocade Encryption Interoperability Matrix on www.brocade.com for the most current list of supported cluster applications.

The main issue when configuring clustered servers is to ensure that all the paths to a LUN have been defined in the CTC before the data on the LUN is encrypted. Imagine the consequences of beginning a first-time encryption operation on a LUN with only one path to the LUN defined at the beginning of the operation. As the FTE progresses, a failover occurs of the application server and is now moved to a different server, whose path to the LUN is not defined in the CTC. The data is now sent to the LUN, which is only partially encrypted at this point, and does not get redirected to the encryption switch (since it is not defined in the CTC). This data is in cleartext format and may be destined to a portion of the LUN that was previously encrypted by the FTE operation, resulting in data corruption by mixing cleartext and ciphertext data on the same LUN.

It is possible to perform an FTE operation when not all paths to the LUN are defined in the CTC only if the other paths are disabled until they are added to the CTC. Given the complexity of this procedure, it is not recommended.

BEST PRACTICE:

- In clustered applications, enter all paths to the LUN in the CTC before initiating a first-time encryption.
- Add all paths to the LUN in cleartext mode first. Once all paths to the LUN have been clearly defined in cleartext, only then do you modify it to “encrypt”.
- Never attempt a first-time encryption of existing data on a LUN unless you are completely confident of how to perform this safely. Performing this incorrectly may result in a corrupted LUN.
- **IMPORTANT NOTE:** Zoning of the hosts must be done prior to the creation or modification of a CTC. If you need to change the zoning for a host after it has been added to a CTC, it is essential to disable the device while this takes place; otherwise, this may result in corruption of the data on the LUN.

Cleaning Up After a POC

Generally, Brocade encryption solutions are deployed using brand new encryption devices arriving directly from the factory with no prior configuration. Installation and configuration of such devices are relatively straightforward. Occasionally, however, a POC is performed before installing into the production environment. It is extremely important to properly clean up the encryption device and LUNs used in a POC prior to installing it in a production environment.

Cleaning Up the Encryption Device

When an encryption device needs to be reset to its factory defaults—such as after a POC or when it is being decommissioned—it is necessary, and generally a good practice, to wipe out all sensitive information in the encryption device. This information includes the DEKs and configuration information. Please refer to your local Brocade SE or other Brocade technical resource to provide you with further guidance on how to accomplish this.

Decommissioning a LUN

Once an encrypted LUN is no longer used, the data on it needs to be destroyed. At the very least, you should destroy the key used to decrypt the data. Decommissioning a LUN that has been encrypted using a Brocade encryption solution requires several steps to maintain the confidentiality of the data. The following general steps should be used for this purpose:

1. Clear the metadata on the LUN to render the contents of the LUN useless.
2. Remove the LUN from the CTC across all paths.
3. Flag the associated DEK in the key manager to a “decommission” state.
4. Manually delete the DEK in the key manager after the decommissioning is complete.

Cleaning Up a LUN

If a LUN was used in a POC, was previously encrypted during this process, and is subsequently put into production, there is no means of determining what is on the first 16 blocks (8K) of data—that is, whether it is metadata or production data. When the data on the LUN is wiped clean following a POC, part of the file system might also be wiped out. Furthermore, the metadata will not be erased from the LUN, which can potentially result in encryption metadata still being present if this same LUN is added in the future as a cleartext LUN. Therefore, the only guaranteed method to remove the existing encryption metadata is to perform a "modify" and convert the encrypted data back to cleartext, as well as to back up the data on the LUN to tape or disk. Converting the encrypted data back to cleartext removes the metadata from the LUN that would otherwise remain.

First-Time Encryption Operations

Many organizations have a policy regarding a sensitive operation, such as a data migration or encryption of data on a LUN, to quiesce the environment first and then perform this operation offline. Other organizations cannot afford any downtime and must perform an FTE or rekey operation online.

Brocade encryption solutions allow for both online and offline FTE or rekey operations. An online FTE or rekey operation may result in performance degradation of the applications accessing the LUN during this process, as a result of I/O contention between the application requirements and the FTE or rekey operation. To minimize the impact on the production environment while performing an FTE/rekey operation, any frames arriving from the production environment are placed at the top of the I/O queue to give them priority over the FTE/rekey operation. For most typical environments, the FTE/rekey operation has minimal impact on the production environment, except for those with more I/O intensive applications.

A first-time encryption operation is very I/O intensive, and the disk drives involved in this process most likely are heavily used from the constant successive read and write operations. Encryption bandwidth is also utilized heavily during these operations. When a Brocade encryption solution has been purchased to encrypt both disk and tape data concurrently, the bandwidth for the encryption device is now shared between the disk and tape encryption operations, which results in less overall available bandwidth for disk or tape encryption alone. Given that an FTE operation is very I/O intensive and requires more encryption bandwidth, Brocade does not support performing an FTE or rekey operation at the same time as a concurrent encrypted backup. Furthermore, contention for the disk resources is also increased significantly, both with the constant read/write operations for the FTE operation and also for the backup of the same volume at the same time. Performance in this scenario is reduced drastically and backup times significantly increased if the tape drives are not fed the data from the disk fast enough to maintain a constant data stream. This generally results in a "shoe shining" effect—where the tape drive starts and stops frequently as it is waiting for data, which results in a dramatic drop in performance. The increased mechanical start and stop motion also taxes the read/write mechanisms of the tape drives significantly, leading to greater wear and a shortened lifespan for the tape drives. For this reason, performing concurrent encrypted backups while also performing a rekey or FTE is not supported by Brocade.

As a best practice, it is highly recommended that you schedule an FTE operation outside of the normal backup hours to avoid increased contention for encryption resources on the encryption device. It is also recommended that you perform a full backup of the LUN to be encrypted prior to performing this operation, as a precautionary procedure

in the event of a misconfiguration or error that might result in a corruption of the data on the LUN. This would

require a complete restore of the volume. When the proper procedures are followed to encrypt a LUN, corruption

of the LUN is not possible; however, human error during the procedure may result in corruption if incorrect parameters are given. For example, a LUN with existing production data can be corrupted if it is encrypted without the "-enable_encexistingdata" parameter.

Although it is very difficult to arrive at an exact figure as to how much time it will take to perform an FTE/rekey operation, a general guideline can be provided. Of course, the amount of cache, disk speeds, number of spindles per volumes, and other characteristics of the disk arrays all have an impact on the overall performance

and time to complete an FTE/rekey operation. As a general guideline, you can assume that it will require approximately 24 hours to encrypt 1.2 TB of data. It is important to note that performing 10 concurrent FTE/rekey operations will not result exactly in the encryption of 12 TB per 24-hour period, but rather that the number will likely approximate 10 TB.

BEST PRACTICE:

- Perform an FTE operation during off-peak hours whenever possible.
- Perform a test of an FTE of a LUN on each encryption engine and each array before attempting on other production LUNs.
- Perform multiple FTE operations simultaneously (up to 10) to maximize the utilization of resources and to reduce total time to perform this operation for larger environments.
- Do not perform an FTE or rekey operation while performing concurrent encrypted backups.

Tape Storage Considerations

Tape Pools

Some backup applications support tape pools, which can be used to back up groups of servers to a given set of tapes. As a best practice, it is advisable to encrypt tape backups using a unique DEK per tape media. However, for backup applications that support tape pools, it is possible to assign a unique key for all the tapes in a specified tape pool. This can be useful in some corner cases where, for example, a set of tapes containing data to be loaded at another facility (such as for auditing purposes) and only one key are required to decrypt the data on the entire

tape set.

The following backup applications support tape pools:

- Symantec NetBackup
- EMC Networker

The following backup applications do not support tape pools:

- Commvault
- IBM Tivoli Storage Manager (TSM)
- HP DataProtector
- CA ArcServe

Double Encryption and Compression

Tape drives all have native compression capabilities built into the hardware. Since the Brocade encryption device also performs compression, the question arises whether the native compression capability of the tape drive is somehow affected by the Brocade compression. In fact, it is required that you keep the native compression capability of the tape drive enabled when encrypting tapes using a Brocade encryption solution. This may seem a bit redundant, given that both encrypted data and compressed data are not compressible; what is the point of attempting to compress the data further at the tape drive? In fact, the Brocade compression pads the compressed data stream with binary zeros to retain the same length as the original cleartext data stream. This is done to ensure constant streaming of the data to the tape drive. Once the compressed data stream with the padded 0s reaches the tape drive, the native compression of the tape drive compresses the pad of binary zeros, resulting in fully compressed data on the tape medium.

Some tape drives also have built-in hardware encryption capabilities, such as LTO-4 tape drives. It is entirely possible to encrypt data using a Brocade encryption solution, then to encrypt the data a second time using the built-in tape drive encryption feature. Depending on the tape drive technology, the encryption process may add additional performance degradation. In this case, it may be preferable to disable the native encryption capability

of the tape drive, since the Brocade encryption does not decrease the performance of the backup. At any rate, it is entirely possible to perform a double encryption of data without any negative effects to the data. However, recovering the data does require the use of an encrypting tape drive to do the first decryption pass prior to doing the second decryption pass using the Brocade encryption device.

Since the data is encrypted using a Brocade encryption solution, it is unnecessary to double-encrypt. For the sake of simplifying the management of encrypted backups, it is preferable to disable the native encryption capability of a tape drive when using the Brocade encryption solution to encrypt tape data. This avoids any complexities that result from managing two different encryption solutions. This may be particularly challenging when performing a restore from a double-encrypted tape.

BEST PRACTICE:

- Do not use double-encryption; disable the native encryption capability of a tape drive when encrypting tape data using a Brocade encryption solution.

MANAGEMENT CONSIDERATIONS

This section covers the management best practices to use once the Brocade encryption has been installed and configured for operation in a production environment.

Reverting Back to Cleartext

Occasionally, someone may wish to convert an encrypted LUN back to cleartext format. Perhaps the incorrect LUN has been selected by mistake. Perhaps the data has been reclassified and no longer requires encryption. Perhaps a production LUN was used as part of a POC and needs to be reverted back to cleartext after the POC (not recommended). Brocade encryption solutions do not have the ability to unencrypt previously encrypted LUNs. The only way this can be accomplished is to use a host-based replication method and create a separate LUN of equal size to the encrypted LUN. You would then copy the data from the encrypted LUN, decrypting it through the Brocade Encryption Switch and writing it back in cleartext to the new LUN. Clearly, this is a labor-intensive operation and is likely to be disruptive, which highlights the importance of proper planning when selecting which LUNs need to be encrypted.

The general procedure described below outlines the recommended method to properly convert an encrypted LUN to cleartext:

- Always back up your data before beginning the process.
- Stop all traffic to the encrypted LUNs.
- Create new LUNs as needed for the cleartext data on the disk array.
- Add the new LUNs to the CTC (all CTCs, if multipathed) and present them as cleartext LUNs to the server.
- Mount and format the new LUNs (this procedure is OS-dependent).
- Copy the data from the encrypted LUNs to the cleartext LUNs.
- Validate the integrity of the data on the cleartext LUNs.
- Decommission the encrypted LUNs that are to be removed.
- Remove the decommissioned LUNs from the CTC (all CTCs, if multipathed).
- Delete the decommissioned LUNs from the array.

FTE and Rekey Operations

A rekey operation might be required after the DEK of the LUN has been compromised or after it has expired. It is possible to configure the Brocade encryption device to automatically begin a rekey operation once the DEK expires. However, best practice is to configure the encryption device to perform the rekey operation manually. Since a rekey operation is very I/O intensive and may degrade performance of the application, a

manual rekey allows the scheduling of the rekey operation at a time when it is most convenient, such as during off-peak hours.

An FTE or rekey operation is very I/O intensive; the disk drives involved in this process are most likely heavily used from the constant successive read and write operations. Encryption bandwidth is also utilized heavily during these operations. When a Brocade encryption solution has been purchased to encrypt both disk and tape data concurrently, the bandwidth for the encryption device is now shared between the disk and tape encryption operations. This results in less overall available bandwidth for disk or tape encryption alone. Given that an FTE or rekey operation is going to be very I/O intensive and requires more encryption bandwidth, it is important to consider the impact on available encryption bandwidth while performing these operations. Furthermore, contention for the disk resources is also increased significantly with both the constant read/write operations for the FTE or rekey operation and also for the backup of the same volume at the same time. Performance in this scenario is reduced drastically and backup times increased significantly if the tape drives are not fed the data from the disk fast enough to maintain a constant data stream. This generally results in a "shoe shining" effect, where the tape drive starts and stops frequently as it is waiting for data. This results in a dramatic drop in performance. The increased mechanical start and stop motion also taxes the read/write mechanisms of the tape drives significantly, leading to greater wear and shortened lifespan of the tape drives.

An encryption device is capable of performing up to 10 concurrent rekey operations. As a best practice, especially for larger environments with large numbers of LUNs to encrypt, it is recommended that you perform more than one FTE or rekey simultaneously. This reduces the total time required to rekey or perform an FTE in a particular production environment. However, if the production environment has many I/O intensive applications, and contention for resources becomes an issue, then the number of concurrent FTE/rekey operations should be reduced.

As a best practice, it is highly recommended that you schedule an FTE or rekey operation outside of the normal backup hours to avoid increased contention for encryption resources on the encryption device. It is also recommended that you perform a full backup of the LUN to be encrypted prior to performing this operation. This is a precautionary procedure in the event of a misconfiguration or error that might result in corruption of the data on the LUN, which would then require a complete restore of the volume. When the proper procedures are followed to encrypt a LUN, corruption of the LUN is not possible. However, human error during the procedure may result in corruption if the incorrect parameters are given. For example, a LUN with existing production data can be corrupted if it is encrypted without the "-enable_encexistingdata" parameter.

BEST PRACTICE:

- Perform an FTE operation during off-peak hours.
- Perform multiple FTE operations simultaneously (up to 10) to maximize the utilization of resources and to reduce total time to perform this operation for larger environments.

Managing Encrypted Backups

It is possible to assign a unique DEK to a group or pool of tapes. As a best practice, it is recommended that a unique DEK is assigned to a specific tape media or cartridge, since a compromised DEK affects only a single tape media instead of an entire group of tapes within a tape pool. Nevertheless, a customer may choose to encrypt a group of tapes with the same DEK to avoid having to deal with multiple keys when transferring tapes. One specific military application, for instance, requires a common key across an entire environment, given that the media are all very mobile and the solution needs to be designed so that the data can be recovered using this single key anywhere in the world. Obviously, this is a very limited use design and is generally not applicable to most commercial operations. The point is that the Brocade encryption solution does have some flexibility in this area if specific requirements demand it.

BEST PRACTICE:

- Use a unique DEK for each tape medium.

Recovering Encrypted Backups at a DR Site

When backup tapes are used to restore data at a DR site, the DEK used to encrypt the data on the tape must be available. Also, an encryption device must be available at the DR site to decrypt the data on the tape medium. This can be accomplished using several methods. One method is to synchronize the DEK and key state information between the encryption device at the primary data center and the encryption switch at the DR site, by placing both switches in the same encryption group. Another method, if the encryption device at the DR site is not synchronized with the primary site encryption device, is to retrieve the keys from the key manager to the encryption device at the DR site. An IP connection to the key manager is required to accomplish this. When the Key ID is retrieved from the tape, and the associated DEK is not found in the encryption device cache, a GET_KEY command is performed to retrieve the necessary key from the key manager.

Managing Encryption Devices

The Brocade encryption devices can be managed using the CLI, but most administrators prefer using a GUI such as Brocade Network Advisor. With very few exceptions, Brocade Network Advisor is able to perform almost all operations that can be performed by the CLI and at least those that are used on a day-to-day management basis. At a minimum, Brocade Network Advisor Professional Plus is required to manage a dual-fabric disk encryption configuration. Brocade Network Advisor Professional has the ability to manage only a single fabric at one time, and it is important when configuring a LUN for encryption that all paths to that LUN (usually several) are configured. Since Brocade Network Advisor Professional presents only one fabric or path at one time, it might be easy to forget to configure the second path or to misidentify the second path when switching Brocade Network Advisor Professional to the other fabric. Brocade Network Advisor Professional Plus or Enterprise version can display multiple fabrics. It displays all paths to a specific LUN, making it easier to configure a LUN for encryption.

Zeroizing the DEKs

A situation may occur requiring the removal of the DEKs that are cached on the encryption device. For instance, the encryption device may be decommissioned. In certain military field applications, destruction of the keys can be critical if there is a threat of the encryption device being captured or compromised. There is no “panic button” on the encryption device that can be pressed to automatically wipe out or zeroize the keys stored in its cache. However, it is possible to create a script that could zeroize the keys on the encryption device, which could be triggered from a panic button or switch. In some cases, a master panic button could be designed that would activate several scripts to perform a “self-destruct” process to protect several devices containing sensitive information, such as encryption keys.

Group Leader Loses a Group Member

Occasionally, an encryption device may fail. When it is part of an encryption group, the group leader device has lost one of its members. This is referred to as the “split-brain” problem. Since the procedure to recover a lost group leader may change over time, please refer to the Encryption Admin Guide for up-to-date instructions.

Brocade FOS and Firmware Upgrades

As a best practice, it is always recommended that you use the latest version of Brocade FOS that is supported by the OEM vendor that owns the support contract for the encryption devices. It is important to note that a firmware upgrade on the Brocade encryption device is disruptive to encryption traffic I/O. All Layer 2 FC traffic that is not being redirected is not affected, but any traffic that is redirected is affected, since the encryption engines and blade processor must reset.

As a general rule, it is best not to perform a firmware upgrade on a Brocade Encryption Switch or Brocade DCX 8510/DCX/DCX-4S Backbone with encryption blades while other I/O intensive operations are taking place, such as a first-time encryption or rekey operation. To avoid production downtime for disk environments using a dual-fabric configuration, upgrade the encryption devices on one fabric at a time. First, fail over the traffic from Fabric A onto Fabric B, then upgrade Fabric A. Next, when Fabric A is back online, fail over the traffic from Fabric B to Fabric A and upgrade Fabric B. It is also a good practice to perform the firmware upgrade one encryption device at a time.

To avoid affecting production for tape environments using a single fabric, it is recommended that you perform the upgrade during off-peak hours or in the next available maintenance window.

BEST PRACTICE:

- It is recommended to always use the most current version of Brocade FOS supported by the OEM.
- Never perform a firmware upgrade when performing a first-time encryption or rekey operation.
- Upgrade the firmware on one Brocade Encryption Switch or Brocade DCX 8510/DCX/DCX-4S with encryption blades at a time.
- Make sure automatic rekeys are not set to commence during a firmware download operation.

Encryption Performance Monitoring

With Brocade FOS v7.1, the Brocade CLI provides visibility into the redirected data throughput, providing valuable information to help administrators identify bottlenecks, resolve performance issues, and optimize encryption throughput. It also facilitates capacity planning, providing guidance for scaling the encryption environment as it grows.

As with measuring performance of any real-world environment, the adage “your mileage may vary” applies. The throughput speeds observed will not in all likelihood reach the theoretical maximums of 48 Gbps or 96 Gbps, due to the limitations inherent in the end-to-end solution. You can ensure maximum throughput performance of your Brocade encryption solution by following the system configuration and design guidance provided within this reference guide and by working with your Brocade System Engineer and the Brocade Professional Services organization.

APPENDIX A—SAMPLE USE CASES

This section describes some common use cases of Brocade encryption solutions.

Single Encryption Switch Fabric

One outsourcing customer decided on the following architecture. Although this customer had a rather large SAN environment, they had one isolated SAN for a specific customer. This existing customer needed to encrypt the disk data for compliance reasons. Since this SAN consisted only of two small fabrics, each with one 32-port switch, it was decided to simply replace the existing FC switches with the Brocade Encryption Switch.

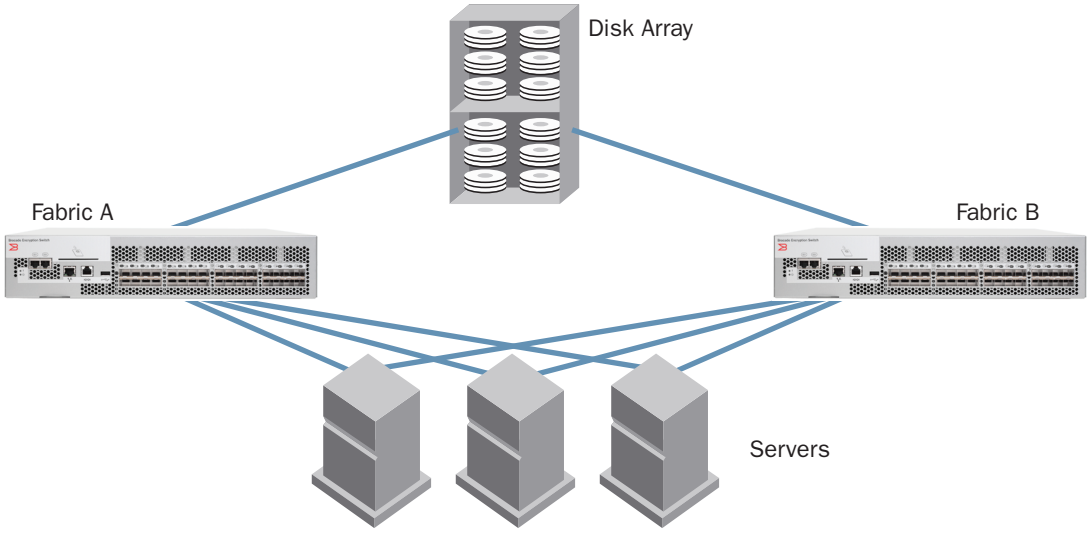

Figure 20. Dual-fabric SAN with only one Brocade Encryption Switch making up each fabric.

Encryption Switch Added to Existing Fabric

This architecture is probably the most common implementation of the Brocade Encryption Switch. Customers with an existing dual-fabric SAN in their production environment want to add the encryption capability to their SAN by simply adding a Brocade Encryption Switch in each fabric. The following example has a core-edge fabric topology, but the Brocade Encryption Switch can be added to any fabric topology. Notice that, in this example, none of the devices (hosts or storage devices) involved in the encryption process are actually physically connected to the Brocade Encryption Switch. It is entirely possible to connect the storage array or any of the hosts to the Brocade Encryption Switch, but it is not essential to do so for this solution to work. As explained previously, the frame redirection technology redirects frames through the encryption switch from the host to the actual LUN on the disk array.

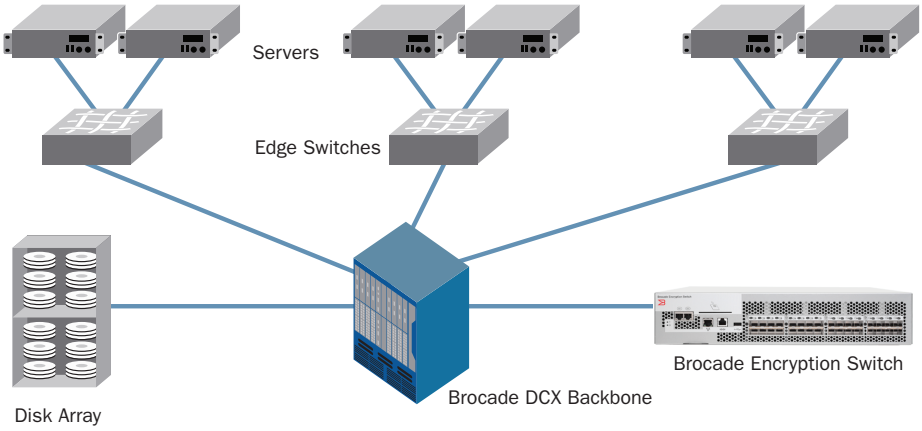

Figure 21. Integrating a Brocade Encryption Switch into an existing fabric.

Note that only one fabric of the dual-fabric configuration is shown in the above example, to simplify the illustration. Additionally, for simplification, only one ISL is shown between the core switch and the Brocade Encryption Switch. For optimal performance, the number of ISLs to the Brocade Encryption Switch should be sufficient to provide the necessary bandwidth required for this environment.

Tape Backup Fabric with Encryption

Another common use of Brocade encryption solutions is for the encryption of backup tape data. In this specific example, a customer uses a separate fabric for backups other than the fabric they use for their production disk environment—a common practice in many larger data centers.

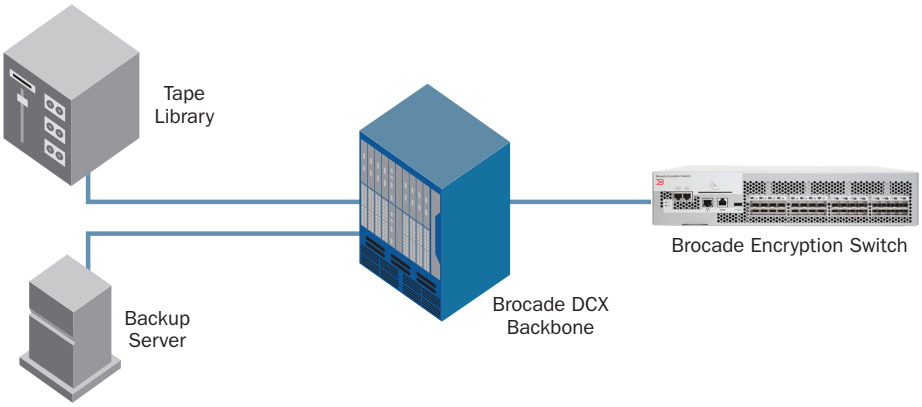

Figure 22. Tape backup SAN with a Brocade Encryption Switch.

Note that for simplification, only one ISL is shown between the core switch and the Brocade Encryption Switch. For optimal performance, the number of ISLs to the Brocade Encryption Switch should be sufficient to provide the necessary bandwidth required for this environment. Additionally, many customers have simply chosen to add a Brocade FS8-18 blade in the Brocade DCX 8510/DCX Backbone instead of adding an external Brocade Encryption Switch device for this type of configuration—either topology will work.

APPENDIX B—GLOSSARY OF TERMS

Table 3. Glossary

Term	Meaning
Ciphertext	The output of an encryption algorithm, that is, encrypted data that is unreadable (the opposite of plaintext)
Cleartext	Readable data that is transmitted or stored in an unencrypted or readable format
CryptoTarget Container (CTC)	A combination of storage port and host initiator that is used to create the redirection zone
Cryptographic algorithm or cipher	The actual procedure used to manipulate a readable message and render it unreadable
Cryptographic system or cryptosystem	The hardware or software implementation used to convert plaintext into ciphertext, and vice versa
Data Encryption Key (DEK)	The key used to encrypt data-at-rest
Decryption	The process of converting unreadable data into readable data (the opposite of encryption)
Encryption	The process of converting readable data into an unreadable format
Encryption device	Either a specialized standalone switch or a blade designed to encrypt data-at-rest
Encryption engine	Another name for an encryption device (see above)
Key	The secret code made up of a sequence of characters, bits, and/or instructions used in conjunction with an encryption algorithm to encrypt and decrypt messages made up of a sequence of characters, bits, and/or instructions
Key Encryption Key (KEK)	The key used to encrypt DEKs prior to exporting them to an opaque key vault, such as those from HP, IBM, RSA, and Thales
Key space	The total number of possible keys that exist using a given key size and algorithm
Plaintext	A cryptographic term that refers to the input to an encryption algorithm. The difference between ”cleartext” and “plaintext” is subtle, and they are often used interchangeably, although this is incorrect.
Substitution cipher	A code that mixes up the characters in a sequence (such as an alphabet) in a random order to encrypt data
Transposition cipher	A code that shifts or slides the characters in a sequence (such as an alphabet) either to the left or to the right by a specified number of places to encrypt data