Redbooks
ibm.com/redbooks

IBM DS8880 Data-at-rest Encryption

Bert Dufrasne
Sherry Brunson
Andreas Reinhardt
Robert Tondini
Roland Wolf

Security

Storage

IBM

Redpaper

International Technical Support Organization

IBM DS8880 Data-at-rest Encryption

December 2016

REDP-4500-06

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Seventh Edition (December 2016)

This edition applies to the IBM DS8880 system with firmware Release 8.2.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	IBM®	System Storage®
DB2®	IBM Spectrum™	Tivoli®
DB2 Universal Database™	Passport Advantage®	WebSphere®
DS8000®	Redbooks®	XIV®
FlashCopy®	Redpaper™	z Systems™
Global Technology Services®	Redbooks (logo)  ®	z/OS®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

LTO, Ultrium, the LTO Logo and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

IBM® experts recognize the need for data protection, both from hardware or software failures, and also from physical relocation of hardware, theft, and retasking of existing hardware.

The IBM DS8880 supports encryption-capable hard disk drives (HDDs) and flash drives. These Full Disk Encryption (FDE) drive sets are used with key management services that are provided by IBM Security Key Lifecycle Manager software or Gemalto SafeNet KeySecure to allow encryption for data-at-rest on a DS8880. Use of encryption technology involves several considerations that are critical for you to understand to maintain the security and accessibility of encrypted data.

This IBM Redpaper™ publication contains information that can help storage administrators plan for disk encryption. It also explains how to install and manage the encrypted storage and how to comply with IBM requirements for using the IBM DS8000® encrypted disk storage system.

Important: Failure to follow these requirements can result in an *encryption deadlock*.

This edition focuses on IBM Security Key Lifecycle Manager Version 2.6. It also introduces Gemalto SafeNet KeySecure Version 8.3.2, which supports the Key Management Interoperability Protocol (KMIP) with the DS8000 Release V8.1 code and updated GUI for encryption functions.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Bert Dufrasne is an IBM Certified Consulting IT Specialist and Project Leader for IBM System Storage® disk and flash products at the ITSO, San Jose Center. He has worked at IBM in various IT areas. Bertrand has written many IBM Redbooks® publications and has also developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect in the retail, banking, telecommunication, and healthcare industries. He holds a master's degree in electrical engineering.

Sherry Brunson joined IBM in March of 1985 and worked as large system IBM service representative before becoming a Top Gun in 1990. Sherry is a Top Gun in the Eastern US for all storage products, power systems, and IBM z Systems®. She has supported and implemented DS8000 and Scaled out Network Appliance storage products globally as well as developing and teaching educational classes. She also has taught z System classes in the United States.

Andreas Reinhardt is an IBM Certified Specialist for high-end disk systems in Mainz, Germany, and has worked in various IT areas at IBM for 15 years. Andreas works for IBM Global Technology Services® and started as an IBM System Service Representative (IBM SSR). He is now a member of the DS8000 and FlashSystem PFE teams with a key role in DS8000 fast recovery for DS8000 disk storage systems, FlashSystem, and with the DS8000 encryption implementation team in EMEA.

Robert Tondini is a Certified Senior IT specialist who is based in IBM Australia and provides storage technical support. He has 20 years of experience in the mainframe and storage environments. His areas of expertise include IBM high-end disk and tape storage subsystems and disaster recovery solutions. He has co-authored several books and held workshops for IBM enterprise storage systems, Advanced Copy Services, and IBM Copy Services Manager.

Roland Wolf is a Senior Systems Engineer with IBM Business Partner SVA System Vertrieb Alexander GmbH in Germany. He has worked at IBM for 28 years, with the last 23 years in storage with a background of mainframe high-end storage and all disk storage products. For the last two years, he has worked for an IBM Business Partner in Germany. Roland is co-author of many IBM Redbooks publications that cover ESS, the DS8000, IBM XIV®, and copy services. Roland holds a PhD in Physics.

Special thanks to the following people for their input and advice in preparation of this edition:

Jacob Sheppard, Justin Cripps, Pamela Schull, Tony Ciaravella

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and client satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form:

  ibm.com/redbooks

- Send your comments in an email message:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:
  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Summary of changes

This section describes the technical changes that are made in this edition of the paper and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for IBM DS8880 Data-at-rest Encryption

December 2016, Seventh Edition

This revision includes the following new and changed information.

New information

- Added Key Management Interoperability Protocol (KMIP) support information.
- Documented Gemalto SafeNet KeySecure product support.
- Added the IBM Security Key Lifecycle Manager replication procedure.

Changed information

- IBM Security Key Lifecycle Manager V2.5 was replaced by IBM Security Key Lifecycle Manager V2.6.
- Various changes that are related to GUI support for encryption.
- IBM Tivoli® Key Lifecycle Manager specific information was removed.

1

Encryption overview

Strong security is a must in today's round-the-clock, global business environment. Ensuring the protection and security of an organization's information is the foundation of any successful business.

Encrypting data at rest is a key element when addressing these concerns. The IBM DS8000 series supports hardware level self-encrypting Full Disk Encryption (FDE) disks and flexible key manager software. DS8000 encryption secures data at rest and offers a simple, cost-effective solution for securely erasing any disk or flash drive that is being retired or repurposed (cryptographic erasure).

However, encryption must not be deployed without careful planning and a thorough understanding of encryption techniques and encryption management products.

Important (encryption deadlock): Improper handling or implementation can result in a *permanent encryption deadlock*, which is mostly equivalent to the permanent loss of all key-server managed encrypted data, as described in 2.4, "Encryption deadlock" on page 36.

To gain access to data, even in a deadlock situation, the DS8000 offers a recovery key (RK) implementation. The RK can be set only as the *first activity* when setting up a DS8000. The RK can be configured as *disabled* in those environments where you do not want to maintain a RK.

This chapter covers the following topics:

- Business context for encryption
- Encryption concepts and terminology:
  - Symmetric key encryption
  - Asymmetric key encryption
  - Digital certificate
  - Hybrid encryption

- Introduction to Gemalto SafeNet KeySecure and Key Management Interoperability Protocol (KMIP)
- Encryption infrastructure and management with the IBM Security Key Lifecycle Manager

1.1 Business context

Businesses need tools to protect against the known threats, but also guard against as yet unknown threats. Effective threat and vulnerability management must be proactive rather than reactive, preventing problems rather than responding to them. To be efficient and effective, businesses must address prevention, detection, and compliance in an integrated way.

1.1.1 Threats and security challenges

Figure 1-1 illustrates how threats and challenges add to the complexity and the cost of running your business.

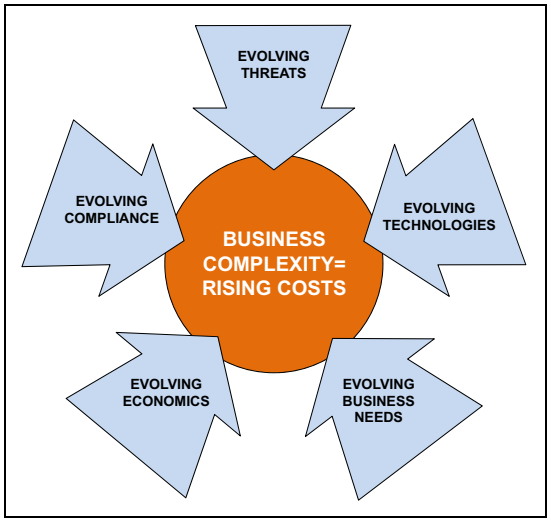


Figure 1-1 Business complexity

Companies face certain threats and security challenges:

- Increasing number and sophistication of threats. Businesses face more than just viruses and worms. You must be able to defend against all threats rather than respond only to intrusions.
- Prevention of data breaches and inappropriate data disclosure, and ensuring no impact on business and productivity.
- Intrusions that affect the bottom line in both client confidence and business productivity. Security breaches can destroy your brand image and affect your critical business processes.
- Growing demand for regulatory compliance and reporting. You must meet a growing number of compliance initiatives without diverting resources from core activities.
- Protecting your data and maintaining appropriate levels of access.
- Security issues are both internal *and* external. How do you protect against the well-intentioned employee who mishandles information, and the malicious outsider?

- Having your business comply with a growing number of corporate standards and government regulations. You must have tools that can document the status of your application security.
- Growing number of regulatory mandates. You must prove that your physical assets are secure.

1.1.2 Need for encryption

In particular, organizations experience a continued push to minimize the risks of data breaches. There is a new focus on privacy management tools with the capability to mask data. This focus reinforces the need for cryptography, and subsequent demand to simplify the complexity of the key-based algorithms and management of keys throughout the lifecycle.

A significant concern is when disk drives leave the company premises, which usually happens when a disk drive fails and the IBM technician replaces it with a new drive. Often, the drive is not damaged and data can still be accessed. IBM has a procedure to delete all data on the drive. However, this task is no longer under the control of the client. Some clients buy back the drives and destroy them themselves. This procedure can be expensive. Another concern is when the whole DS8000 is returned to IBM. The IBM technician erases all data, but this step is not sufficient for some clients. IBM offers a service (IBM Certified Secure Data Overwrite) to erase all data (several passes) in compliance with the American Department of Defense regulations (DoD 5220.20-M).

All of these concerns become obsolete when data on the drives is encrypted. Without a decryption key, the data is unreadable.

What should you encrypt and what should you not encrypt? Simply encrypt everything that you can encrypt and still be able to recover data if there is a disaster. If system data can be separated from application data, encrypting everything with no performance impact is easier than choosing which data falls into which legislation for encryption, and trying to keep current on the dynamic privacy rights rules and regulations.

Before using any encryption technology, understanding the encryption concepts and the requirements to maintain the security and the accessibility of the encrypted data is essential.

You do not want the encryption solution to affect negatively your storage environment and the applications that depend on it. You want an encryption solution that does not degrade application performance or jeopardize your disaster recovery plan. You also need the assurance that encryption does not cause any data loss and that all the appropriate measures are taken to protect and safeguard the encryption keys.

To address these concerns, the DS8000 encryption approach uses disks that have encryption hardware and can perform symmetric encryption and decryption of data at full disk speed and with no impact on performance. The disk-based encryption is combined with an enterprise-scale key management infrastructure. That infrastructure is based on the IBM Security Key Lifecycle Manager or Gemalto SafeNet KeySecure, both providing the same capabilities. These security lifecycle management software products help organizations efficiently deploy, back up, restore, and delete keys and certificates securely and consistently.

Important (additional encryption): The DS8000 provides disk-based encryption for data at rest on disk. If encryption over the network is required, additional encryption services must be investigated and deployed.

For a successful deployment, following the instructions and guidelines that are outlined in this document is also imperative.

For more information, see the IBM Security products website:

http://www.ibm.com/security/index.html

1.2 Encryption concepts and terminology

Encryption transforms data that is unprotected, or *plain text*, into encrypted data, or *ciphertext,* by using a *key*. Without knowledge of the encryption key, the ciphertext cannot be converted back to plain text.

Computer technology enabled increasingly sophisticated encryption algorithms. Working with the U.S. Government National Institute of Standards and Technology (NIST), IBM invented one of the first computer-based algorithms, Data Encryption Standard (DES), in 1974. Today, several widely used encryption algorithms exist, including 3Key-3DES and the more secure Advanced Encryption Standard (AES) for symmetric encryption, Rivest-Shamir-Adleman(RSA), which is commonly used for public keys, and Secure Hash Algorithm (SHA) for key derivation functions. There are many more encryption algorithms that are not mentioned here.

1.2.1 Symmetric key encryption

Early encryption methods used the same key to encrypt plain text to generate ciphertext, and to decrypt the ciphertext to regenerate the plain text. Because the same key is used for both encryption and decryption, this method is called *symmetric encryption*. All of the encryption algorithms that are previously mentioned use symmetric encryption.

Everyone who obtains knowledge of the key can transform the ciphertext back to plain text. If you want to preserve confidentiality, you must protect your key and keep it a secret. Therefore, symmetric encryption is also called *private* or *secret key encryption*, which is not to be confused with the private key in an asymmetric key system.

Figure 1-2 on page 5 shows a sample encryption and decryption data flow path. In the figure, the AES_256_ITSO symmetric key is used to encrypt plain text by using the AES encryption algorithm, which yields encrypted data. The decryption of the enciphered text uses the same AES_256_ITSO symmetric key and the AES algorithm to decrypt the data back to its plain text format.

Symmetric key encryption algorithms are much faster than asymmetric encryption algorithms, which makes symmetric encryption an ideal candidate for encrypting large amounts of data.

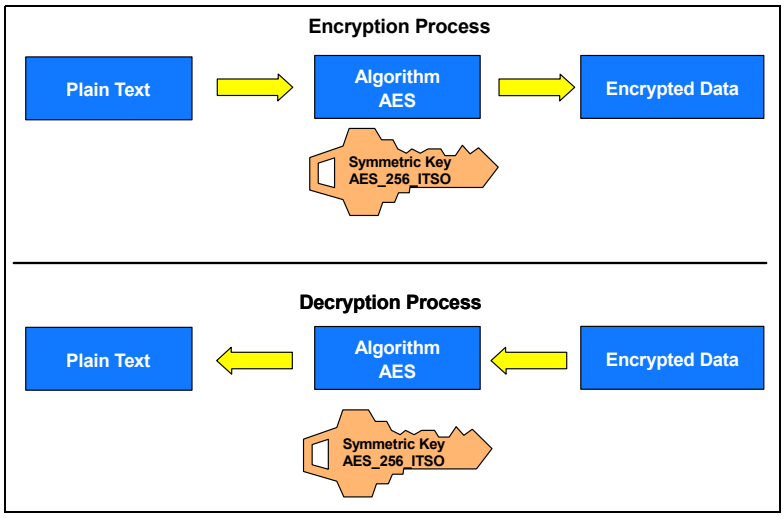


Figure 1-2 Symmetric key encryption

1.2.2 Asymmetric key encryption

In the 1970s, cryptographers invented asymmetric key algorithms for encryption and decryption. Encryption methods that use separate keys for encryption and decryption are called *asymmetric encryption.* Asymmetric encryption addresses certain drawbacks of symmetric encryption, which became more important with computer-based cryptography.

Asymmetric key encryption uses one key for encrypting (*public key*) and one key (*private key*) for decrypting data. Because the key that is used for encrypting a message cannot be used for decrypting, this key does not have to be kept a secret. It can be widely shared and is called a *public key*. Anyone who wants to send secure data to an organization can use its public key. The receiving organization then uses its *private key* to decrypt the data. The private key must always be kept a secret. Because asymmetric encryption uses public/private key pairs, it is also called *public/private key encryption* or *public key encryption.*

Public/private key encryption is widely used on the Internet today to secure transactions, including Secure Sockets Layer (SSL).

To encrypt data requires an algorithm. Today, the RSA algorithm[1] is the most widely used public key technique.

The advantage of asymmetric key encryption is the ability to share secret data without sharing the encryption key. But disadvantages exist too. Asymmetric key encryption is computationally more intensive and slower than symmetric key encryption. In practice, you often use a combination of symmetric and asymmetric encryption. This method is described in 1.2.3, “Hybrid encryption” on page 8. With the DS8000, the IBM solution uses a combination of symmetric and asymmetric encryption methods. This combination (*hybrid encryption*) is prevalent in many security solutions.

Important: The FDE solution uses only the asymmetric RSA algorithm to encrypt symmetric AES keys that are used for data encryption.

[1] Ronald L. Rivest, Adi Shamir, and Leonard Adleman developed this algorithm in 1977.

Figure 1-3 shows an encryption and decryption data path when using public key encryption.

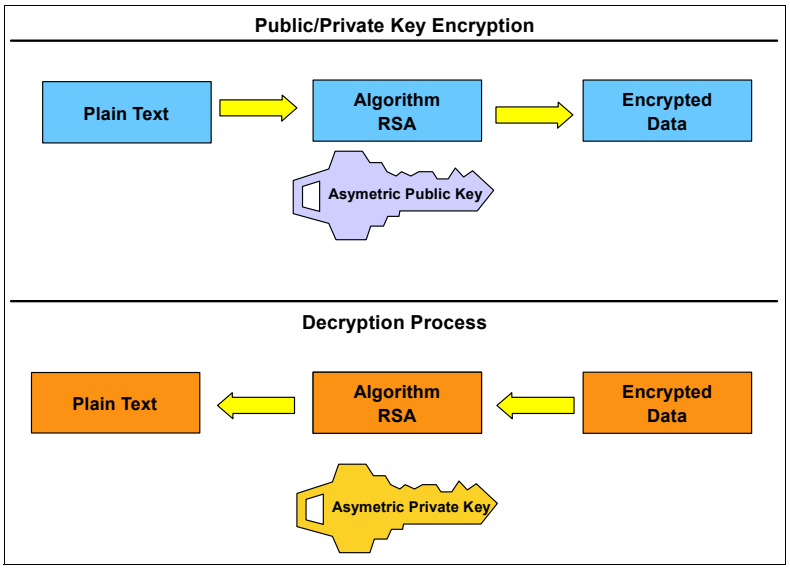


Figure 1-3 Public/private key encryption

Digital signature

You can use public/private key pairs to protect the content of a message and to sign digitally a message. When a digitally signed message is sent, the receiver can be sure that the sender sent it because the receiver can provide proof by using the public key from the sender. In practice, predominantly for efficiency reasons, a hash value of the message is signed rather than the whole message, but the overall procedure is the same.

Figure 1-4 on page 7 shows how the digital signature is used in the communication between the DS8000 and the key server, such as IBM Security Key Lifecycle Manager or Gemalto SafeNet KeySecure, by using an asymmetric key pair. It illustrates a mechanism that is used as part of the DS8000 encryption process. The DS8000 has a private key, and the key server has a copy of the DS8000 public key.

The DS8000 sends the key server a message that is encrypted with the DS8000 disk storage system's private key. The key server then uses the DS8000 public key to validate the message that is sent from the DS8000. The key server cannot use the public key to decrypt the encrypted data, but it can, with the DS8000 public key, validate that the message was encrypted with the DS8000 private key. This approach proves to the key server that it is communicating with the DS8000 because only the DS8000 has a copy of its private key. Then, the key server uses the DS8000 public key to encrypt the communication that it wants to protect and sends the data to the DS8000. The DS8000 can use its private key to decrypt the data.

IBM Security Key Lifecycle Manager replaces IBM Tivoli Key Lifecycle Manager. Tivoli Key Lifecycle Manager is no longer covered in this book. IBM Security Key Lifecycle Manager Version 2.6 is the current version.

Note: IBM Security Key Lifecycle Manager Version 2.6 or Gemalto SafeNet KeySecure is a requirement if you want to be compliant with the NIST Special Publication (SP) 800-131a.

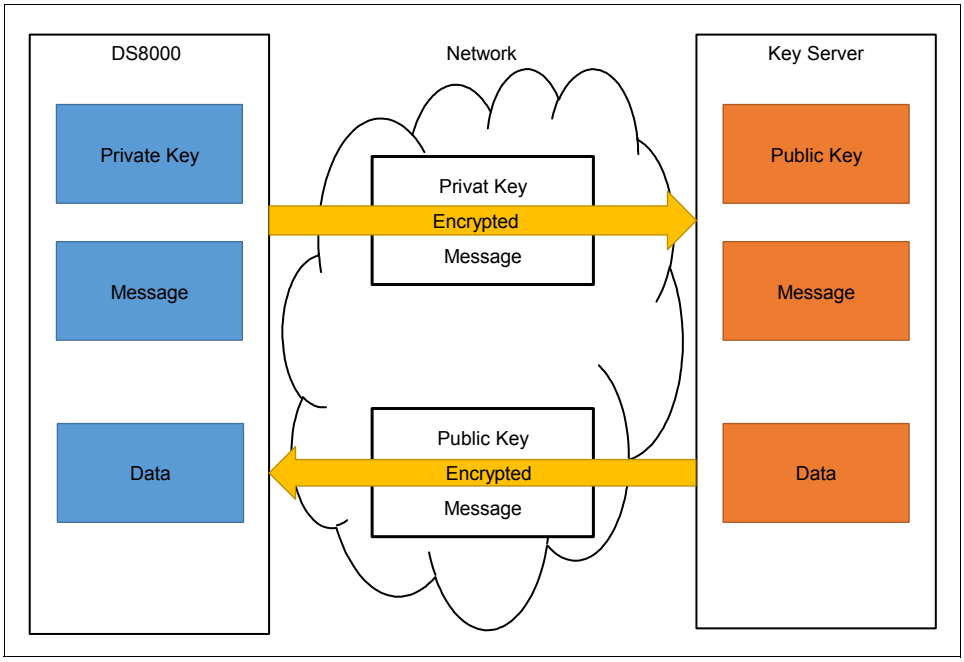


Figure 1-4 Identity verification by using public/private key encryption

Digital certificates

Another possibility is to make sure that the sender can trust the receiver by using a *certificate*, which is signed by a *certificate authority* (CA).

Digital certificates are a way to bind public key information with an identity. The certificates are signed by a CA. If users trust the CA and can verify the CA's signature, then they can also verify that a certain public key does indeed belong to whomever (a person or an entity) is identified in the certificate.

Part of the information that is stored in a digital certificate includes the following items:

- Name of the issuer
- Subject Distinguished Name (DN)
- Public key belonging to the owner
- Validity date for the public key
- Serial number of the digital certificate
- Digital signature of the issuer

Note: For the DS8000, digital certificates are created and set by manufacturing for each Storage Facility. IBM initially introduced disk encryption on the DS8000 with 80-bit security strength; IBM calls this the Gen-1 certificate.

DS8000 Release 7.2 and later offer 112-bit security strength; IBM calls this the Gen-2 certificate. Any DS8000 that is delivered by manufacturing with Release 8.1 *does not* support Gen-1 80-bit security strength certificates.

Both asymmetric and symmetric key encryption schemes are powerful ways to protect and secure data. Sections 2.2, "IBM Security Key Lifecycle Manager Encryption key management" on page 24 or see 2.3, "SafeNet KeySecure key management with KMIP" on page 34s give details about their use with the IBM DS8000 series. Both provide an extremely secure way of protecting data.

1.2.3 Hybrid encryption

In practice, encryption methods often combine symmetric and asymmetric encryption. Thus, the methods can take advantage of fast encryption with symmetric encryption and still securely exchange keys by using asymmetric encryption.

Hybrid methods use a symmetric data key to encrypt and decrypt data. They do not transfer this symmetric data key in the clear, but use public/private key encryption to encrypt the data key. The recipient can decrypt the encrypted data key and use the data key to encrypt or decrypt a message.

With hybrid encryption methods, you can combine secure and convenient key exchange with fast and efficient encryption of large amounts of data.

The FDE solution uses a symmetric AES data key to encrypt and decrypt data. This data key is protected by the asymmetric RSA algorithm and is not available in plaintext when the storage device communicates with the IBM Security Key Lifecycle Manager or any other third-party Key Server, such as Gemalto SafeNet KeySecure. For more information, see 1.4, “Key Lifecycle Manager” on page 10.

1.2.4 Communication protocols IPP, SSL/TLS V1.2, and KMIP

This section covers the different protocols that are supported by DS8000 encryption.

IBM Proprietary Protocol

The DS8000 exclusively supports IBM Proprietary Protocol (IPP) while communicating with the IBM Security Key Lifecycle Manager. The IPP protocol was first developed for tape drives to communicate with IBM Encryption Key Manager (IBM EKM), the predecessor to Tivoli Key Lifecycle Manager. The DS8000 incorporated IPP into its earliest encryption versions and continues to support it. The IPP is wrapped with TCP when communicating with the IBM Security Key Lifecycle Manager, and uses the default port 3801.

Secure Sockets Layer/Transport Layer Security V1.2

For the United States federal government, a minimum security strength of 80 bits was the recommendation until 2010. This requirement was achieved by IPP. By the beginning of 2011, the minimum key strength number increased from 80 bits to 112 bits. However, with the acceptance of a certain amount of risk, the minimum of 80 bits of security strength can be used until the end of 2013. NIST SP 800-131a requires longer key lengths and stronger cryptography than other standards. The standard requires cryptographic algorithms that have key strengths of at least 112 bits. IPP is wrapped into Secure Sockets Layer (SSL) secured by Transport Layer Security (TLS) when communicating with the IBM Security Key Lifecycle Manager by using default port 441 and having NIST mode enabled.

Strict enforcement of NIST SP 800-131a imposes the usage of the TLS 1.2 protocol for the SSL context.

For more information about how the DS8870 and subsequent products address compliance with NIST SP 800-131a, see *IBM DS8870 and NIST SP 800-131a Compliance*, REDP-5069.

Key Management Interoperability Protocol

KMIP is an industry standard that strives to be a common language for key management systems and encryption systems of all varieties. There are many commercial key server vendors who support KMIP. Many systems, ranging from email databases to storage devices and that offer encryption, also support KMIP as their communication protocol.

By supporting KMIP, DS8880 Release 8.1 provides customers with more flexibility and choice in key management. DS8000 customers now may take advantage of encryption if KMIP is a requirement in their infrastructure.

Note: Only DS8000 Release 8.1 and later support KMIP with Gemalto SafeNet KeySecure.

Note: The encryption at a glance is shown in Figure 1-5 on page 20.

1.3 Encryption challenges

Encryption depends on encryption keys. Those keys must be kept secure and available, and responsibilities must be split:

- Keys security

  To preserve the security of encryption keys, the implementation must be set up so that no one individual (person or system) has access to all the information that is required to determine the encryption key. In a system-based solution, the encryption data keys are encrypted with a wrapping key (another key to encrypt/decrypt the data keys). This wrapped key method is used with the DS8000 by separating the storage of a wrapped data key that is stored on the disk from the storage of the wrap/unwrap keys within a key server.

- Key availability

  More than one individual (person or system) has access to any single piece of information that is necessary to determine the encryption key. In a system-based solution, redundancy is provided by having multiple isolated key servers. In addition, backups of the key server's data are maintained.

- Separation of responsibilities

  The DS8000 offers a RK to get access to data if none of the key servers are available. To prevent one person from gaining access to the data, the handling of a RK requires two people with separate roles: the Security Administrator and the Storage Administrator. It is also possible to disable the RK, but this is at the client's own risk.

  The sensitivity of possessing and maintaining encryption keys and the complexity of managing the number of encryption keys in a typical environment results in a client requirement for a key server. A key server is integrated with encrypting storage products to resolve most of the security and usability issues that are associated with key management for encrypted storage.

Lifecycle management tools: IBM offers an enterprise-scale key management infrastructure through the IBM Security Key Lifecycle Manager and lifecycle management tools to help organizations efficiently deploy, back up, restore, and delete keys and certificates in a secure and consistent fashion.

However, the client must still be sufficiently aware of how these products interact to provide the appropriate management of the IT environment. Even with a key server, in general, at least one encryption key (the overall key that manages access to all other encryption keys, or a key that encrypts the data that is used by the key server) or a RK must manually be maintained.

One critical consideration with a key server implementation is that all code and data objects, which are required to make the key server operational, must not be kept on storage that depends on any key server being accessed.

A situation where all key servers cannot become operational because there is data or code that cannot be accessed without an operational key server is referred to as an *encryption deadlock*. It is analogous to having a bank vault that can be unlocked with a combination, but the only copy of the combination is locked inside the vault.

This situation, and the policies and mechanisms that are required to avoid it, are described in Chapter 2, “IBM DS8000 encryption mechanism” on page 21.

1.4 Key Lifecycle Manager

In an enterprise, many symmetric keys, asymmetric keys, and certificates can exist. All of these keys and certificates must be managed, which can be handled by the IBM Security Key Lifecycle Manager or Gemalto SafeNet KeySecure. Both key managers provide a centralized encryption and key management solution for minimized risk of exposure and reduced operational costs.

Important: IBM Security Key Lifecycle Manager V2.6 was announced November 24, 2015 and replaces the IBM Security Key Lifecycle Manager V2.5. To use the KMIP protocol with DS8000 Release 8.1, a third-party key server, specifically Gemalto SafeNet KeySecure, is required. This section covers the IBM Security Key Lifecycle Manager functions only.

The IBM Security Key Lifecycle Manager provides key storage, key serving, and key lifecycle management for storage devices from IBM and other vendors. This storage includes various IBM tape drives, tape libraries, DS8000 disk storage systems, XIV storage systems, DS3700 storage systems, and also some Quantum tape libraries, Emulex HBAs, and Network Appliance storage models. For a detailed list of supported products, see the IBM Security Key Lifecycle Manager IBM Knowledge Center:

`http://ibm.biz/SKLMv26KC`

The focus of this document is on IBM key server interoperability with the DS8000. The DS8000 supports data encryption with the *FDE* feature. The FDE drives can encrypt and decrypt at interface speeds, so there is no impact on performance. All disk and flash drives in the DS8880 are capable of encryption by default. To use the encryption feature, the DS8000 must be configured to communicate with the Key Lifecycle Manager.

Indicators: Feature codes to enable Encryption Support are no longer required in the DS8000. The optional feature code 1760 (MT 2421-AP1 with feature codes 1761/1762) was introduced for the DS8000. It enables the new storage appliance server options that are available in addition to the existing isolated key managers.

Specific tasks to enable encryption must still be done (see Chapter 4, “IBM DS8000 encryption implementation” on page 55).

The DS8000 must be either all encrypted or all not encrypted. An environment verification process or solution assurance must be completed to ensure that preferred practices regarding the configuration of the encryption solution are taken. This verification can be requested from IBM Lab-Based Services, or completed by the client’s staff, but is a prerequisite for activating the encryption solution.

1.4.1 IBM Security Key Lifecycle Manager features overview

All features and functions that are previously supported by Tivoli Key Lifecycle Manager V2.0.1 and IBM Security Key Lifecycle Manager V2.5 are now supported in IBM Security Key Lifecycle Manager V2.6.

Note: The Disk Encryption mechanism has not changed, but TLS 1.2 encryption for network communication created the need to verify all components of the storage environment, including the version of the web browser. IBM Spectrum™ Control and Copy Services Manager are examples of where support for TLS 1.2 must be considered during planning.

Here are some of the functions:

- Automated clone replication: IBM Security Key Lifecycle Manager automated clone replication uses a program to clone a master IBM Security Key Lifecycle Manager server with up to 20 copies.
- KMIP: Extension of support to devices by using industry-standard KMIP for encryption of stored data and the corresponding cryptographic key management. DS8880 Release 8.1 does not support KMIP with IBM Security Key Lifecycle Manager.
- Master key in Hardware Security Module (HSM) support: IBM Security Key Lifecycle Manager supports the HSM to store the master key to protect all passwords that are stored in the database.

The following IBM Security Key Lifecycle Manager features are related to the DS8000:

- Role-based access control that provides permissions to do tasks such as create, modify, and delete for specific device groups. Most permissions are associated with specific device groups.
- A graphical user interface (GUI) and command-line interface (CLI) to manage keys, certificates, and devices.
- Encrypted keys to one or more devices to which the IBM Security Key Lifecycle Manager server is connected.
- Storage of keys, certificates, and metadata about these keys and certificates in a database.
- Cross-platform backup and restore to protect critical data and other IBM Security Key Lifecycle Manager data, such as the configuration files and current database information.
- Migration of the IBM Security Key Lifecycle Manager Version 1.0, 2.0, 2.0.1, and 2.5, and IBM EKM V2.1 component during installation.
- Audit records that are based on selected events occurring as a result of successful operations, unsuccessful operations, or both.
- A set of operations to replicate automatically current active files and data across operating systems. This replication enables cloning of IBM Security Key Lifecycle Manager environments on multiple servers in a manner that is independent of operating systems and the directory structure of the server.

For more information about the features of the IBM Security Key Lifecycle Manager and its predecessors, see the IBM Security Key Lifecycle Manager IBM Knowledge Center:

http://ibm.biz/KLM_welcome

1.4.2 New in IBM Security Key Lifecycle Manager V2.6

IBM Security Key Lifecycle Manager V2.6. provides several usability and interoperability improvements for installing, configuring, and migrating an IBM Security Key Lifecycle Manager infrastructure. In addition, the IBM Security Key Lifecycle Manager provides several advantages:

- Operating system independent UI-based replication configuration
- Operating system independent backup and restore operations
- Compliance with KIMP 1.2 and Storage Networking Industry Association Secure Storage Industry Forum (SNIA-SSIF) certification
- Setup for exporting a SSL/KMIP server certificate through the GUI
- NSA Suite B compliance
- Server Configuration Wizard to configure IBM Security Key Lifecycle Manager for SSL/TLS handshake

For a detailed list of new features, see the IBM Security Key Lifecycle Manager IBM Knowledge Center:

http://ibm.biz/SKLMv26KC

Note: Starting with IBM Security Key Lifecycle Manager V2.6, the Solaris operating system is no longer supported.

IBM Security Key Lifecycle Manager V2.6 does not support KMIP with the DS8000.

1.4.3 Key serving

The information about key serving, which is summarized in this section, can help you become familiar with the terms and statements that are used in the following chapters.

IBM Security Key Lifecycle Manager enables the definition and serving of keys, or groups of keys, which can be associated with a device. IBM Security Key Lifecycle Manager deploys separate key types to separate devices that request them.

Consider the following information:

- A key can be a member of a single *key group*, and deleting a *key group* deletes all keys in that group.
- The *key metadata* includes information such as a key alias, algorithm, and activation date. IBM Security Key Lifecycle Manager stores metadata for a key in the IBM Security Key Lifecycle Manager database.
- A key or certificate can be in the following states, which define the level of use that is allowed:
  - Pending
  - Pre-active
  - Active
  - Compromised
  - Deactivated
  - Destroyed
  - Destroyed-compromised

An object that is no longer active might change states from deactivated to destroyed, deactivated to compromised, compromised to destroyed-compromised, or destroyed to destroyed-compromised.

When a new key is generated, the key and its alias are stored in a designated *keystore*. Metadata for the key is stored in a key table in the IBM Security Key Lifecycle Manager database. The new key enters an active state immediately. When a certificate request is created, IBM Security Key Lifecycle Manager creates a key entry that is in a pending state. It waits for the return of a certificate that was approved and certified by a certificate authority.

Changing attributes: The compromised and information attributes of a key, regardless of its state, can be changed by using the CLI.

- Standard and operating system-specific Java keystore methods are supported by IBM Security Key Lifecycle Manager to store public/private key and certificate information. The Java Cryptography Extension KeyStore (JCEKS) *keystore type* is supported as the IBM Java Cryptography Extension (JCE) software provider (IBMJCE). It can be used for all distributed operating systems.

1.4.4 How to protect IBM Security Key Lifecycle Manager data

The IBM Security Key Lifecycle Manager contains critical information that must be protected. Several options are described in the following list:

- Backup: You can back up critical data files either on a secure computer at a geographically separate location or on a replica computer that provides another IBM Security Key Lifecycle Manager server. The replica computer enables quick recovery at times when the primary IBM Security Key Lifecycle Manager server is not available.
- Restore: A restore operation returns the IBM Security Key Lifecycle Manager server to a known state by using backed-up production data, such as the IBM Security Key Lifecycle Manager keystore and other critical information.
- Audit: On distributed systems, audit logs are stored in the Common Base Event format by IBM Security Key Lifecycle Manager.
- Automated clone replication: A master IBM Security Key Lifecycle Manager clone can be replicated to up to 20 copies. The replicated data includes tables in IBM Security Key Lifecycle Manager database, keys and certificates in a keystore, and IBM Security Key Lifecycle Manager configuration files.

Note: This data is the same as that included in an IBM Security Key Lifecycle Manager backup, except that during a replication, the replication configuration file and the SSL server certificate are not backed up and passed to the clone.

- Master key in hardware security modules: IBM Security Key Lifecycle Manager can use a hardware security module to store the master key to protect all passwords that are stored in the database, which adds an extra protection to the storage and use of the master key.

1.4.5 IBM Security Key Lifecycle Manager for open systems

IBM Security Key Lifecycle Manager V2.6 for open systems supports the hardware that is listed in Table 1-1. All systems must be writable.

Table 1-1 IBM Security Key Lifecycle Manager V2.6 hardware requirements

System component	Minimum value[a]	Recommended value[b]
System memory (RAM)	4 GB	8 GB
Processor speed	For Linux and Windows systems: 1.0 GHz single processor For IBM AIX®: 1.5 GHz (2-way)	For Linux and Windows systems: 3.0 GHz dual processors For AIX systems: 1.5 GHz (4-way)
Disk space free for product and prerequisite products, such as IBM DB2® Universal Database	16 GB	30 GB
Disk space free in `/tmp` or `C:\temp` directory	4 GB (12 GB)[c]	4 GB (12 GB)
Disk space free in `/home` directory for IBM DB2 Universal Database™	7 GB	25 GB
Disk space free in /var directory for DB2	1 GB on Linux and UNIX operating systems	1 GB on Linux and UNIX operating systems

a. Minimum values: These enable a basic use of IBM Security Key Lifecycle Manager.
b. Recommended values: You might need to use larger values that are appropriate for your production environment. The most critical requirements are to provide adequate system memory, and free disk and swap space. Processor speed is less important.
c. IBM Security Key Lifecycle Manager V2.6 and fix pack source files are about 8 GB in total. Consider reserving this space when not using DVD disks.

On Linux and UNIX operating systems, 4 GB of free space is required in the `$HOME` directory.

On Windows operating systems, the following free space is required in addition to what is required for your DB2 product:

- 40 MB in the system drive
- 60 MB in the `/temp` folder, which is specified by the temp environment variable

IBM Security Key Lifecycle Manager V2.6 supports the following software:

- AIX V6.1 and V7.1 in 32-bit mode. (64-bit kernel is required for all versions.)
- Red Hat Enterprise Linux Version 6.0 Update 6.0 and Version 7.0 on x86 64-bit mode
- Red Hat Enterprise Linux Version 6.0 Update 6.0 and Version 7.0 (IBM z Systems™) on x86 64-bit mode
- SUSE® Linux Enterprise Server Version 10 and Version 11 on x86 (64-bit mode)
- SUSE Linux Enterprise Server Version 11 (z Systems) on x86 64-bit mode
- Microsoft Windows Server 2012 on x86 64–bit
- Microsoft Windows Server 2012 R2 on x86 64–bit mode

1.5 IBM Security Key Lifecycle Manager for z/OS

As an alternative to the IBM Security Key Lifecycle Manager v2.6 for open systems, you can use the IBM Security Key Lifecycle Manager for z/OS® product. Previous names for this product are IBM EKM and IBM Tivoli Key Lifecycle Manager for the z/OS.

Attention: Do not confuse the IBM Security Key Lifecycle Manager for z/OS with the IBM Security Key Lifecycle Manager V2.6 for open systems or later.

IBM Security Key Lifecycle Manager for z/OS also helps when generating, protecting, storing, and maintaining encryption keys that are used to encrypt information being written to and decrypt information being read from devices.

IBM Security Key Lifecycle Manager for z/OS supports System Management Facilities (SMFs) for audit records. The IBM Security Key Lifecycle Manager for z/OS is part of the IBM Java environment, and uses the IBM Java Security components for its cryptographic capabilities.

1.5.1 IBM Security Key Lifecycle Manager for z/OS components

IBM Security Key Lifecycle Manager for z/OS has the following components:

- Java security keystore
- Configuration files
- Device table
- KeyGroups.xml file

Java security keystore

The keystore is defined as part of the Java Cryptography Extension (JCE). The keystore is an element of the Java Security components, which are part of the Java runtime environment. A keystore holds the certificates and keys (or pointers to the certificates and keys) that are used by the IBM Security Key Lifecycle Manager for z/OS to do cryptographic operations.

IBM Security Key Lifecycle Manager for z/OS supports non-hardware and hardware-assisted keystores. Hardware-based JCECCARACFKS keystores need a hardware cryptographic services provider. This support for hardware-assisted keystores makes IBM Security Key Lifecycle Manager for z/OS the preferred key server for a z/OS environment, at least for tape encryption.

Configuration files

With configuration files, you can customize the behavior of IBM Security Key Lifecycle Manager for z/OS to meet the needs of your organization.

Device table

The device table is used by IBM Security Key Lifecycle Manager for z/OS to monitor the devices it supports. The device table is a non-editable, binary file whose location is specified in the configuration file. You can change its location to meet your needs.

KeyGroups.xml file

This password-protected file contains the names of all encryption key groups and the aliases of the encryption keys that are associated with each key group.

1.5.2 Functions that are performed by IBM Security Key Lifecycle Manager for z/OS

IBM Security Key Lifecycle Manager for z/OS requests the generation of encryption keys and passes those keys to TS1120, TS1130, TS1140, TS1150, and LTO Ultrium 4, 5, and 6 tape drives, and DS8000 disk storage systems, to name some of the supported storage devices. When a DS8000 starts, the storage system requests an unlock key from IBM Security Key Lifecycle Manager for z/OS.

If the DS8000 requests a new key for its unlock key, IBM Security Key Lifecycle Manager for z/OS generates an AES key and serves the key to the DS8000 in two protected forms:

- Encrypted (wrapped), by using RSA key pairs. The DS8000 stores this copy of the key.
- Separately wrapped for secure transfer to the DS8000, where it is unwrapped upon arrival and the key inside is used to unlock the DS8000.

If the DS8000 requests an existing unlock key, the protected AES key on the DS8000 is sent to IBM Security Key Lifecycle Manager for z/OS, where the wrapped AES key is unwrapped. The AES key is then wrapped with a different key for secure transfer back to the DS8000, where it is unwrapped and used to unlock the system.

The IBM Security Key Lifecycle Manager for z/OS design allows redundancy and offers high availability. You can have multiple IBM Security Key Lifecycle Manager for z/OS sets that service the same devices. In this way, you can have two IBM Security Key Lifecycle Manager sets. They are mirror images of each other. They have built-in backup of the critical information about your keystores and serve as a failover options if one IBM Security Key Lifecycle Manager for z/OS set is not available. When you configure your DS8000, you can point it to two sets of IBM Security Key Lifecycle Manager for z/OS. If one IBM Security Key Lifecycle Manager for z/OS set is not available, your DS8000 uses the other IBM Security Key Lifecycle Manager for z/OS set.

You can also keep the two IBM Security Key Lifecycle Manager for z/OS sets synchronized. Be sure that you take advantage of this important function when necessary.

1.5.3 Preventing a deadlock situation

If IBM Security Key Lifecycle Manager for z/OS is used as a key manager for DS8000 encryption, you can run into a deadlock situation if you use key servers run only on encrypted DS8000 disk storage systems. When all z/OS LPARs are powered down and are restarted, the DS8000 disk storage systems with enabled encryption must “talk” to an encryption server to get the unlock key. However, IBM Security Key Lifecycle Manager for z/OS cannot start because its data in stored on an encrypted DS8000 disk. A DS8000 must have *all* encrypted data or *no* encrypted data. A mix of encrypted and non-encrypted data is not possible.

To avoid this type of deadlock, be sure that you have one of these setups available:

- A z/OS attached storage system that is not encrypted for the IBM Security Key Lifecycle Manager for z/OS logical partition (LPAR).
- A duplicate IBM Security Key Lifecycle Manager for z/OS set at the disaster recovery site with a backup copy of the data files.
- A stand-alone IBM Security Key Lifecycle Manager for open systems set as an alternative to IBM Security Key Lifecycle Manager for z/OS with a copy of the keys.

If you have an environment with an IBM Security Key Lifecycle Manager on z/OS and a stand-alone IBM Security Key Lifecycle Manager for open systems, you must create a certificate and a private key on IBM Security Key Lifecycle Manager for z/OS and one on IBM Security Key Lifecycle Manager for open systems. Export the certificates and then import the certificates for each other. This task means that the IBM Security Key Lifecycle Manager for z/OS certificate goes to IBM Security Key Lifecycle Manager for open systems and the IBM Security Key Lifecycle Manager for open systems certificate goes to IBM Security Key Lifecycle Manager for z/OS. Configure the DS8000 to use both certificates.

The DS8000 must communicate with at least *two* key servers because you cannot configure a DS8000 with encryption enabled if the DS8000 cannot communicate with two key servers. For a power-on operation, it is sufficient for the DS8000 to access only *one* key server.

1.5.4 Installing the IBM Security Key Lifecycle Manager for z/OS and keystores

Install IBM Security Key Lifecycle Manager for z/OS as instructed in *Program Directory for IBM Security Key Lifecycle Manager for z/OS V1.1.0*, found at:

`http://bit.ly/XrkOtq`

This section includes only an overview of the installation steps.

You can set up IBM Security Key Lifecycle Manager for z/OS in several ways, depending on the keystore types. IBM Security Key Lifecycle Manager for z/OS supports the following keystores:

- JCEKS
- JCECCAKS
- JCERACFKS
- JCECCARACFKS

IBM Security Key Lifecycle Manager for z/OS requires IBM Java Software Developer Kit 5.0 or 6.0 and later. It also requires the unrestricted policy files for Java. The files are available at the following website:

`https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk`

Here are the highlights of the installation steps:

1. Install the Java SDK.
2. Copy the unrestricted policy files. You must replace the `US_export_policy.jar` and `local_policy.jar` files in the following directory with an unrestricted version of these files:

   `$JAVA_HOME/lib/security`
3. Select the keystore type:
   - Add the Java hardware provider if you want to use hardware cryptography.

     If you decide to use a keystore type of either JCECCAKS or JCECCARACFKS so that you can use the security advantages of ICSF, you must add the Java hardware provider.

     You cannot use both JCERACFKS and JCECCARACFKS keystore types concurrently in the IBM Security Key Lifecycle Manager for z/OS configuration file. You must specify only one of these types in the configuration file. To add the Java hardware provider, you must edit the following file and complete the following steps:

     `$JAVA_HOME/lib/security/java.security`

i. If you want the RSA key to be secure and not visible in the clear, create your RSA keys in the ICSF PKDS by using either the **RACDCERT PCICC** option or **hwkeytool** with the **-hardwaretype** PKDS flag.

ii. If you want the data encryption key (DEK) to be secure and not visible in the clear, change the configuration to set the `requireHardwareProtectionForSymmetricKeys` property to `true`.

iii. Ensure that the IBMJCECCA provider is installed in your `java.security` provider list.

– If you are not using hardware cryptography, use a JCEKS keystore type by completing the following steps:

i. Obtain a list of all the aliases (or key labels) for the RSA keys that you want to use. For more information, see your keystore documentation.

ii. Obtain a list of all the type Drive Serial Numbers that you need to register. This is optional if you set `drive.acceptUnknownDrives = true` for automatic addition of tape drives to device table and `ds8k.acceptUnknownDrives=true` to accept automatically new DS8000 drives.

iii. Edit the `ISKLMConfig.properties.zos` file, as shown in *Configuration Basics*, to customize the entries that are appropriate for your installation. You can find Configuration Basics in the IBM Knowledge Center at:

`http://www.ibm.com/support/knowledgecenter/SSB2KG_1.0.0/com.ibm.tivoli.isklm.doc_11/top_EKMipug_configuring.html`

4. Set up a user to run the IBM Security Key Lifecycle Manager for z/OS.
5. Get digital certificates.
6. Set up the IBM Security Key Lifecycle Manager for z/OS configuration file.

   Edit `ISKLMConfig.properties.zos` to update the following values (ISKLM for z/OS must not be running when you edit the `ISKLMConfig.properties.zos` file):

   a. `Audit.handler.file.directory`: Specify a location where audit logs are stored.

   b. `Audit.metadata.file.name`: Specify a fully qualified path and file name for the metadata XML file.

   c. `config.drivetable.file.url`: Specify a location for information about drives that are known to IBM Security Key Lifecycle Manager for z/OS. This file is not required before starting the server or CLI client. If it does not exist, it is created during shutdown of the IBM Security Key Lifecycle Manager for z/OS server.

   d. `TransportListener.ssl.keystore.name`: Specify the path and file name of the keystore that is created in step 1 on page 17.

   e. `TransportListener.ssl.truststore.name`: Specify the path and file name of the keystore that is created in step 1 on page 17.

   f. `Admin.ssl.keystore.name`: Specify the path and file name of the keystore that is created in step 3 on page 17.

   g. `Admin.ssl.truststore.name`: Specify the path and file name of the keystore that is created in step 3 on page 17.

   h. `config.keystore.file`: Specify the path and file name of the keystore that is created in step 3 on page 17.

   i. `drive.acceptUnknownDrives`: Specify true or false. A value of true allows new tape drives that contact IBM Security Key Lifecycle Manager for z/OS to be automatically added to the device table. The default is `false`.

j. `ds8k.acceptUnknownDrives`: Specify `true` or `false`. A value of `true` allows a new DS8000 that contacts IBM Security Key Lifecycle Manager for z/OS to be automatically added to the device table. The default is `false`.

The following optional password entries can be added or omitted. If these entries are not specified in `ISKLMConfig.properties.zos`, IBM Security Key Lifecycle Manager for z/OS prompts for the keystore password during the start of the server. When added to the `ISKLMConfig.properties.zos` file, IBM Security Key Lifecycle Manager for z/OS obfuscates these passwords for additional security. Obfuscating the passwords ensures that they do not appear in the clear in the properties file.

- `Admin.ssl.keystore.password`: Specify the password of the keystore that is created in step 3 on page 17.
- `config.keystore.password`: Specify the password of the keystore.
- `TransportListener.ssl.keystore.password`: Specify the password of the keystore.

7. Define IBM Security Key Lifecycle Manager for z/OS as a started task:
   - To start IBM Security Key Lifecycle Manager for z/OS with JCERACFKS, run the following command:

     ```
     java -Djava.protocol.handler.pkgs=com.ibm.crypto.provider
     com.ibm.ltklm.ISKLMServer ISKLMConfig.properties.zos
     ```

   - To start IBM Security Key Lifecycle Manager for z/OS with JCECCARACFKS, run the following command:

     ```
     java -Djava.protocol.handler.pkgs=com.ibm.crypto.hdwrCCA.provider
     com.ibm.ltklm.ISKLMServer ISKLMConfig.properties.zos
     ```

   - To start IBM Security Key Lifecycle Manager for z/OS with JCEKS or JCECCAKS, run the following command:

     ```
     java com.ibm.ltklm.ISKLMServer ISKLMConfig.properties.zos
     ```

For more information, see the following resources:

- *IBM Security Key Lifecycle Manager for z/OS Version 1.1 Planning, and User's Guide*, SC14-7628
- The IBM Security Key Lifecycle Manager for z/OS Version 1.1. IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSB2KG_1.0.0/KC_ditamaps/welcome.html

1.6 Gemalto SafeNet KeySecure

Gemalto SafeNet Key Secure is a third-party, centralized key management platform. It offers an alternative to IBM Security Key Lifecycle Manager for clients who are required to use a KMIP infrastructure. It is fully supported by DS8880 Release 8.1 and later.

Gemalto SafeNet provides KeySecure as hardware and virtual software appliance. At the time of writing, the current version is 8.3.2 RevA. It supports the KMIP 1.1 (used with DS8000 Release 8.1), PKCS #11, JCE, MS-CAPI, ICAPI, and .NET APIs. LDAP and Active Directory authentication are included too, and multiple network management protocols.

Like IBM Security Key Lifecycle Manager, Gemalto SafeNet KeySecure supports 128-bit encryption and provides a GUI named "Gemalto SafeNet KeySecure Management Console" and an SSH CLI.

Gemalto SafeNet KeySecure can manage up to 1,000,000 keys and 1,000 devices, and it supports HSM to store the master key.

For more information about Gemalto SafeNet KeySecure, see the following website:

http://www.safenet-inc.com/data-encryption/enterprise-key-management/key-secure/

1.7 Encryption at a glance

In DS8000 encryption, many definitions appear, which can be easily misunderstood, misused, or falsely interpreted. There are protocols, standards, and software versions that are listed, whose intermixture is necessary to understand the whole picture. Figure 1-5 provides a description and explanation of the terms and concepts that are used by the DS8000 encryption process in an overview.

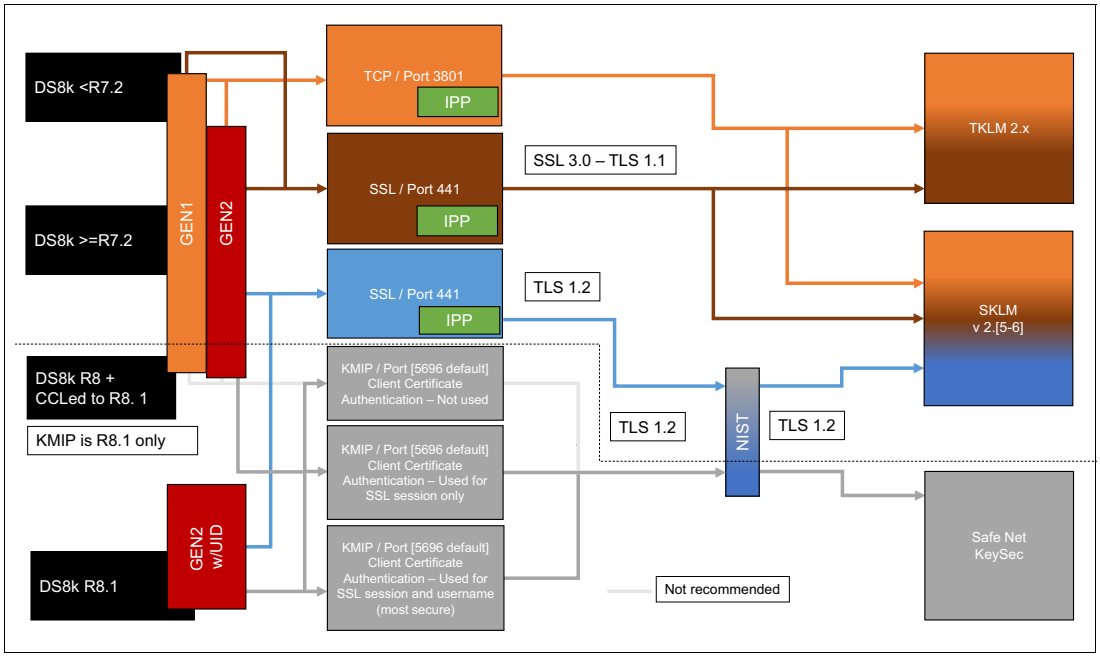


Figure 1-5 Encryption at a glance

2

IBM DS8000 encryption mechanism

This chapter provides information about the DS8000 disk encryption mechanisms.

This chapter contains the following topics:

- DS8000 disk encryption
- IBM Security Key Lifecycle Manager Encryption key management
- SafeNet KeySecure key management with KMIP
- Encryption deadlock
- Working with a recovery key
- Dual key server support

2.1 DS8000 disk encryption

The DS8000 supports data encryption in systems that are equipped with Full Disk Encryption (FDE) capable hard disk drives (HDDs) and flash devices (SSD in storage enclosure and in High Performance Flash Enclosure (HPFE)). All devices (disk or flash) in the DS8000 must be FDE capable. Intermix with non IBM FDE devices is not allowed.

FDE devices have encryption hardware, and can perform symmetric encryption and decryption of data at full disk speed with no impact on performance.

The disk encryption hardware is used with IBM Security Key Lifecycle Manager or Gemalto SafeNet KeySecure. IBM Security Key Lifecycle Manager uses a wrapped key method, and Gemalto SafeNet KeySecure uses the direct key method to deliver keys to encrypting storage devices. The DS8000 uses an external key server method to secure keys by separating the storage of a data key (DK) that is stored within the device from the storage of the keys within the key server. The wrap/unwrap keys are also referred to as the key encryption/key decryption keys. Without these keys, which are managed by the key servers, nobody can decrypt the data on disk.

Cryptographically erased: If all copies of the decryption key are lost (whether intentionally or accidentally), then no feasible way exists to decrypt the associated ciphertext, and the data that is contained in the ciphertext is said to be *cryptographically erased*. The data is lost because it cannot be decrypted without the key.

For more details about encryption key management, see 2.2, “IBM Security Key Lifecycle Manager Encryption key management” on page 24 and 2.3, “SafeNet KeySecure key management with KMIP” on page 34.

The DS8880 is equipped by default with FDE capable disks.

A DS8000 with FDE disks is referred to as being *encryption-capable*. An encryption-capable DS8000 can be configured to either enable or disable encryption for all data that is stored on client disks.

Attention: Enabling encryption cryptographically erases all data on the disks. Therefore, encryption must be enabled directly at the beginning, not when data is already stored in the DS8000.

The DS8000 must be configured to communicate with *at least two* key servers to enable encryption. Two key servers are required for redundancy. After the DS8000 powers on, it must be able to communicate with at least one of the key servers to get the unlock keys. The communication between the DS8000 and the key server is done through the Hardware Management Console (HMC).

The physical connection between the DS8000 HMC and the key server is through a TCP/IP network, as shown in Figure 2-1.

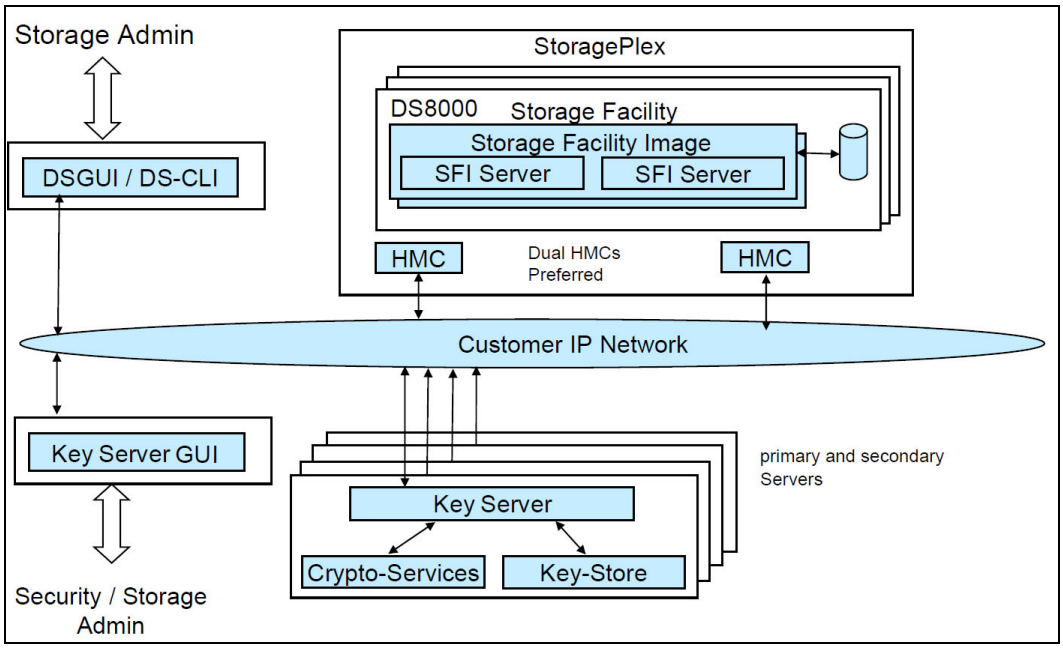


Figure 2-1 Connection between the DS8000 HMC key servers

Before explaining the various keys that are used by DS8000 and IBM Security Key Lifecycle Manager and Gemalto SafeNet KeySecure for encryption and how messages can be exchanged between two systems in a secure way (generically), you must learn about the concept of digital signatures.

Digital signatures are used to authenticate a sender. The digital signatures are generated by using the private and public keys. Figure 2-2 shows the following steps:

1. The sender writes its message.
2. According to a mathematical formula, a digital string, usually of a fixed length, is derived from the message. This string is called a *hash*. Although a hash is derived from and uniquely linked to the data, deriving the data from the hash is not possible.
3. The hash is encrypted with the *sender's private key.* The encrypted hash is called a *digital signature*.
4. The digital signature is attached to the message.
5. Both message and digital signature are encrypted with the receiver's public key.
6. The encrypted message is sent to the receiver.
7. The receiver decrypts the message and signature combination.

   Now the sender reproduces the message hash in two ways:

   – The receiver decrypts the digital signature with the sender's public key to get the original hash.

   – The receiver calculates the hash from the received message.
8. If both hashes match, the receiver has good reason to trust the message.

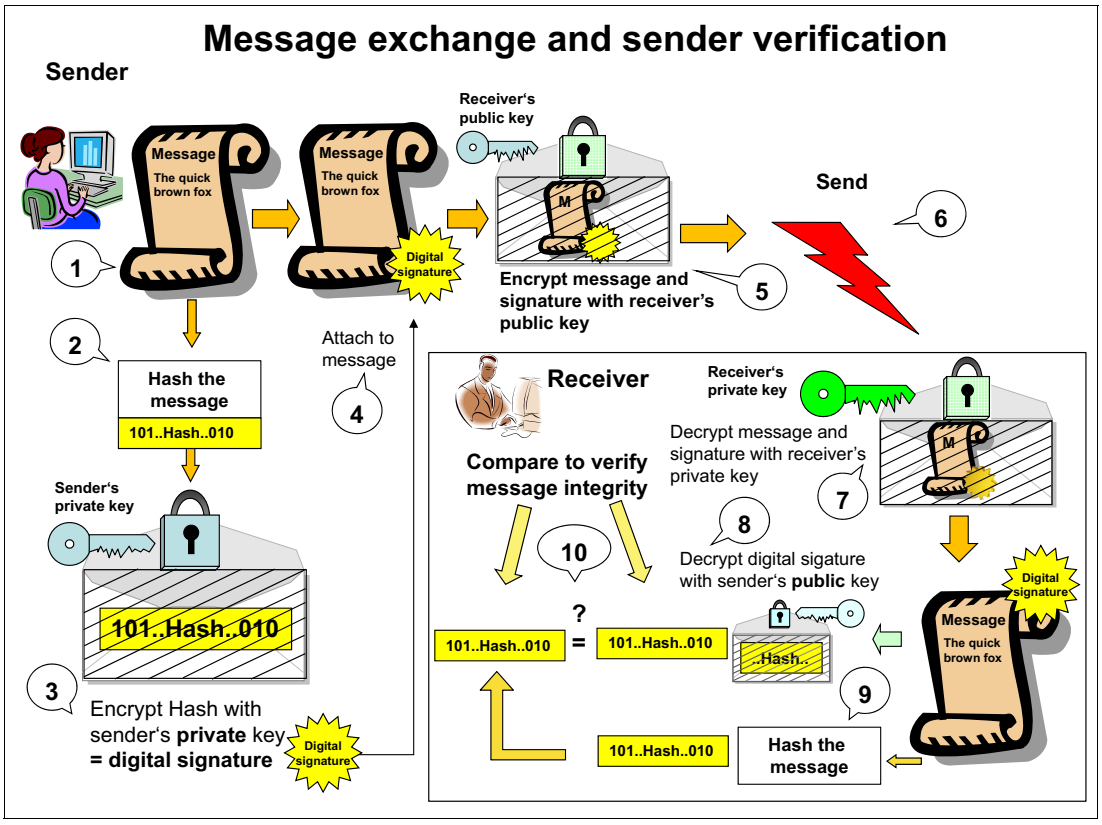


Figure 2-2 Authentication with digital signatures

2.2 IBM Security Key Lifecycle Manager Encryption key management

This section has details about how the IBM Security Key Lifecycle Manager key server manages and creates the encryption keys that are used by the DS8000 during key label, encryption group, rank creation, and DS8000 power-on time.

Important: Key negotiation and authentication between the IBM Security Key Lifecycle Manager and DS8000 take place at DS8000 power-on time only. There is no increased traffic use in an encrypted DS8000 at run time that is created by key negotiation.

The IBM Security Key Lifecycle Manager key server uses the wrapped key method to serve keys to an encryption-enabled DS8000. The wrap and unwrap keys on the key server are a public/private asymmetric key pair. The wrap key is referred to as the *public key encrypting key* (KEK) and the unwrap key is referred to as the *private key encrypting key* (KEK').

The configuration processes on the key server and the storage device (the DS8000) define one or more key labels. For more information, see 4.2, "IBM Security Key Lifecycle Manager Version 2.6 configuration" on page 60.

The key label is a user-specified text string that is associated with the asymmetric key label pair (KEK/KEK'), which is generated by the key server when the key label is configured (see Figure 2-3). The key generation and propagation processes on the key server associates a key label with each wrap/unwrap key pair. This key label is a user-specified text string that is

retained with each wrap/unwrap key pair. The key encrypting key-pair key is kept secret by the IBM Security Key Lifecycle Manager in a keystore.

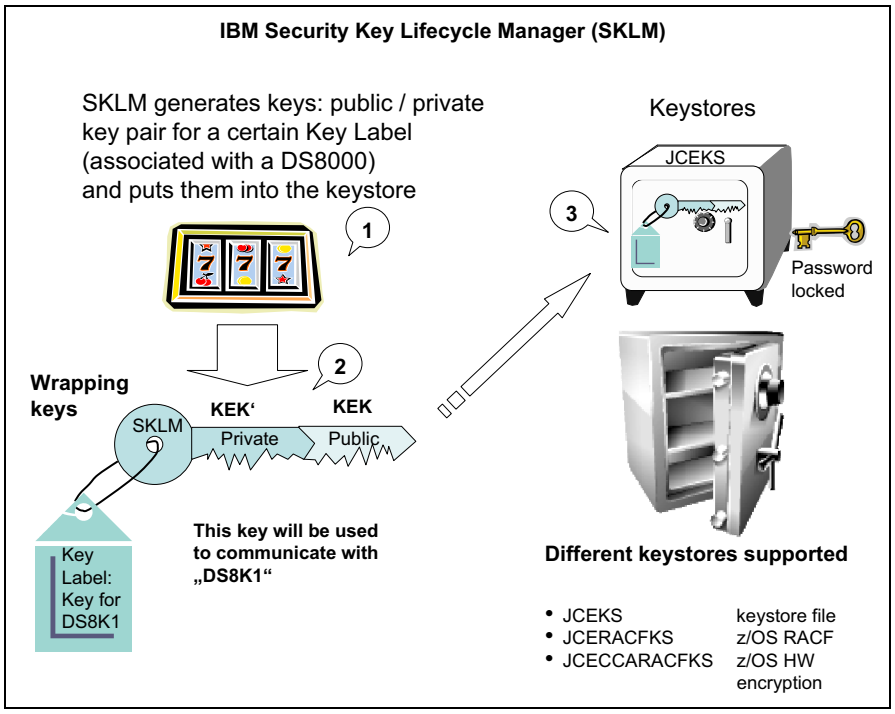


Figure 2-3 Configure IBM Security Key Lifecycle Manager key label

Rekey data key feature: Licensed Machine Code (LMC) level 6.5.1.xx (bundle Version 75.1.xx.xx) and later enables the rekey data key feature. This feature allows a user to change the DK labels (see 5.1, “Rekeying the data key” on page 150).

Now, the user (Storage Administrator) can use the DS8000 GUI to register the key server on the DS8000. Next, still using the DS8000 GUI, an encryption group is created. For more information, see 4.4.4, “GUI configuration for DS8000 encryption” on page 101).

As part of creating the encryption group with IPP, you must specify the key label that was set when configuring the IBM Security Key Lifecycle Manager server, which was configured for a particular DS8000.

Note: Currently, the DS8000 has only one encryption group.

While creating the encryption group, the DS8000, which is referred to as DS8K1 in our illustrated scenario, generates a “Device Session Key pair (device session public key/device session private key, respectively noted as DSK/DSK’) from a random number. The public/private key pair is associated with a key label. The DSK’ is kept secret by the DS8000.

The key label, DSK, and the DS8000 storage facility certificate, which was set and stored on the DS8000 by manufacturing, are sent to the IBM Security Key Lifecycle Manager on the key server to request a DK (see Figure 2-4).

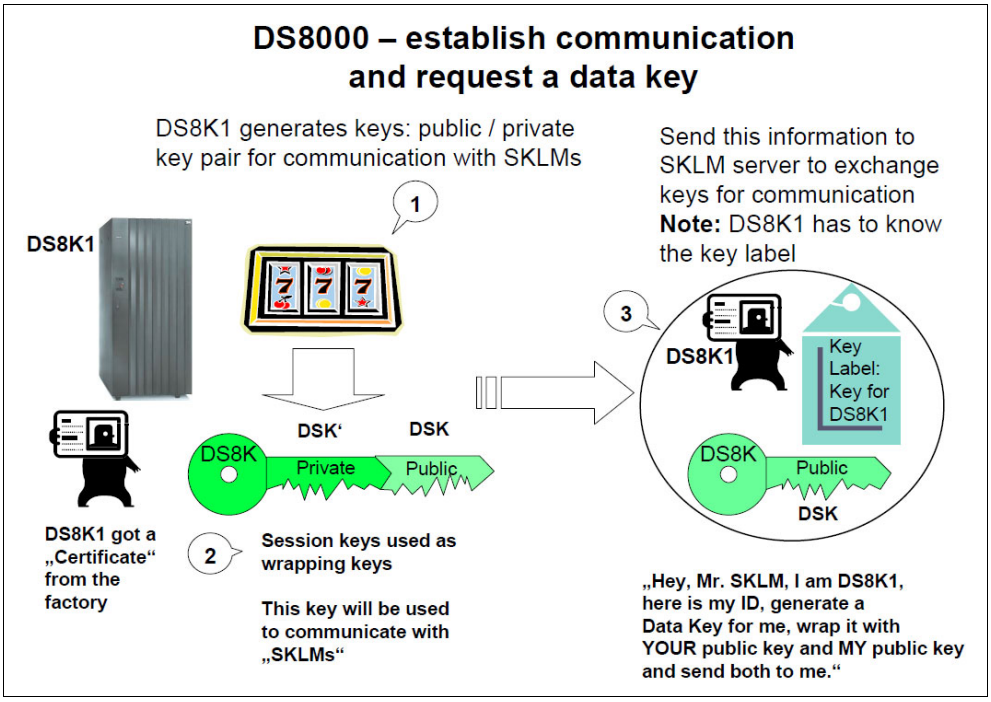


Figure 2-4 DS8000 creates session keys and requests a data key

Upon reception of these elements, IBM Security Key Lifecycle Manager carries out the following steps (see Figure 2-5 on page 27):

1. It validates the DS8000 certificate.
2. It generates the DK.
3. The DK is wrapped with DS8000 disk storage system's DSK and stored in a structure that is referred to as the session encrypted data key (SEDK).
4. From the key label, IBM Security Key Lifecycle Manager retrieves the KEK/KEK' pair for the specified key label. The DK is wrapped with the KEK and stored in a structure referred to as the Externally Encrypted Data Key (EEDK).

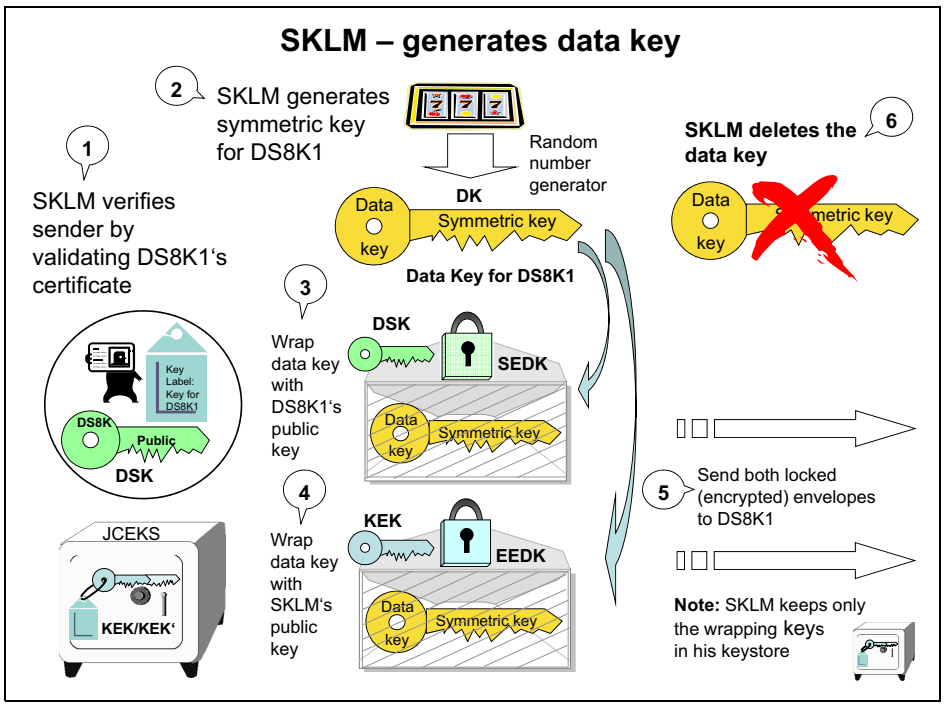


Figure 2-5 IBM Security Key Lifecycle Manager generates data key

Now, IBM Security Key Lifecycle Manager transfers the SEDK and EEDK to the DS8000 and the following steps occur at the DS8000:

1. The DS8000 receives the encrypted structures with the DK in it.
2. To re-create the DK at the DS8000, the SEDK is unwrapped with the DS8000 disk storage system's DSK'. The DS8000 holds the DK in memory, as shown in Figure 2-6.

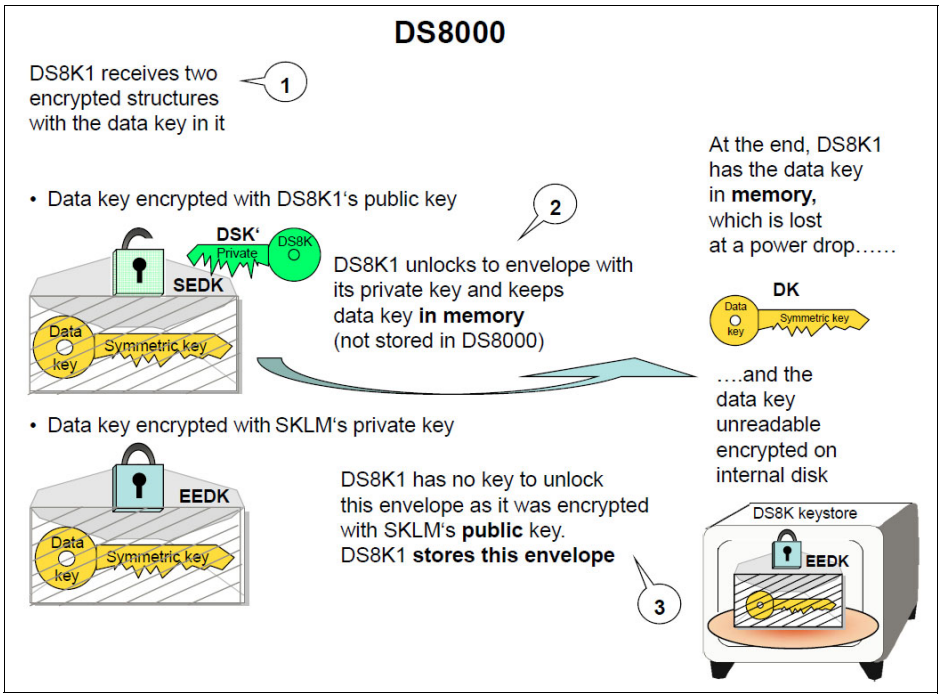


Figure 2-6 DS8000 unwraps data key and stores encrypted data key

3. The EEDK is stored in DS8000 disk storage system's keystore. The DS8000 does not have the key to unlock this structure.
4. The DS8000 generates a random 256-bit group key (GK) for the encryption group. See Figure 2-7.

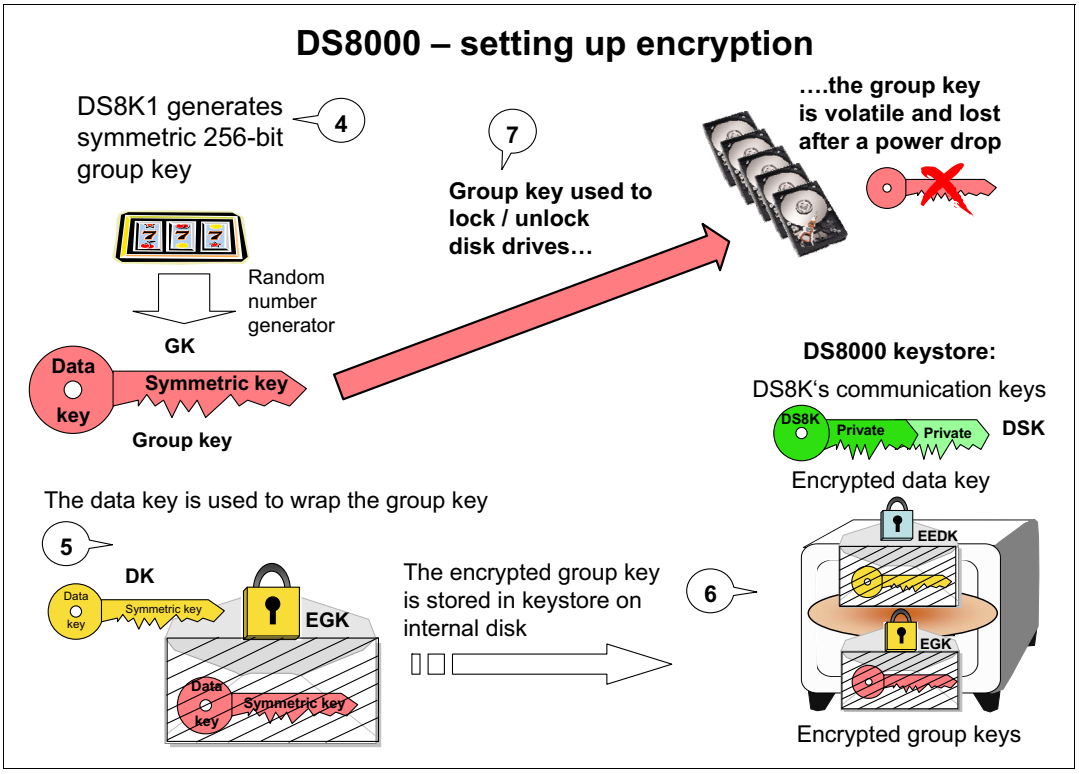


Figure 2-7 Set up encryption

5. The GK is wrapped with the DK and stored in a structure that is referred as the Encrypted group key (EGK).
6. The EGK is persistently stored in the key repository (KR) of the DS8000. Both the EEDK and the EGK are stored in multiple places in the DS8000 for reliability.

   This dual control (from the DS8000 and IBM Security Key Lifecycle Manager) improves security. The DS8000 does not maintain a persistent copy of the DK on disk in the clear, and cannot encrypt or decrypt data without access to IBM Security Key Lifecycle Manager.

   The DK is *erased* by the DS8000 at power off, such that each time it is powered on, the DS8000 must communicate with IBM Security Key Lifecycle Manager to obtain the DK again.

When the user configures a rank, the DS8000 creates, for each DDM in this rank, an access credential to lock the drive, as shown in Figure 2-8. The following steps occur during configuration of the rank:

1. The DS8000 reads the serial number of each disk.
2. The serial number is hashed with the GK to create the access credential.
3. The access credential is sent to the drive.
4. In the drive, the encryption key is wrapped with the access credential. A hash of the access credential is also stored on the drive.

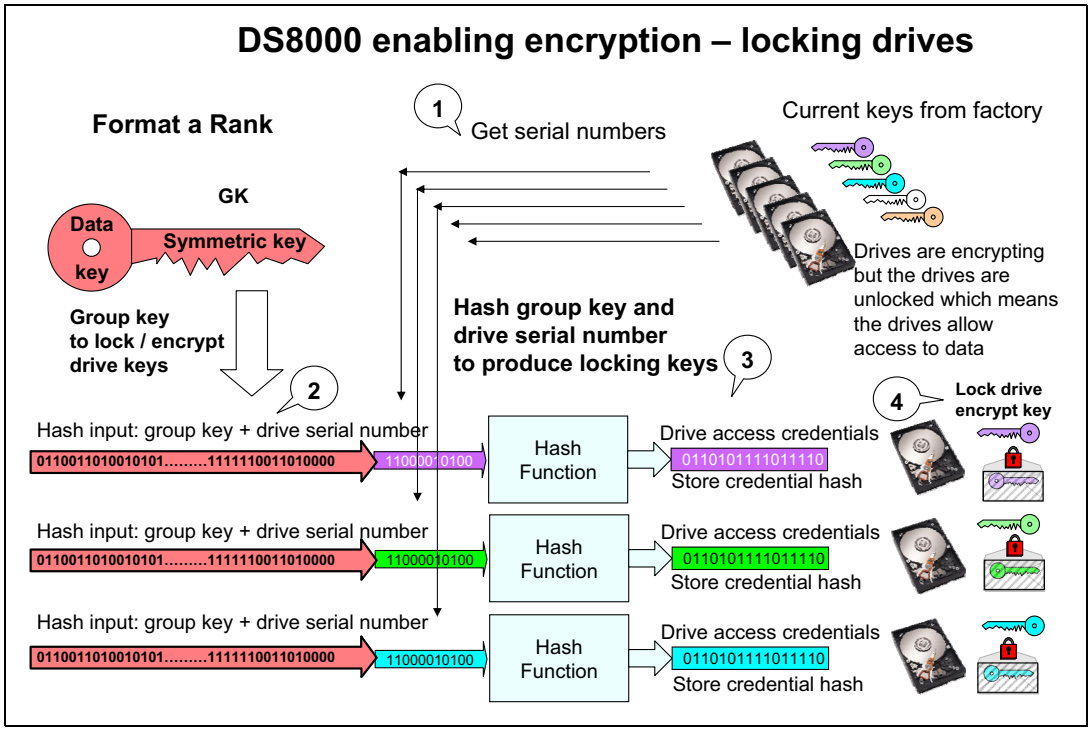


Figure 2-8 Set up encryption: Lock the drives

The drives are locked now, which means after a power-off and a power-on, the drives grant access to data only when the encrypted encryption key that is stored on the drives is unlocked by providing access credentials and an unlock key, as shown in Figure 2-9 on page 30.

Disk encryption details

Each FDE drive has an encryption key for the area of the drive that contains client data (Band 1). As shown in Figure 2-9 on page 30, Band 0 is for internal global data, which is also encrypted.

When the client data area is *unlocked*, the FDE drive still encrypts/decrypts the data with a data encryption key (DEK) and this DEK is also wrapped (encrypted) with access credentials. Here, a default encryption key is used to encrypt the DEK, but it is done transparently to the initiator (the DS8000). However, if someone takes the disk plate without the interface, trying to read from the disks is impossible because the data is encrypted.

The DEK for the data area is *wrapped* (encrypted) with an access credential that is produced with the GK. This access credential is converted to a secure hash and stored on the disk. At that stage, the client data area is *locked*.

After a disk power loss, the read/write access to the data on a locked area is blocked until the DS8000 is authenticated by supplying the currently active access credential, that is, the GK, as shown in Figure 2-9. (The DS8000 must first get unlock keys for the GK from the IBM Security Key Lifecycle Manager server). The following steps occur:

1. The disk drive verifies the access credentials (containing the GK).
2. The drive validates the access credentials with the one stored on the disk drive.
3. The drive reads the stored encrypted DK.
4. The encrypted DK is decrypted by using access credentials (GK).

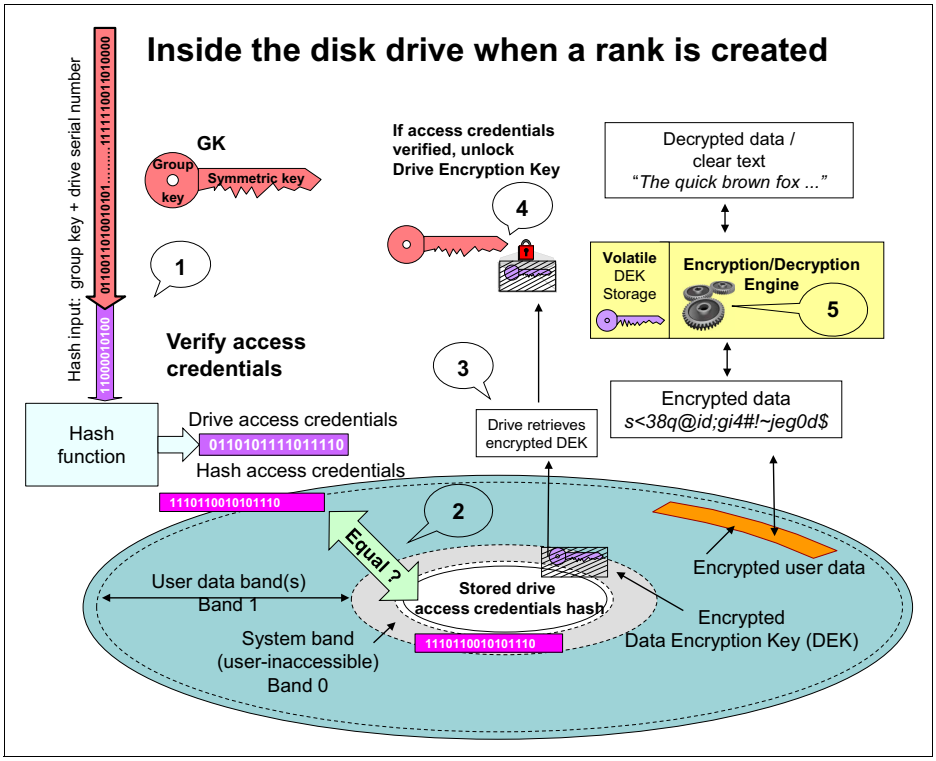


Figure 2-9 Unlock drives

An FDE drive that is made a member of an encryption-enabled rank is locked. The FDE drive is unlocked when it is unassigned, or is a spare. Locking occurs when an FDE drive is added to an encryption-enabled rank either during rank creation or sparing. Unlocking occurs when an encryption-enabled rank is deleted or a member of an encryption-enabled rank is reused as a spare. Unlocking always results in a cryptographic erasure of an FDE drive (the disk resets its own encryption key). This also happens when an encryption-disabled rank is deleted.

In a cryptographic erasure, a new DEK is generated in each disk drive. See Figure 2-10. The new key is encrypted with default access credentials, and both the access credentials and the encrypted DEK are stored on Band 0 of the drive. Now, the drive is unlocked. If someone tries to read the old data, nonsense data is returned because decryption now uses another key, which no longer decrypts the data.

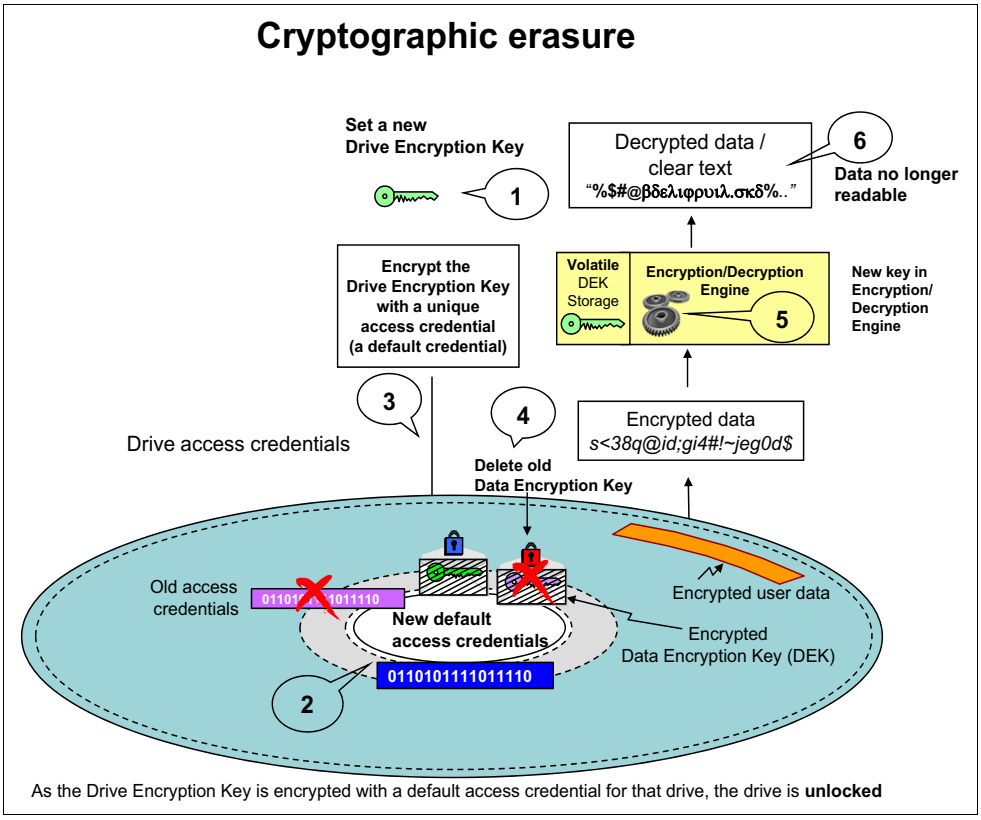


Figure 2-10 Cryptographic erasure

FDE drives are *not* cryptographically erased when a drive fails. In this case, there is no guarantee that the device adapter (DA) can communicate with the disk to cryptographically erase it. More specifically, the DA intentionally fences the failing drive from the device interface immediately to prevent it from causing other problems on the interface. However, because the currently active encryption key that still exists in the failed FDE drive is encrypted, the data is not readable.

Getting access to data after a power-on

After powering off and powering on, the DS8000 no longer has a DK or a GK in the clear, as shown in Figure 2-11. The DEKs in the drives are encrypted, the GK to unlock the drives is encrypted, and the DK to access to the GK keystore is encrypted. But, the DS8000 does not have access to all these keys. It must first get a key to unlock the DK from IBM Security Key Lifecycle Manager.

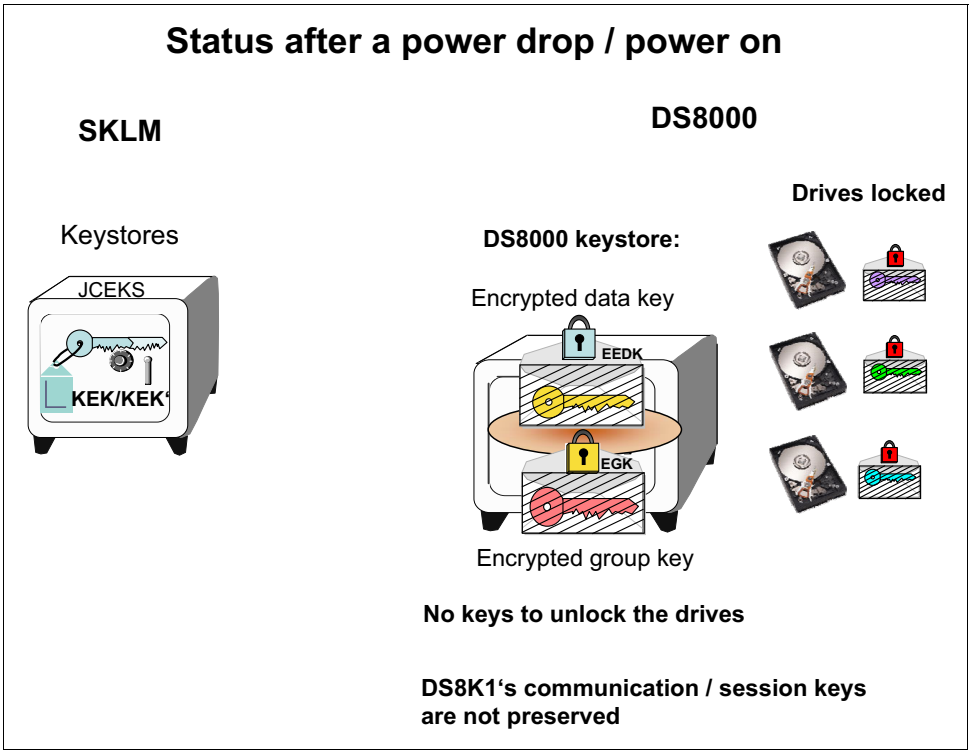


Figure 2-11 DS8000 status after a power off/on

The DS8000 must do the following steps (see Figure 2-12) to regain access to locked drives and data at power-on:

1. The DS8000 generates a new session key pair (private and public) to communicate with IBM Security Key Lifecycle Manager.

   Important: The DS8000 must be able to communicate with at least one IBM Security Key Lifecycle Manager server at power-on.

2. The DS8000 gets the EEDK from its keystore.
3. The DS8000 requests IBM Security Key Lifecycle Manager to unwrap an existing wrapped DK by sending the request to IBM Security Key Lifecycle Manager with the saved EEDK, the DSK, and DS8000 disk storage system's certificate.
4. The IBM Security Key Lifecycle Manager unwraps the EEDK with its key-label private key to obtain the DK.
5. The DK is wrapped with DS8000 disk storage system's DSK to create the SEDK.
6. The SEDK is returned to the DS8000.

7. The SEDK is decrypted with DS8000 disk storage system's DSK' to obtain the DK.
8. The DK is then used to unwrap the EGK to get the GK.
9. The serial number of the disk is read and hashed with the GK to obtain the access credential. The hashed access credential is sent to disk and the validity of the access credential is verified. If the access credential is valid, the disk encrypted DK is unwrapped to gain access to the data.

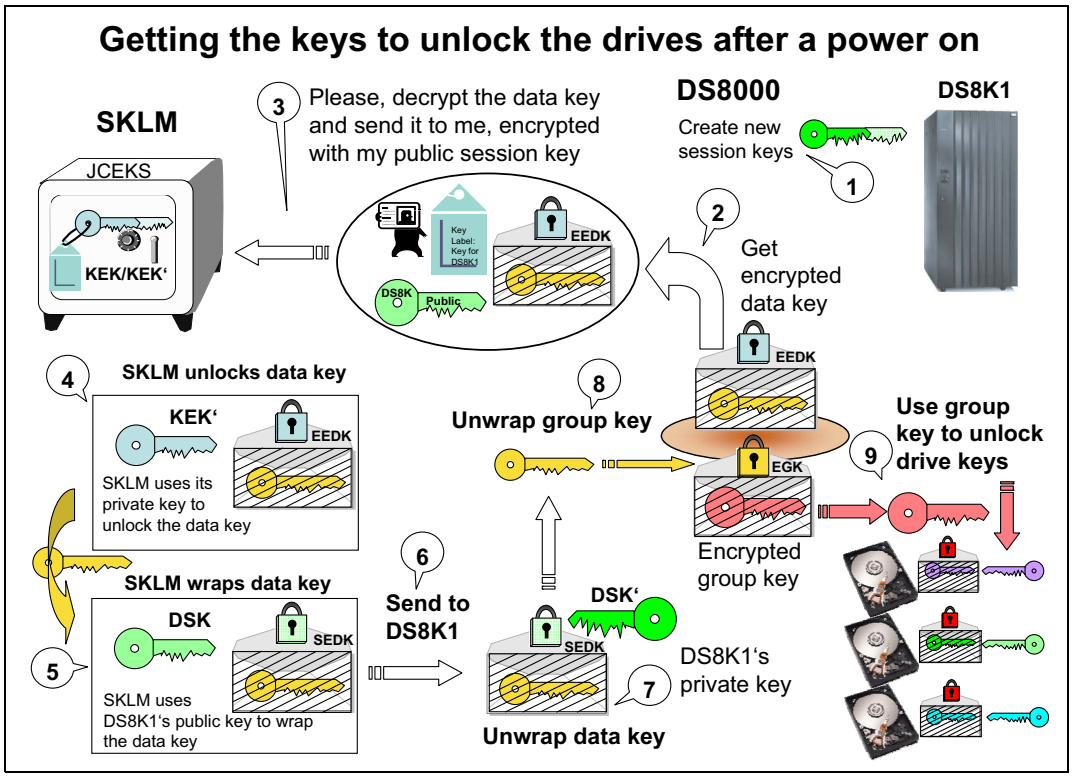


Figure 2-12 Steps to regain access to data after a power-on

2.3 SafeNet KeySecure key management with KMIP

This section has details about how the Gemalto SafeNet KeySecure key server manages and creates the encryption keys that are used by the DS8000 during encryption group and rank creation and at DS8000 power-on time.

Important: Key negotiation and authentication between the Gemalto SafeNet KeySecure and DS8000 take place at DS8000 encryption configuration and power-on time only. There is no increased traffic use in an encrypted DS8000 at run time that is created by key negotiation.

The Gemalto SafeNet KeySecure key server uses the direct key method to serve keys to encryption-enabled DS8000 while IBM Security Key Lifecycle Manager is using the wrapped key method.

In the direct key model, the DK is created and stored on the external key server, upon request from the DS8000. It is created and registered on the key server. When the storage device requires the DK for its cryptographic purposes, the storage device requests the key and the key server delivers it.

A single DK is associated to the DS8000. During a rekey, the DS8000 requests that a new DK be created by the key server.

The DS8000 authenticates itself with a root SSL certificate at the key server, or, for more security, with the root SSL certificate and a user ID, which are created during manufacturing and set into the SSL certificate on the DS8000.

For more information about how to set up the Gemalto SafeNet KeySecure key secure servers with user ID and SSL certificates for DK delivery, see 4.3, “Configuring Gemalto SafeNet KeySecure with KMIP” on page 79.

Now, the user (Storage Administrator) can use the DS8000 GUI or CLI to register the key server on the DS8000. For more information about how to perform this action, see 4.4, “DS8000 GUI configuration for encryption” on page 96.

Next, still using the DS8000 GUI, an encryption group must be created. As part of creating the encryption group, you must specify the encryption protocol KMIP. For more information about how to perform this action, see 4.4, “DS8000 GUI configuration for encryption” on page 96.

Note: Currently, the DS8000 supports only one encryption group.

The following steps take place during encryption group creation:

1. Creation of an encryption group is requested by the user.
2. The DS8000 requests a DK generation from one of the previously created key servers.
3. The key server generates the DK and replicates it to all other key servers in the cluster, which can be up to four. It stores the key in an encrypted database.
4. The key server returns a Unique Universal Identifier (UUID) back to the DS8000.
5. The DS8000 requests the DK from the key server by using that UUID that is received during the generate key request.

Note: The UUID is a random 64-byte unique identifier with no relationship to the DK. It is created during initial encryption group creation when requesting the DK for the first time and used for identification.

6. The key server returns the DK, secured by TLS/SSL, to the DS8000.
7. The DS8000 creates the GK and wraps it with the DK to get an EGK.
8. The DS8000 stores the encrypted GK, the UUID, and the protocol information (KMIP) in its KR.
9. The DS8000 temporarily stores the DK in protected memory but not on disk.
10. The DS8000 request to retrieve the DK from all configured key servers and compares it with the one in memory for verification. At least two of the configured key servers must return the correct DK.
11. The DS8000 deletes DK and the GK from local (working) memory after successful verification.

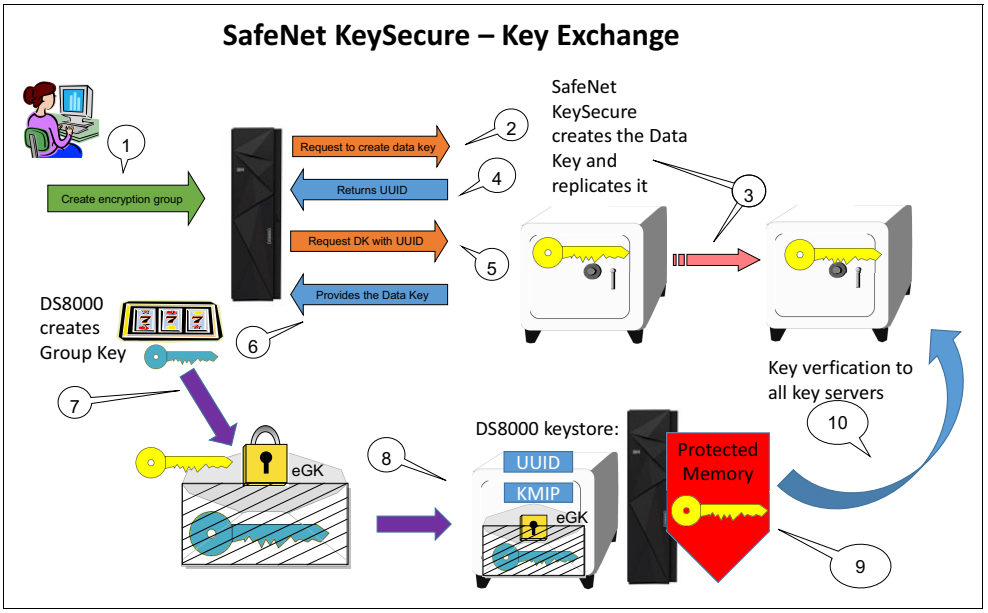


Figure 2-13 Gemalto SafeNet KeySecure Key exchange

After the encryption GK is created, the remaining drive encryption steps are the same ones being used with IBM Security Key Lifecycle Manager because the internal DS8000 encryption mechanism did not change. For more information, see 2.2, “IBM Security Key Lifecycle Manager Encryption key management” on page 24. Cryptographic erase, rekey, and recovery key (RK) usage and actions are also mostly unchanged.

Getting access to data after a power-on

After powering off and powering on, the DS8000 no longer has a DK or a GK in the clear. The DEKs in the drives are encrypted, the GK to unlock the drives is encrypted, and the DK to get access to the GK in keystore is encrypted. But, the DS8000 does not have access to all these keys. It must first get the unlock DK from Gemalto SafeNet KeySecure key server.

As the DK is now stored encrypted in the key server, the DS8000 asks during power-on for the DK and authenticates itself with the UUID and its certificate. The key server then provides the key and the DS8000 can unlock the encrypted GK and continues to power-on.

2.4 Encryption deadlock

The key server platform provides the operating environment for the key server application to run in, to access its keystore on persistent storage, and to interface with client storage devices, such as the DS8000 that requires key server services.

The keystore data is accessed by the key server application through a password that is specified by the client. As such, the keystore data is encrypted at rest, independently of where it is stored. However, any online data that is required to initiate the key server must not be stored on a storage server that depends on the key server to enable access. If this constraint is not met, the key server cannot complete its initial program load (IPL) and does not become operational.

This required data includes the boot image for the operating system that runs on the key server and any other data that is required by that operating system and its associated software stack to run the key server application to allow the key server to access its keystore, and to allow the key server to communicate with its storage device clients. Similarly, any backups of the keystore must not be stored on storage that depends on a key server to access data.

Not strictly following these implementation requirements might result in the situation where the encrypted data can no longer be accessed either temporarily, or worse, permanently. This situation is referred to as *encryption deadlock*.

Important (encryption deadlock): Any data that is required to make the IBM Security Key Lifecycle Manager key server operational must *not* be stored on an encrypted storage device that is managed by this particular key server. Again, this situation is referred to as an *encryption deadlock.* This situation is similar to having a bank vault that is unlocked with a combination and the only copy of the combination is locked inside the vault.

The differences between a temporary encryption deadlock and a permanent encryption deadlock are as follows:

- Temporary encryption deadlock

  The temporary encryption deadlock indicates a situation where the DS8000 cannot access its disk devices because IBM Security Key Lifecycle Manager servers are not online, the network is down, or for any other temporary hardware-related errors. This temporary failure can be fixed at the client site.

- Permanent encryption deadlock

  This permanent encryption deadlock is the worse case. Here, all key servers that manage some set of data cannot be made operational because they depend on inaccessible encrypted storage, or all encrypted online and offline data that is managed by the set of key servers is, in effect, cryptographically erased and for all practical purposes permanently lost.

When considering encryption in your environment, note the following factors:

- As the availability of encryption-capable devices becomes more pervasive, more data is migrated from non-encrypted storage to encrypted storage. Even if the key servers are initially configured correctly, it is possible that a Storage Administrator might accidentally migrate some data that is required by the key server from non-encrypted to encrypted storage.
- Generally, a number of layers of virtualization in the I/O stack hierarchy can cause difficulties for the client to maintain awareness of where all the files (necessary to make

the key server, and its associated keystore, available) are stored. The key server can access its data through a database that runs on a file system that runs on a logical volume manager, which communicates with a storage subsystem that provisions logical volumes with capacity that is obtained from other subordinate storage arrays. The data that is required by the key server might end up provisioned over various storage devices, each of which can be independently encryption-capable or encryption-enabled.

- Consolidation of servers and storage tends to drive data migration and tends to move increasingly more data under a generalized shared storage environment. This storage environment becomes encryption-capable as time goes on.
- All IBM server platforms support fabric-attached boot devices and storage. Some servers do not support internal boot devices. Therefore, boot devices are commonly present within the generalized storage environment. These storage devices are accessible to generalized storage management tools that support data management and relocation.

To mitigate the risk of an encryption deadlock, a stand-alone IBM Security Key Lifecycle Manager server (also called an Isolated Key Server) is mandatory and the client must be directly involved in managing the encryption environment. For more information, see Chapter 3, “Planning and guidelines for IBM DS8000 encryption” on page 45 and Chapter 4, “IBM DS8000 encryption implementation” on page 55.

2.5 Working with a recovery key

To get out of a deadlock situation or, as a recovery option if all key servers are destroyed and unrecoverable, the DS8000 allows you to create a *recovery key* (RK). With an RK, a Security Administrator can unlock a DS8000 without the involvement of a key server. It is also possible to *disable* RK management.

Important (creating or disabling an RK): An RK can be created only during the encryption enablement process. You cannot create an RK when a DS8000 is already configured as encrypted. Similarly, disabling the RK management is allowed only for an unconfigured DS8000. Creating or disabling an RK must be one of the first actions when setting up the DS8000 for encryption.

Managing the RK requires two people (roles): A Storage Administrator (admin) and a Security Administrator (secadmin). The Security Administrator is a new role for DS8000 users. A Storage Administrator cannot create a Security Administrator user on a DS8000 and vice versa. The Security Administrator maintains the RK and keeps it safe; the Storage Administrator has to approve every action of the Security Administrator.

Client responsibility: Although DS8000 supports two roles, Storage Administrator and Security Administrator, the client is responsible to assign these roles to two *separate* individuals.

2.5.1 Recovery key management

This section summarizes the actions that are allowed in a RK-enabled scenario.

Creating a recovery key

Setting up an RK involves the following steps, which are shown in Figure 2-14:

1. The Security Administrator user requests the creation of an RK, which can be done with the DS CLI or the GUI. This request function is not available to other users.

2. At some stage in the process, the Storage Administrator must approve the action that the Security Administrator is going to perform.
3. Having obtained the request to generate an RK, the DS8000 generates a random 256-bit RK.
4. The DS8000 generates a secure hash of the RK producing the recovery signature (RS).
5. The storage facility generates an asymmetric public / private key pair from a random 2048-bit number. The private key is referred to as the primary recovery key (PRK) and the public key is referred to as the secondary recovery key (SRK).
6. Next, the DS8000 wraps the PRK with the RK to produce the encrypted primary recovery key (EPRK).
7. The EPRK, the SRK, and the RS are stored in multiple places within the storage facility for reliability.
8. The storage facility provides the RK to the Security Administrator. The system follows a verification process, which is not further detailed here (the Security Administrator must reinput the RK).
9. The DS8000 deletes the PRK and the RK.

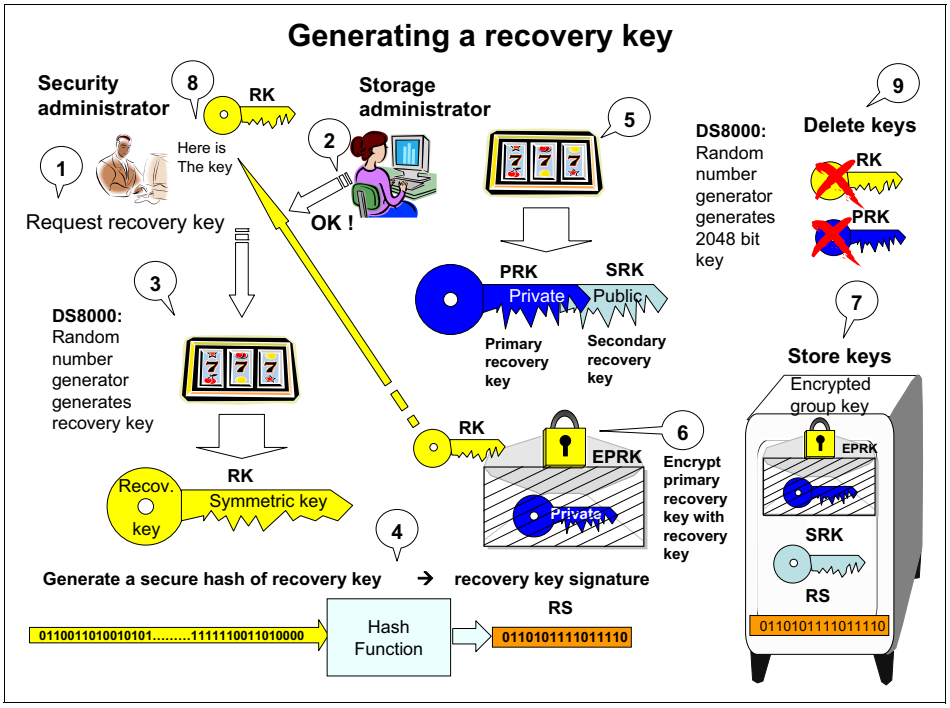


Figure 2-14 Generate a recovery key

When you configure an encryption group with an RK defined, you must complete other steps in addition to those shown in Figure 2-7 on page 28 (see Figure 2-15):

1. The storage facility wraps the GK with the SRK to produce the encrypted group recovery key (EGRK).
2. The EGRK is stored together with the EPRK and the other encrypted keys (EEDK and EKG) in the DS8000 keystore.

After the encryption group is configured, ranks can be created and assigned to the encryption group.

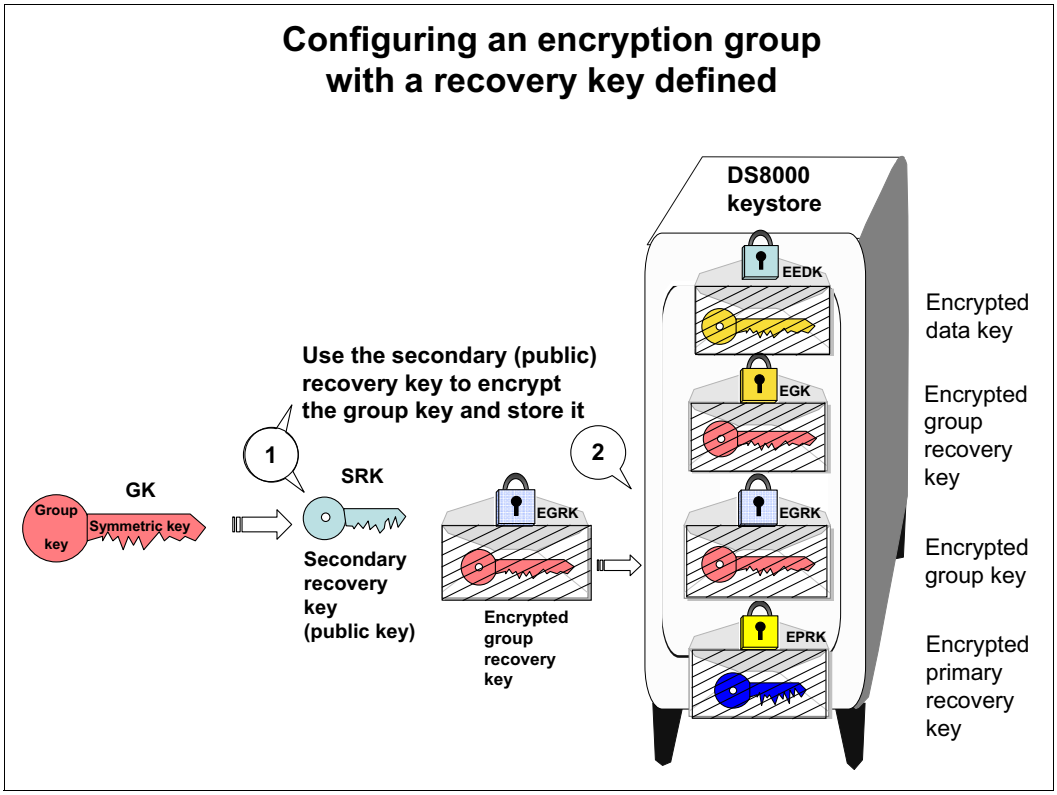


Figure 2-15 Set up encryption with a recovery key defined

Using a recovery key to unlock a DS8000

If the DS8000, after a power-off and power-on, cannot obtain the required DK from a key server, it attempts to contact all other configured IBM Security Key Lifecycle Manager servers to obtain the required key.

On a DS8000 with an RK configured, an option exists to let a Security Administrator input the RK.

If the Security Administrator provides the RK and the Storage Administrator approves this operation, the DS8000 uses the RK to unwrap the “EPRK to obtain the PRK (see Figure 2-16):

1. The DS8000 cannot communicate with any IBM Security Key Lifecycle Manager server.
2. The DS8000 can ask for an RK.
3. The Security Administrator enters the RK.
4. The Storage Administrator approves the action.
5. The RK is used to unlock the PRK.
6. The PRK is used to unlock the GK.
7. The GK is used to unlock the drives.

Now, access to data is restored.

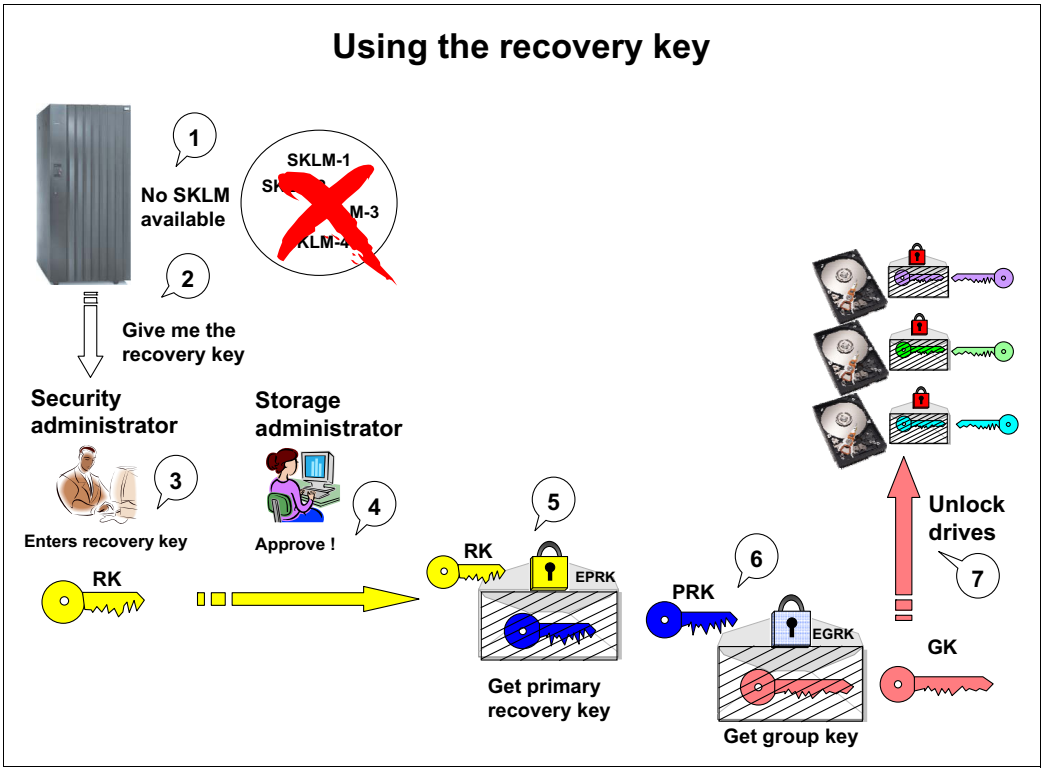


Figure 2-16 Use the recovery key

Changing the recovery key

The DS8000 also supports functions to rekey, verify, and unconfigure an RK.

The rekey and verify RK functions can be performed at anytime while the RK is configured and a key manager server is available. Access to a key manager server is required. It allows the DS8000 to verify that it is in the correct environment. Only when the key manager can decrypt the DK can the DS8000 be sure that it is in the same environment (see Figure 2-17). Only then, it generates a new RK. For example, on a DS8000 that was stolen and put in a separate environment, rekeying the RK is not possible.

During the rekey operation, the following steps are done:

1. The DS8000 sends the EEDK and its public key to IBM Security Key Lifecycle Manager and requests a rekey validation.
2. IBM Security Key Lifecycle Manager tries to decrypt the DK.
3. If IBM Security Key Lifecycle Manager can decrypt the DK, it signals the DS8000 that it can proceed to generate a new RK.
4. The DS8000 generates a new RK.

Changing the RK does not erase the data.

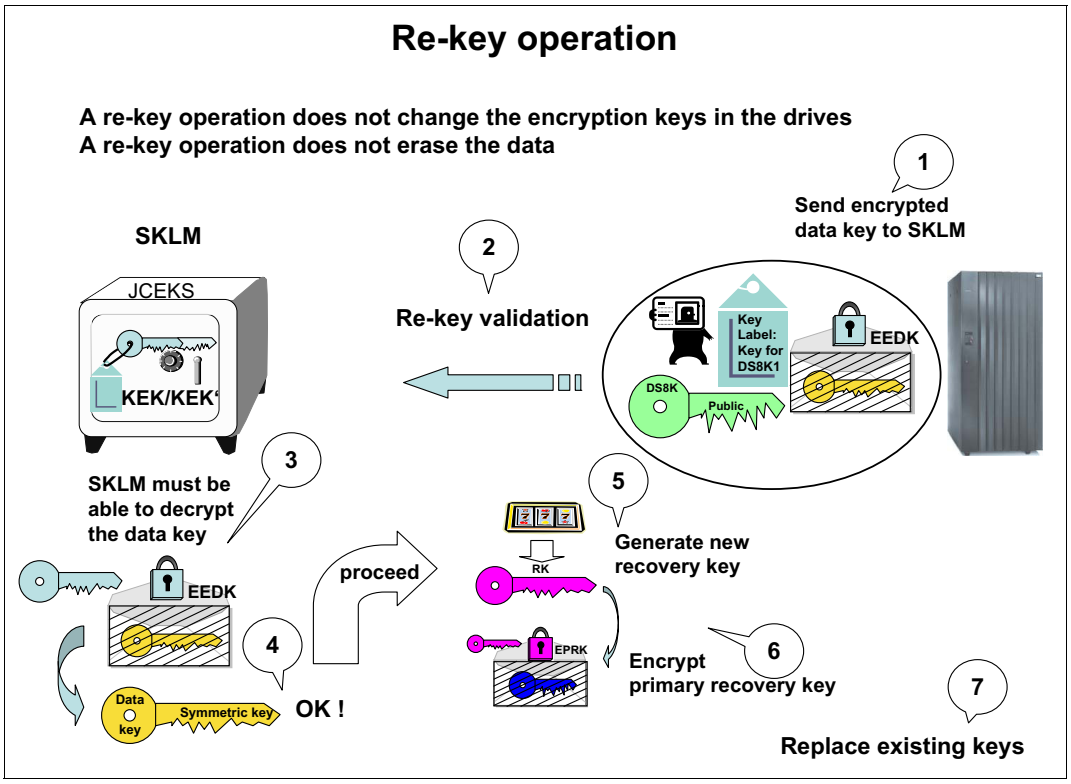


Figure 2-17 Rekey the recovery key

2.5.2 Disabling or enabling a recovery key

The RK can be optionally disabled, or configured, before establishing an initial logical configuration. This section describes the actions that are required for disabling or enabling the RK.

Disabling a recovery key

If you do not want to manage an RK in your environment, it can be disabled. However, this must be done as your first action before defining the encryption group. The process of disabling the RK is as follows:

1. The Security Administrator (secadmin) requests the disabling of an RK. This can be done with the DS CLI or the GUI. This request function is available only to a user with the Security Administrator role.
2. The Storage Administrator (admin) approves the disabling status.
3. The RK goes into the *disable* state.

The encryption group can now be defined.

These actions are illustrated in 4.5, “Command-line configuration for DS8000 encryption” on page 125.

Enabling a recovery key

When an RK is disabled, it can later be reenabled. This action is disruptive. All data (on the DS8000) must be erased as a prerequisite.

Enabling the RK management involves the following steps:

1. The Security Administrator user requests the enabling of a disabled RK. This can be done with the DS CLI or the GUI. This request function is not available to other users.
2. The Storage Administrator approves the enabling status.

The RK can now be created as described in “Creating a recovery key” on page 37.

For an illustration of these actions, see 4.5, “Command-line configuration for DS8000 encryption” on page 125.

2.6 Dual key server support

The DS8000 supports the configuration of either one or two key labels for the encryption group.

When all key server platforms operate their key stores in clear-key mode or when only a single host platform is used for all key servers, a single key label is typically sufficient to allow all key severs to interoperate with the DS8000. In this case, it is possible for the asymmetric key pair that is maintained for the key label by the IBM Security Key Lifecycle Manager to be propagated across all supporting key servers so that each key server has the necessary keys to wrap and unwrap the one EEDK that is maintained on the DS8000.

When there are two key server platforms and at least one of the key server platforms is operating in secure key mode (which is available on the z/OS platform), a second key label is typically required.

Note: Having a key server platform in secure key mode on z/OS platform for the DS8000 is not common at all. Typically, the z/OS runs on volumes that are on a DS8000, which is encrypted. This increases the chance of running into a deadlock situation. Having a second DS8000 that is not encrypted to run the key server is not common either.

A key server operating in secure key mode typically does not support the export of any private keys outside of the key server platform. In this case, the following actions are performed to synchronize keys between key servers (see Figure 2-18):

- Key label 1 (with public and private key) is configured on a UNIX platform.
- Key label 2 (with public and private key) is configured on z/OS platform.
- The public key from key label 1 is exported to the IBM Security Key Lifecycle Manager for z/OS (abbreviated as SKLM-z/OS in the figures).
- The public key from key label 2 is exported to platform IBM Security Key Lifecycle Manager for UNIX (abbreviated as SKLM-UNIX in the figures).

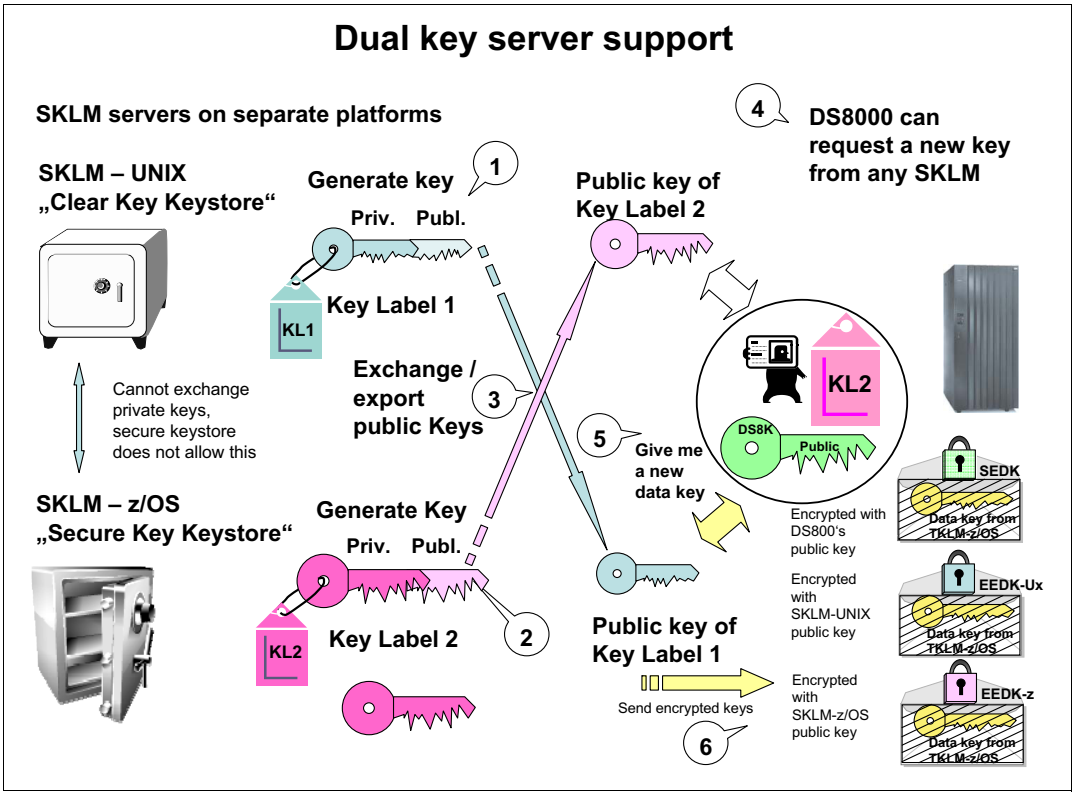


Figure 2-18 Dual key server support

Now, all key servers on both platforms have the *public keys for both key labels* and the *private of one or the other key label*. The DS8000 can request a new key from *any* key server and store an EEDK associated with each key label. Therefore, the DS8000 has two separate EEDKs now.

Each IBM Security Key Lifecycle Manager has a “public key” of each key label so it can generate two EEDKs.

The following steps summarize the example that is shown in Figure 2-18:

1. IBM Security Key Lifecycle Manager for UNIX creates a public/private key pair for key label 1.
2. IBM Security Key Lifecycle Manager for z/OS creates a public/private key pair for key label 2.
3. Both Security Key Lifecycle Managers exchange their public keys.

4. A DS8000 can request a DK from any IBM Security Key Lifecycle Manager. For this example, assume it requests the DK from IBM Security Key Lifecycle Manager for z/OS.
5. IBM Security Key Lifecycle Manager for z/OS generates the DK, wraps it with the DS8000 disk storage system's public key to produce the SEDK and wraps the DK with its own public key to produce external encrypted data key with z (EEDK-z), and wraps the DK with IBM Security Key Lifecycle Manager for the UNIX public key to produce external encrypted data key with UNIX (EEDK-Ux). Then, the SEDK, the EEDK-z, and the EEDK-Ux are sent to the DS8000.

The DS8000 can request the EEDKs to be unwrapped by any key server because the request contains both EEDKs (see Figure 2-19), and any key server has the private key for at least one of the two EEDKs in the request. Secure key mode operation is maintained during the exporting of secure keys because only the public key is exported.

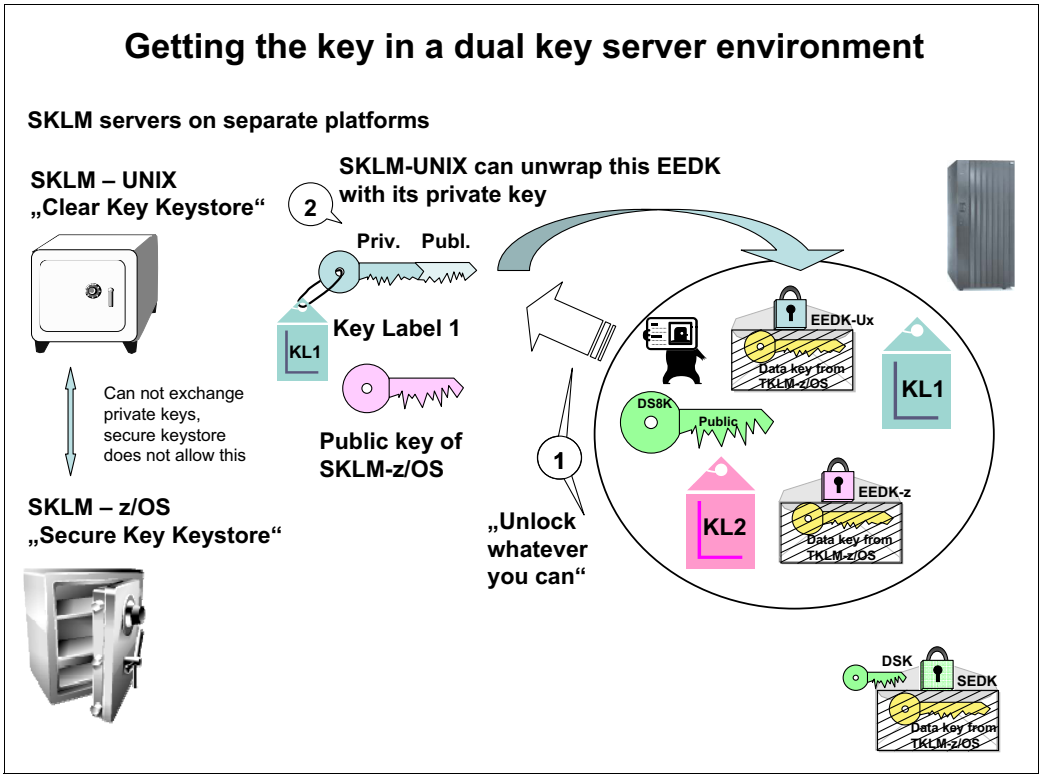


Figure 2-19 Get the recovery key in a dual server environment

In the example that is shown in Figure 2-19, the DS8000 sends its request to decrypt the DK to IBM Security Key Lifecycle Manager for UNIX with both key labels and both EEDKs. IBM Security Key Lifecycle Manager for UNIX can decrypt the EEDK-Ux that is associated with key label 1.

IBM Security Key Lifecycle Manager for UNIX can now send the encrypted DK back to the DS8000.

3

Planning and guidelines for IBM DS8000 encryption

This chapter provides information about planning for an IBM DS8000 encryption-capable storage system. It covers the following topics:

- Planning and implementation process flow
- Encryption-capable DS8000 ordering and configuration
- Licensing
- Requirements for encrypting storage
- Advice for encryption in storage environments
- Multiple IBM Security Key Lifecycle Managers for redundancy

3.1 About certificates

DS8000 Release 7.2 implemented new security features that must be considered when you plan to implement disk encryption or migrate an existing encryption environment in the past.

This situation changed in DS8000 Release 8.1. There are two different certificates that are supported on DS8000 Release 7.2 and later, called *Gen-1* and *Gen-2*:

- Gen-1 certificates have 80-bit security strength and have been in use since disk encryption was introduced in DS8000 Release 4.2 in the 5.4.21.xx Licensed Managed Code (LMC) code levels.
- Gen-2 certificates have 112-bit security strength and were introduced in Release 7.2 in the 7.7.20.xx LMC levels. The Gen-2 certificates meet the requirements of the NIST Special Publication 800-131a: *Transitions Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*. In addition, Gen-2 certificates in DS8000 Release 8.1 have a UID added to the Client Certificate Authentication, which enables the most secure way to connect a Key Management Interoperability Protocol (KMIP) capable key server, such as Gemalto SafeNet KeySecure, to the DS8000 by SSL session and a specific user name.

The Gen-1 certificates are supported for all machines in the DS8000 series except DS8000 machines that were shipped with Release 8.1 and newer. The Gen-2 certificates are supported on DS8000 Release 7.2 disk storage systems and later and DS8000 machines that were updated from Release 8.0 to Release 8.1.

Careful planning is required when selecting which certificate is used, especially when migrating an existing encryption environment.

DS8870 Release 7.2 and later also supports Transport Layer Security (TLS) 1.2 for network communication. All components in the storage environment must support TLS 1.2 before implementing the Gen-2 certificates and TLS 1.2. DS8000 Release 8.1 does not support Tivoli Key Lifecycle Manager Version 2.1; upgrading to its replacement, IBM Security Key Lifecycle Manager V2.6, is necessary because you need TLS 1.2 support.

3.2 Planning and implementation process flow

Figure 3-1 shows the planning and implementation process for an encryption-capable DS8000. The details for this process are described in subsequent sections of this chapter. This diagram shows the overall decision flow and outcomes.

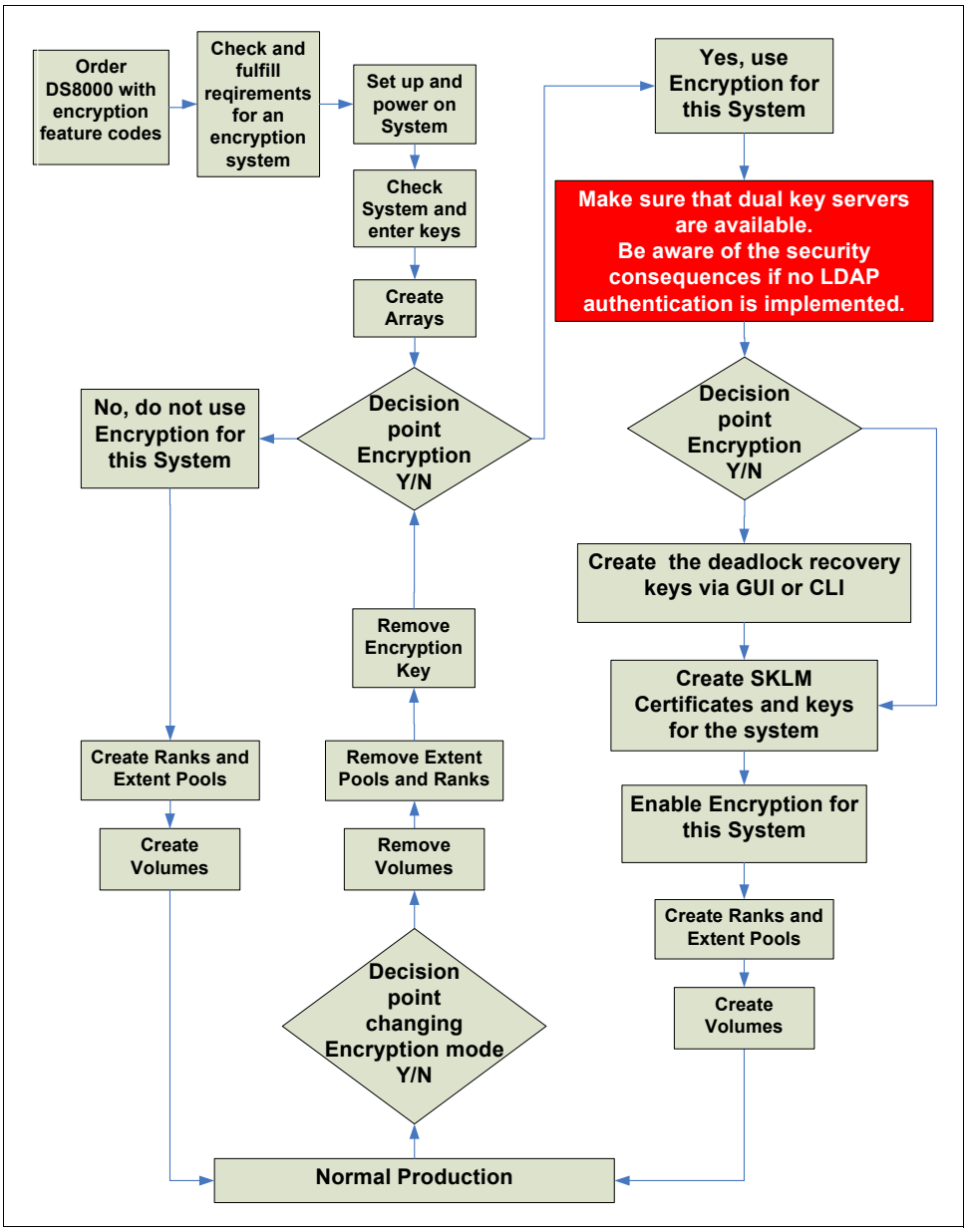


Figure 3-1 Encryption implementation planning flow

3.3 Encryption-capable DS8000 ordering and configuration

To enable encryption on a DS8000, the Full Disk Encryption (FDE) feature is required. The DS8000 is equipped with FDE drives by default.

When you plan to use IBM Security Key Lifecycle Manager as your external key server, two options are available for the required isolated key servers:

- Single/dual storage appliance server: Feature codes 1761 and 1762 are available in Machine type Model 2421-AP1. New storage appliance server options are available in addition to the existing IBM Security Key Lifecycle Manager isolated key manager (feature code #0204).
- Client acquired isolated key server: For the hardware and operating system requirements, see *IBM Security Key Lifecycle Manager Installation and Configuration Guide*, SC27-5335.

You must still do the specific tasks to enable or disable encryption, as described in Chapter 4, “IBM DS8000 encryption implementation” on page 55.

Country requirements before proceeding with the steps: In certain countries, clients might be required to sign an Import Agreement to import or export FDE drives.

Complete the following steps:

1. After the ordering and verification process is complete, IBM delivers the DS8000, the IBM service representative installs the DS8000, and IBM provides you with an Encryption Authorization Licensed Internal Code feature key (or keys) for the DS8000. Each key is unique to the DS8000 for which it is generated. If you want encryption, be sure to include feature code 1761 or 1762 in your order. With this feature, the client can use the Disk Storage Feature Activation (DSFA) website to download the function authorization and turn on encryption. If you want encryption, enable it at first use.
2. After the storage system is made available to the client, the deadlock recovery key must be created by using the GUI or DS system CLI.
3. Now that the system can be recovered at any time by using the recovery key, set up the IBM Security Key Lifecycle Manager connection and make sure that it is functioning.

   After this step is complete, the system is fully enabled and activated for encryption.
4. You can now configure ranks or extent pools, as described in 4.5.7, “Creating encrypted extent pools” on page 135.

Notes:

- All ranks and extent pools on a particular encryption-capable DS8000 must be configured with the same encryption group attribute. The first rank or encryption group that is configured determines how the remaining objects must be configured. A value of zero indicates that encryption is disabled. A value other than zero indicates that encryption is enabled.
- To change between encryption-enabled and encryption-disabled, all ranks and extent pools must be unconfigured. Unconfiguring an encryption-enabled rank causes any data that was stored on the rank to be cryptographically erased and, later, overwritten to reinitialize the rank.

3.4 Licensing

When ordering the IBM Security Key Lifecycle Manager, two things must be considered:

- The quantity of IBM Security Key Lifecycle Manager servers per data center
- The quantity of drives to be encrypted

The general license for IBM Security Key Lifecycle Manager Version 2.6 is one server per data center. Each license includes active and warm backup. If a server is actively serving keys, then it needs a license.

The size of the license is based on the number of disk drives to be encrypted per DS8000.

After you license the software, it is available for download through IBM Passport Advantage®.

3.5 Requirements for encrypting storage

To deploy an encryption-capable DS8000, strictly adhere to the following requirements:

- Configuring the deadlock recovery key is the recommended first step. If you do not want to set a recovery key, you must disable it now, as described in 2.5.1, “Recovery key management” on page 37 and 2.5.2, “Disabling or enabling a recovery key” on page 41. The key is generated from the DS GUI or DS CLI. The DS8000 *secadmin* and *admin* users must cooperate to establish the encryption deadlock recovery key.
- Any DS8000 that is used with encryption must be configured to at least one isolated key server.

  This key server can be configured to serve keys to any device that IBM Security Key Lifecycle Manager supports, including other encryption-enabled DS8000 disk storage systems or supported IBM tape drives.

  The isolated key server is hardware that must be purchased separately. The customer is responsible for purchasing the software licenses that are loaded on to the storage appliance 2421 model AP1. The customer is also responsible for updating both the operating system and software whenever updates are required. The SUSE Linux Enterprise Server operating system is shipped as the default operating system for isolated key servers that are ordered with Machine Type 2421-AP1 (IBM Storage Appliance) and feature codes 1761/1762. Customers who order feature number 0206 must purchase their own operating system license.

  The IBM Security Key Lifecycle Manager software is purchased separately from the isolated key server hardware. The IBM Security Key Lifecycle Manager servers are delivered through MT 2421-AP1 IBM Storage Appliance. Feature Code 1762 provides two IBM Security Key Lifecycle Manager isolated servers as a minimum requirement for DS8000 encryption enablement.

  DS8000 Release 7.2 introduces support for IBM Security Key Lifecycle Manager Version 2.5 and later to support the NIST SP 800-131a encryption specification. IBM Security Key Lifecycle Manager is the next generation of key server from the Tivoli Key Lifecycle Manager. You must have an IBM Security Key Lifecycle Manager license to use the IBM Security Key Lifecycle Manager software. You must order the license separately from the stand-alone server hardware.

You can also provision your own hardware, as described in 1.4.5, “IBM Security Key Lifecycle Manager for open systems” on page 14. For more information, consult the IBM Security Key Lifecycle Manager V2.6 documentation:

http://ibm.biz/SKLMv26KC

- An encryption-enabled DS8000 requires at least two key servers to be configured (the isolated server and a backup server). The key servers can be shared by more than one DS8000.

3.6 Advice for encryption in storage environments

The following information can help you find the preferred practices for encryption in storage environments. It includes key techniques for mitigating the risk of an encryption deadlock.

3.6.1 Using LDAP authentication

Ideally, a preferred practice is to manage the physical security of access to hardware through an LDAP implementation. This approach allows a close monitoring of who, when, and what actions were taken by monitoring the audit logs of the DS8000. With a basic security policy, having a single person who handles the *admin* and *secadmin* role of a DS8000 is still possible. With LDAP, a policy can be set up that does not allow having the same user ID for both roles in the DS8000.

3.6.2 Availability

Keep these considerations and preferred practices in mind:

- DS8000:

  The DS8000 must be configured with the dual Hardware Management Console (HMC) option to provide redundant access to the client network.

- IBM Security Key Lifecycle Manager key server:
  - Configure redundant key servers to each encrypting storage device. The client should have independent and redundant key servers on each site.
  - To initiate the IBM Security Key Lifecycle Manager key server operation after start without human intervention, the key server must be set up to start automatically when power is available and to initiate automatically the key server application. The application must be configured to boot automatically, especially when running the key server in a virtualized environment.

3.6.3 Encryption deadlock prevention

Keep the following considerations and preferred practices in mind:

- General:
  - The change management processes at your installation must cover any procedures that are necessary to ensure adherence to guidelines that are required to ensure correct configuration of key servers, encrypted storage, and placement of data that is related to key servers.
  - All personnel who have any of the following assignments or capabilities are required to review at least annually a client document that describes these risks and the processes that are adopted to mitigate them:
    - Responsibility to implement IBM Security Key Lifecycle Manager key servers or encrypted storage products.
    - Responsibility to manage the placement or relocation of data that is related to, or required by, any IBM Security Key Lifecycle Manager key server.
    - Access authority to configure IBM Security Key Lifecycle Manager key servers or encrypted storage products.
    - Responsibility to rekey the deadlock recovery key of the DS8000, if used.
  - You must implement automated monitoring of the availability of any equipment that is associated with management of key services and take appropriate action to keep them operational. This equipment can include but is not limited to key servers, SNMP masters, domain name servers, and DS8000 HMCs.
  - The client must pay particular attention to disaster recovery plans and scenarios and consider the availability of key servers, key server backups, and key server synchronization. A preferred practice is to establish the independence of each recovery site from the other recovery sites.
  - If the recovery key management is enabled, the client must have a documented process to handle and maintain the deadlock recovery keys of each DS8000 instance. This key is the last resort to unlock the DS8000 if the IBM Security Key Lifecycle Manager environment is destroyed or inaccessible. The deadlock recovery key is not used while IBM Security Key Lifecycle Manager remains available.
- IBM Security Key Lifecycle Manager key server:
  - Configuration of redundant key servers (at least two) is required. Redundancy implies independent servers and independent storage devices. For key servers operating in logical partitions (LPARs), do not use data-sharing techniques that result in one copy of the data being shared by multiple instances of the key server.
  - Configuration of one key server with dedicated hardware and non-encrypted storage resources at each recovery site is required. The key server is referred to as the *isolated key server*.

> **Two key servers:** The DS8000 requires at least one isolated key server to be configured, but it is a preferred practice to use two for redundancy.

The objective of this requirement is to avoid encryption deadlock by the following tasks:

- Implementing a key server environment that is independent of all non-key server applications so that management of the key server can be restricted to those personnel that are specifically authorized to manage key servers.
- Implementing a key server that is physically and logically isolated from other applications that might require access to encrypting storage so that the key server environment does not need to be configured with access to any encrypting storage.
- Implementing a key server that is physically and logically isolated from encrypting storage so that the risk of storing (initially or through data migration) code and data objects that are required by the key server on encrypting storage is eliminated.
- Ensuring that a recovery site can operate independently from any other sites by configuring a key server that is not subject to encryption deadlock because of the characteristics of an isolated key server.

– Configuration of additional key servers on generalized server hardware and generalized storage is allowed. Be sure to establish the appropriate procedures and controls to prevent these key servers from having your data access compromised by storing the data on key server managed encrypting storage. These key servers are referred to as *general key servers*.

– Configuration of key servers at independent sites is a preferred practice and provides additional immunity to encryption deadlocks because it reduces the probability that all key servers experience a simultaneous power loss.

– Clients must ensure that all key servers that a particular storage device is configured to communicate with have consistent keystore content relative to any wrapping keys that are used by the storage device. Failure to synchronize the keystores effectively eliminates one or more key servers from the set of redundant key servers for a storage device that uses the keys that are not synchronized.

– Clients should back up key server data after it is updated. The backups should not be stored on encrypted storage media that depends on a key server. For more information, see 5.1, “Rekeying the data key” on page 150.

– Clients should periodically audit to ensure that all online and backup data that is required to make each key server operational is stored on storage or media that does not depend on a key server to access the data.

– Clients must not delete keys on the key server under normal circumstances. Deletion of all copies of a key is a cryptographic erase operation of all encrypted data that is encrypted under this key.

► DS8000:

– Before any IBM Security Key Lifecycle Manager server is connected to the DS8000, run the deadlock recovery key generation process.

– *Suggestion:* Manually configure DS8000 devices on the IBM Security Key Lifecycle Manager key server. The option to configure them automatically can be used, but it increases the risk that an unauthorized DS8000 might gain access to a key server.

– The DS8000 supports up to four IBM Security Key Lifecycle Manager key server ports. A requirement is that at least one port is assigned to one isolated key server. A preferred practice is to assign two ports to isolated key servers. Using key servers at the local site should be preferred to improve reliability.

– When the DS8000 is configured to enable encryption, the DS8000 verifies that at least two IBM Security Key Lifecycle Manager key servers are configured and accessible to the machine.

– The DS8000 rejects the creation of ranks and extent pools with a non-zero encryption group that is specified if the encryption is not activated.

– The DS8000 monitors all configured IBM Security Key Lifecycle Manager key servers. When loss of access to the key servers is detected, notification is provided through the DS8000 client notification mechanism (SNMP traps, email, or both, when configured). Key server-related errors are provided through the same mechanism. Set up monitoring for these indications and take corrective actions when a condition is detected. This reflects a degraded key server environment.

  The following conditions are monitored and reported:

  • If the DS8000 cannot receive a required data key during power-on for a configured encryption group from the key servers, it reports the error condition to the client and to IBM. In this case, logical volumes that are associated with the encryption group are inaccessible to attached hosts. After reporting the error, if the DS8000 can get the required data key from a key server, it reports the condition to the client and to IBM and makes the associated logical volume accessible.

  • DS8000 access to each configured key server is verified at 5-minute intervals. Loss of access is reported to the client.

  • The ability of each key server to unwrap data keys that are configured on the DS8000 is verified at 8-hour intervals. Loss of the ability to unwrap a configured data key is reported to the client and to IBM.

  • The DS8000 detects if fewer than two key servers are configured, if fewer than two key servers are available, or if fewer than two key servers can unwrap data keys that are configured on the DS8000 at 8-hour intervals. If detected, this condition is reported to the client and to IBM.

3.7 Multiple IBM Security Key Lifecycle Managers for redundancy

To ensure continuous key and certificate availability to encrypting devices, configure a primary and a replica IBM Security Key Lifecycle Manager server for your enterprise, and then provide repeated backup and restore or import and export actions to protect critical data.

On Microsoft Windows systems and other systems, such as Linux or AIX, both computers must have the required memory, speed, and available disk space to handle the workload.

This is not a failover or clustered server from an IBM Security Key Lifecycle Manager point of view. The redundancy is managed by setting up multiple key manager destinations at the DS8000.

Synchronization is achieved by backing up one server and restoring the backup configuration on the other server, or by setting up a replication between two or more IBM Security Key Lifecycle Manager servers. The automatic replication process is run only when the new keys are added. Plan to do this backup or restore or export or import process when the following events take place:

- Initial configuration
- Adding keys or devices
- Key or certificate replacement intervals
- Certificate authority (CA) requests

The following IBM Security Key Lifecycle Manager data is replicated:

- Data in the IBM Security Key Lifecycle Manager database tables
- All keys materials in the IBM Security Key Lifecycle Manager database
- IBM Security Key Lifecycle Manager configuration files (except the replication configuration file)

4

IBM DS8000 encryption implementation

This chapter reviews the sequence of tasks that is involved in the deployment of an encryption-capable DS8000, from ordering to installation and use.

This chapter covers the following topics:

- Installing IBM Security Key Lifecycle Manager V2.6 in silent mode (quick installation guide)
- Installing the IBM Security Key Lifecycle Manager V2.6 fixpack in silent mode (quick installation guide)
- Configuring IBM Security Key Lifecycle Manager
- Configuring Gemalto SafeNet KeySecure with Key Management Interoperability Protocol (KMIP)
- GUI disk encryption configuration for the DS8000
- CLI disk encryption configuration for the DS8000
- Encryption and Copy Services considerations
- NIST requirements and migration of Gen1 to Gen2 certificates

4.1 Installing IBM Security Key Lifecycle Manager Version 2.6 in silent mode (quick installation guide)

The IBM Security Key Lifecycle Manager V2.6 installation, including all prerequisites, is covered in the IBM Security Key Lifecycle Manager IBM Knowledge Center:

http://ibm.biz/SKLMv26KC

The IBM Security Key Lifecycle Manager bundle includes the following software components:

- Runtime environment:
  - IBM WebSphere® Application Server Version 8.5.5.7
  - IBM WebSphere SDK Java Technology Edition Version 7.0.9.10
- Database: DB2 Workgroup Server Edition Version 10.5.0.6
- IBM Security Key Lifecycle Manager Version 2.6

In the IBM Security Key Lifecycle Manager base architecture, the WebSphere Application Server runs a Java virtual machine, providing the runtime environment. The application server provides communication security, logging, messaging, and web services. DB2 stores key materials and other essential IBM Security Key Lifecycle Manager information in a relational database. The IBM Security Key Lifecycle Manager base architecture is shown in Figure 4-1.

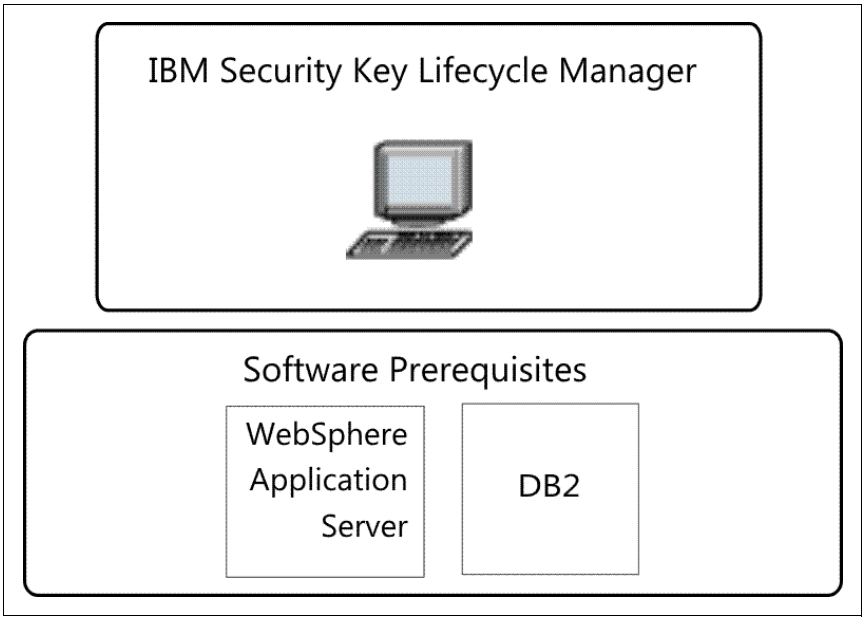


Figure 4-1 IBM Security Key Lifecycle Manager base architecture

Note: This paper describes an IBM Security Key Lifecycle Manager Version 2.6 installation on a Linux platform.

4.1.1 Before starting the installation

Before you install IBM Security Key Lifecycle Manager, complete the following steps:

1. Extract both installation images (disk1 and disk2), previously obtained from IBM Passport Advantage. Use the same subfolder to extract the images, as shown in Example 4-1. Keep the paths as short as possible.

Example 4-1 Extracted installation images

```
/sklm26/ # ls -l
drwxr-xr-x 11  300  300       4096 Apr 28 13:03 disk1
drwxr-xr-x  3  300  300       4096 Nov 14 09:21 disk2
```

2. IBM Security Key Lifecycle Manager includes sample response files that you can use as a template for creating your own response file. The sample file must be modified for the specifics of your environment before it can be used. Create a copy of the sample response file, as shown in Example 4-2.

Example 4-2 Create a response file copy

```
/disk1 # cp SKLM_Silent_Linux_Resp.xml myresp.xml
/disk1 # ls -l *xml
-rwxr-xr-x 1 root root 6180 Nov 14 09:11 SKLM_Silent_Linux_Mig_Resp.xml
-rwxr-xr-x 1 root root 5700 Nov 14 09:11 SKLM_Silent_Linux_Resp.xml
-rwxr-xr-x 1 root root  750 Nov 14 09:11 SKLM_Uninstall_Linux_Resp.xml
-rwxr-xr-x 1 root root 5700 May  6 10:51 myresp.xml
-rw-r--r-- 1 root root 3298 Apr 28 13:03 result.xml
```

4.1.2 Silent mode installation on Linux

A silent installation is a non-interactive installation, which is driven by a response file that provides installation settings.

No user input is required during a silent installation. This type of installation is useful in environments where IBM Security Key Lifecycle Manager is installed on multiple identical systems, such as in a data center or when working from a remote location.

Note: Silent mode installation uses a response file that might contain encrypted password information. For more security, delete the response file immediately after the installation of IBM Security Key Lifecycle Manager.

You must add the encrypted passwords to the relevant elements of the response file. Use the IBM Installation Manager utility to create an encrypted password. Complete the following steps:

1. Navigate to the IBM Security Key Lifecycle Manager directory to which the file was extracted. The password encryption tool is available in `/disk1/im/tools`. See Example 4-3.

Example 4-3 Create the encrypted password

```
/disk1/im/tools # ./imcl encryptString myPasswOrd
HUTvoorFkpZvxu48WXv8qQ==
```

2. Add the encrypted password that you created in the response file, as shown in Example 4-4.

Example 4-4 Password modifications in the response file

```
<data key='user.DB2_ADMIN_PWD,com.ibm.db2.ofng'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.CONFIRM_PASSWORD,com.ibm.db2.ofng'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
...
<data key='user.WAS_ADMIN_PASSWORD,com.ibm.sklm.win32'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.WAS_ADMIN_CONF_PWD,com.ibm.sklm.win32'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
...
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.sklm.win32'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.SKLM_ADMIN_CONF_PWD,com.ibm.sklm.win32'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
```

3. Modify the repository location if it does not match the defaults in the response file, as shown in Example 4-5.

Example 4-5 Repository

```
<repository location='/sklm/disk1/im'/>
<repository location='/sklm/disk1/'/>
```

Note: Do not specify the `disk2` directory as a repository.

4. Start the installation, as shown in Example 4-6.

Example 4-6 Silent installation

```
/disk1 # ./silent_install.sh /sklm26/disk1/myresp.xml
```

4.1.3 Installing Fix Pack 1 (or later) for IBM Security Key Lifecycle Manager V2.6

The fix packs for IBM Security Key Lifecycle Manager V2.6 include the latest fixes and security patches. All IBM Security Key Lifecycle Manager for Distributed Platforms fix packs are cumulative.

Fix packs are available at the IBM Support Portal:

https://www.ibm.com/support/entry/portal/support

Fix packs can be installed in silent mode as well.

Complete the following steps:

1. You must add the encrypted passwords to the relevant elements of the response file. Use the IBM Installation Manager utility to create an encrypted password.

   Navigate to the IBM Installation Manager directory. The password encryption tool is available in `/opt/IBM/InstallationManager/eclipse/tools`. See Example 4-7.

   Example 4-7 Create the encrypted password

```
/opt/IBM/InstallationManager/eclipse/tools # ./imcl encryptString myPasswOrd
HUTvoorFkpZvxu48WXv8qQ==
```

2. IBM Security Key Lifecycle Manager Fix Pack 1 includes a sample response file. The sample file must be modified for the specifics of your environment before it can be used. Do not create your own response file. Add the encrypted password that you created to the given, sample response file shown in Example 4-8.

   Example 4-8 Password modifications in the sample response file

```
<data key='user.WAS_ADMIN_ID,com.ibm.sklm26.linux' value='wasadmin'/>
<data key='user.WAS_ADMIN_PASSWORD,com.ibm.sklm26.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.SKLM_ADMIN_USER,com.ibm.sklm26.linux' value='SKLMAdmin'/>
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.sklm26.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.DB_ADMIN_USER,com.ibm.sklm26.linux' value='sklmdb26'/>
<data key='user.DB_ADMIN_PASSWORD,com.ibm.sklm26.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
```

3. Start the update after making the installer executable file, as shown in Example 4-9. You do not have to specify the response file.

   Example 4-9 Install the fix pack

```
/sklm26/fp1 # chmod +x ./silent_updateSKLM.sh
/sklm26/fp1 # ./silent_updateSKLM.sh /opt/IBM/InstallationManager
/opt/IBM/WebSphere/AppServer wasadmin myPasswOrd
```

4.1.4 Issues with IBM Security Key Lifecycle Manager DB2/WebSphere Application Server starting correctly after a restart on Linux

After restarting IBM Security Key Lifecycle Manager on Linux operating systems, such as SLES or RHEL, IBM WebSphere Application Server starts before DB2 does, thus causing the IBM Security Key Lifecycle Manager application to be unable to connect to DB2 and WebSphere Application Server fails to supply keys to the DS8000.

When recovering from a power loss, this situation can cause serious issues and delays in starting production. Human intervention is required to make IBM Security Key Lifecycle Manager function.

A workaround for Linux operation systems is documented and available in Technote 1969891. See the following website and apply the workaround after you finish the IBM Security Key Lifecycle Manager installation. There is no permanent fix available at the time of writing.

`http://www.ibm.com/support/docview.wss?uid=swg21969891`

4.2 IBM Security Key Lifecycle Manager Version 2.6 configuration

IBM Security Key Lifecycle Manager Version 2.6 for open systems and Microsoft Windows has all of the features and functions that were previously supported in IBM Tivoli Key Lifecycle Manager Version 2.0.1.0 for open systems (*TKLM* in path and file names). IBM Security Key Lifecycle Manager Version 2.6 for open systems replaces the Tivoli Key Lifecycle Manager. IBM Security Key Lifecycle Manager for open systems also supports Microsoft Windows, Linux, and IBM AIX operating systems. (IBM Security Key Lifecycle Manager for z/OS is a different product that is not mentioned again in this chapter.)

IBM Security Key Lifecycle Manager supports the *Gen-1* certificates with 80-bit security strength, which the DS8000 has used since encryption was introduced. IBM Security Key Lifecycle Manager also supports the *Gen-2* certificates with 112-bit encryption strength, which is used by DS8000 disk storage systems running Release 7.2 (LMC level 7.7.20.xx). It is mandatory for DS8000 disk storage systems that are shipped with Release 8.1 by manufacturing.

TKLM used a local keystore that contained all certificates and keys. IBM Security Key Lifecycle Manager uses DB2 to store all certificates and keys. Operationally, these two different key and certificate repositories are equivalent.

IBM Security Key Lifecycle Manager security features follow NIST SP 800-131a requirements and maintain compatibility with previous security and encryption certificates that were used for previous generations of the DS8000 series.

Note: Migrating to IBM Security Key Lifecycle Manager and using the Gen-2 certificate is not reversible, so plan carefully.

Here are some of the enhancements of IBM Security Key Lifecycle Manager compared its predecessor TKLM:

- Support for only 64-bit platforms.
- Transport Layer Security (TLS) 1.2 and Elliptic Curve Digital Signature Algorithm (ECDSA) keys and certificates.

- Keys and certificates are stored in a database (the name keystore is still used).
- Simplified installation by using IBM Installation Manager.

This section describes the procedure to configure IBM Security Key Lifecycle Manager to serve an encryption-enabled DS8000. The instructions are based on the assumption that the IBM Security Key Lifecycle Manager servers are installed and ready for configuration. The system clocks of all key server must be relatively synchronized. Detailed information is available from the IBM Security Key Lifecycle Manager V2.6 IBM Knowledge Center:

`http://ibm.biz/SKLMv26KC`

Configuring IBM Security Key Lifecycle Manager requires several steps to prepare the key server to serve a DS8000 encryption-enabled disk storage system. The following benefits are new to this release:

- Encryption strength of the Rivest-Shamir-Adleman (RSA) algorithm with 2048-bit keys. There is still support for Advanced Encryption Standard (AES) 256-bit keys, which were supported by the previous generation of key server, Tivoli Key Lifecycle Manager V2.0.1.0.
- Support for SSL by using TLS Version 1.2 to encrypt communication between the DS8000 Hardware Management Console (HMC) and IBM Security Key Lifecycle Manager.

Attention: For more information about using TLS 1.2 with IBM Security Key Lifecycle Manager to make it NIST SP 800-131a compliant, see 4.7, “NIST SP 800-131a requirements for key servers” on page 136 and modify the IBM Security Key Lifecycle Manager configuration.

4.2.1 Logging in to the IBM Security Key Lifecycle Manager console

The IBM Security Key Lifecycle Manager installation secures HTTPS transport with a self-signed certificate by default. Depending on the browser and version that is used, an exception might occur. In that case, you must accept the certificate as a trusted certificate by completing the following steps:

1. Log in to IBM Security Key Lifecycle Manager (see Figure 4-2) at the following address:

   `https://<ip address>:9080/ibm/SKLM/login.jsp`

Figure 4-2 IBM Security Key Lifecycle Manager login window

2. Select **Action Items** to begin the configuration of IBM Security Key Lifecycle Manager. The Action Items menu guides you through the configuration steps.
3. In the Welcome window, under Advanced Configuration, click **Server certificates** and then **Add**, as shown in Figure 4-3.

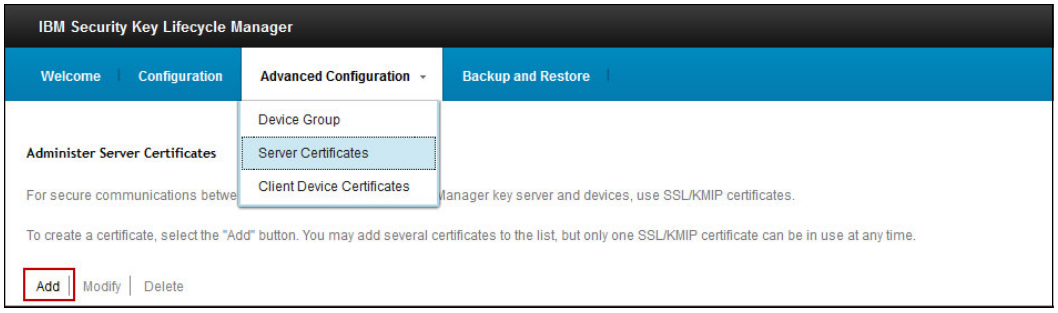


Figure 4-3 IBM Security Key Lifecycle Manager Welcome window

4.2.2 Creating the SSL certificate

The first time that you access IBM Security Key Lifecycle Manager, a link is available to create an SSL certificate. This example uses the self-signed certificate. Requesting a third-party certificate by using an existing certificate or by using no server certificate are also supported. Although using an existing certificate from the keystore is possible, using the certificate that is used to encrypt disk data to also protect communication is not a preferred practice.

Complete the following steps:

1. Provide a certificate label and a certificate description when creating the certificate.

   Figure 4-4 shows the SSL/KMIP for Key Serving window under the Advanced Configuration tab, where you create the certificate. This example is left blank to indicate that this field must be filled in when creating the certificate. The validity period determines how long the certificate is valid. The RSA algorithm uses the 2048-bit key.

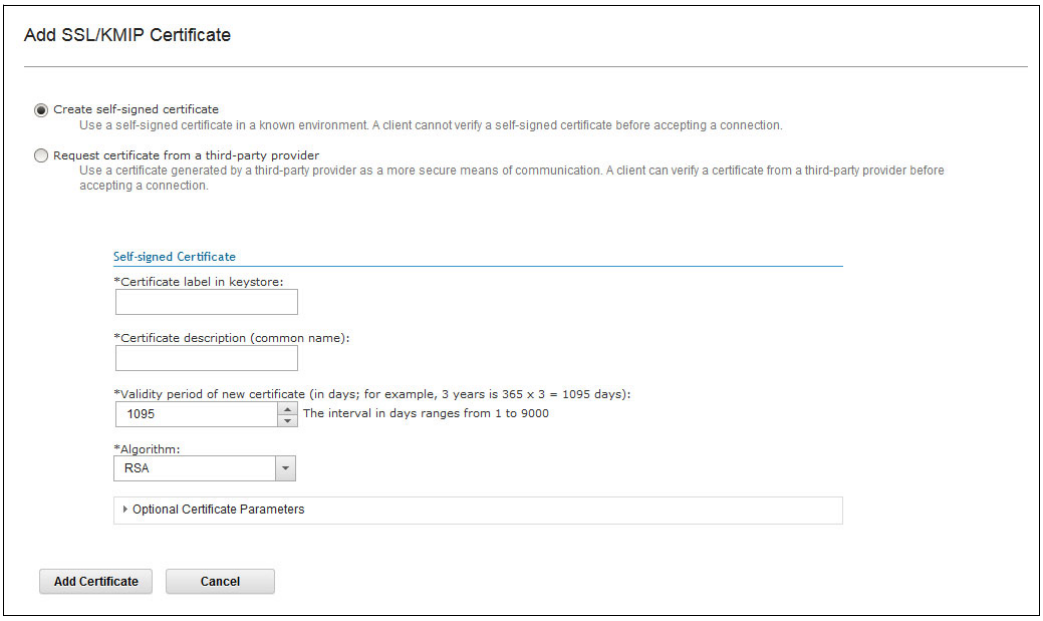


Figure 4-4 SSL/KMIP for Key Serving window

2. After you fill in all fields, click **Add Certificate** to create and add the certificate.
3. Figure 4-5 shows that the SSL certificate is created.

   Log off and stop the IBM Security Key Lifecycle Manager server, and then restart the IBM Security Key Lifecycle Manager server. For more information, see the IBM Security Key Lifecycle Manager IBM Information Center.

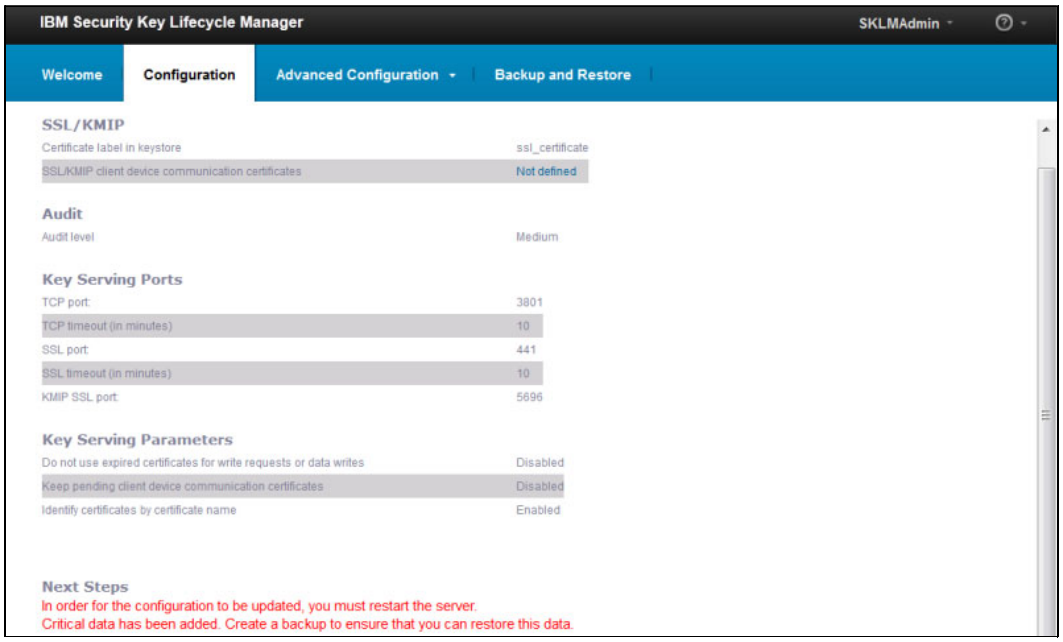


Figure 4-5 SSL certificate created successfully

4. After the IBM Security Key Lifecycle Manager server restarts, log in and create a backup. This backup contains only the certificate that you created. Navigate to the Backup and Restore window that is shown in Figure 4-6.

4.2.3 Creating a backup

To create a backup, complete the following steps:

1. Under the Backup and Restore tab that is shown in Figure 4-6, next to the Backup repository location field, click **Browse** to select a path for the backup.

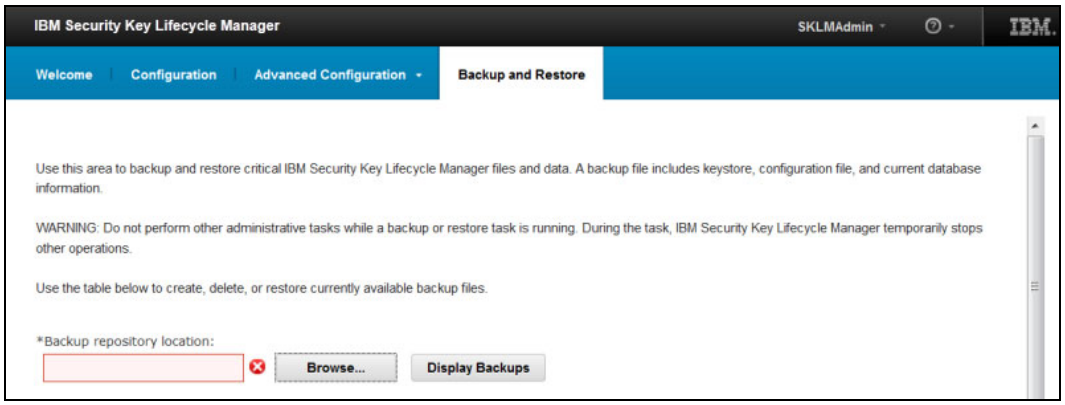


Figure 4-6 Backup and Restore tab

2. After highlighting the directory where the backup will be stored, click **Select**. Figure 4-7 shows `/home` as the directory.

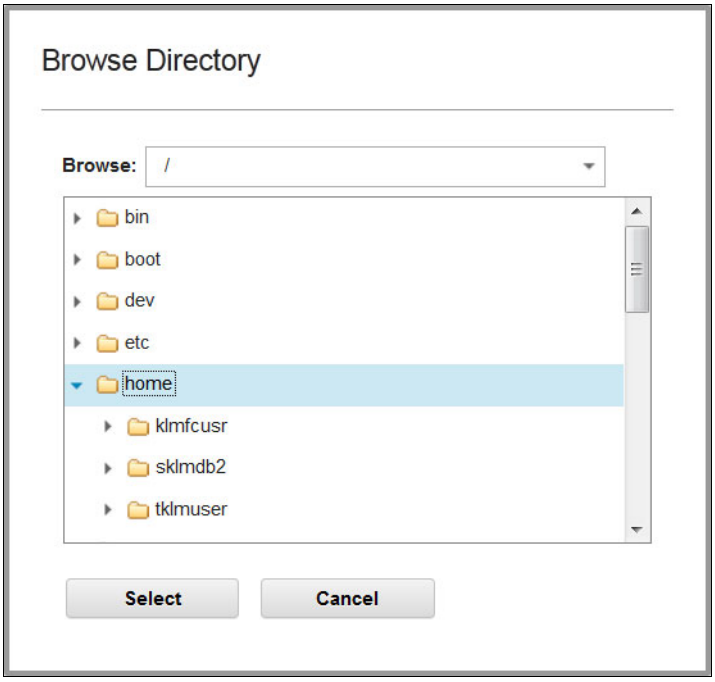


Figure 4-7 Browse the directory for the repository backup location

3. After selecting a directory, a confirmation window opens. Click **OK** to confirm that you want to continue.
4. In the Create Backup window that is shown in Figure 4-8, enter a password for the backup, and then click **Create Backup**. This password is *required* to use the restore function in the future.

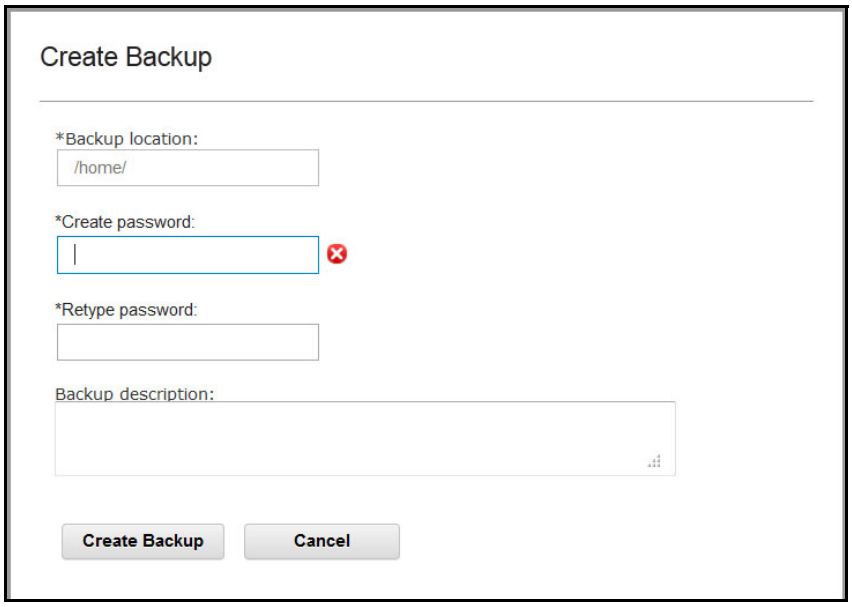


Figure 4-8 Create Backup window to enter the password and backup description

5. When you see the “successfully created” notice that is shown in Figure 4-9, click **Close**.

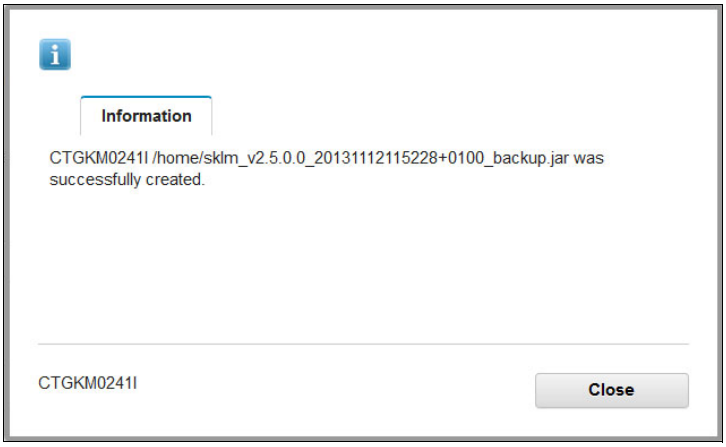


Figure 4-9 Backup was successfully completed

As this is not a failover or clustered server from an IBM Security Key Lifecycle Manager point of view, the redundancy is managed by setting up multiple key manager destinations at the DS8000. Synchronization is achieved by backing up one server and restoring the backup configuration on the other server (see step 1 on page 66) or by setting up remote replication between the IBM Security Key Lifecycle Manager Key servers (see “4.2.5, “Setting up remote replication between IBM Security Key Lifecycle Manager key servers” on page 68”).

Plan to do this backup or restore process when the following events take place:

- Initial configuration
- Adding keys or devices
- Key or certificate replacement intervals
- Certificate authority (CA) requests

Transfer and restore the previously taken backup to all further IBM Security Key Lifecycle Manager Key servers that are installed.

4.2.4 Restoring the backup

To restore the backup, complete the following steps:

1. Log in to the IBM Security Key Lifecycle Manager and navigate to **Backup and Restore** and then click **Browse** to browse for the previously transferred backup file, as shown in Figure 4-10.

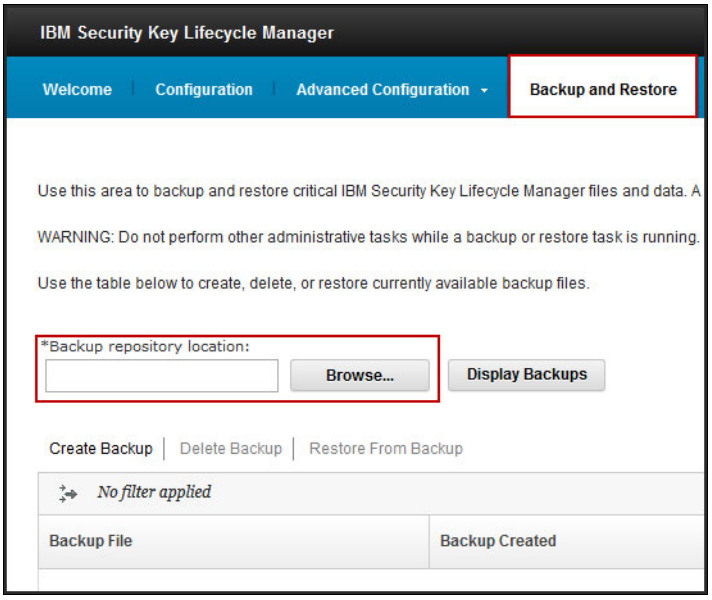


Figure 4-10 Browse for backup

2. Browse to the directory with the stored Backup, as shown in Figure 4-11 and click **Select**.

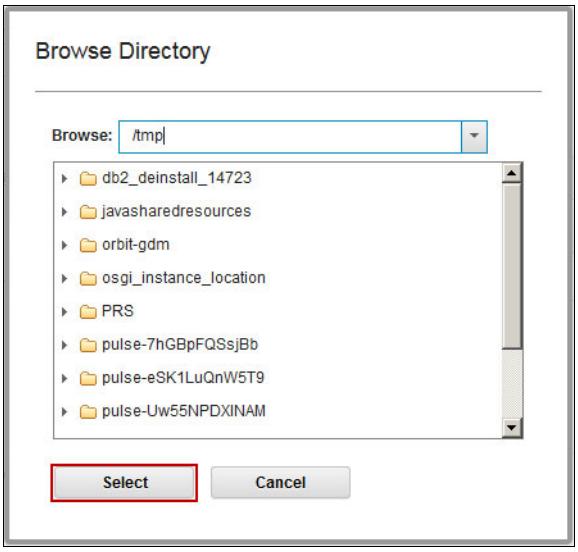


Figure 4-11 Browse to the directory

3. Click **Display Backups**, and window refreshes. The Backup appears as shown in Figure 4-12.

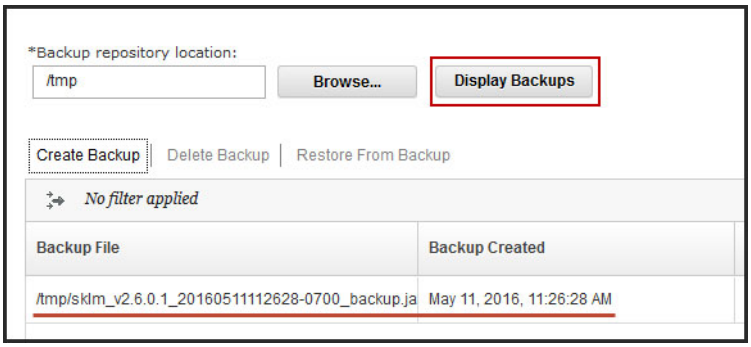


Figure 4-12 Display Backups

4. Select the backup and click **Restore From Backup**, as shown in Figure 4-13.

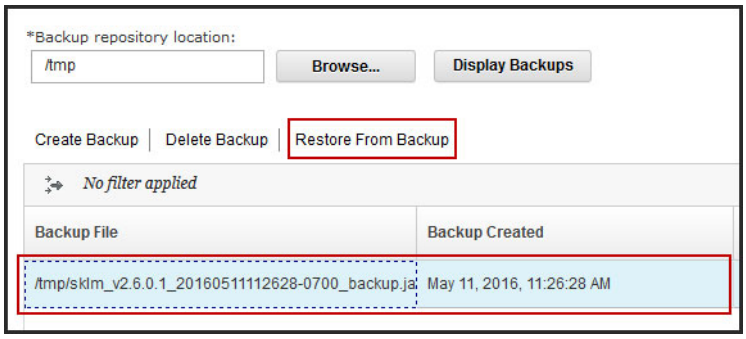


Figure 4-13 Highlight and restore

5. You are asked for the password to encrypt. Type it in and click **Restore Backup**, as shown in Figure 4-14.

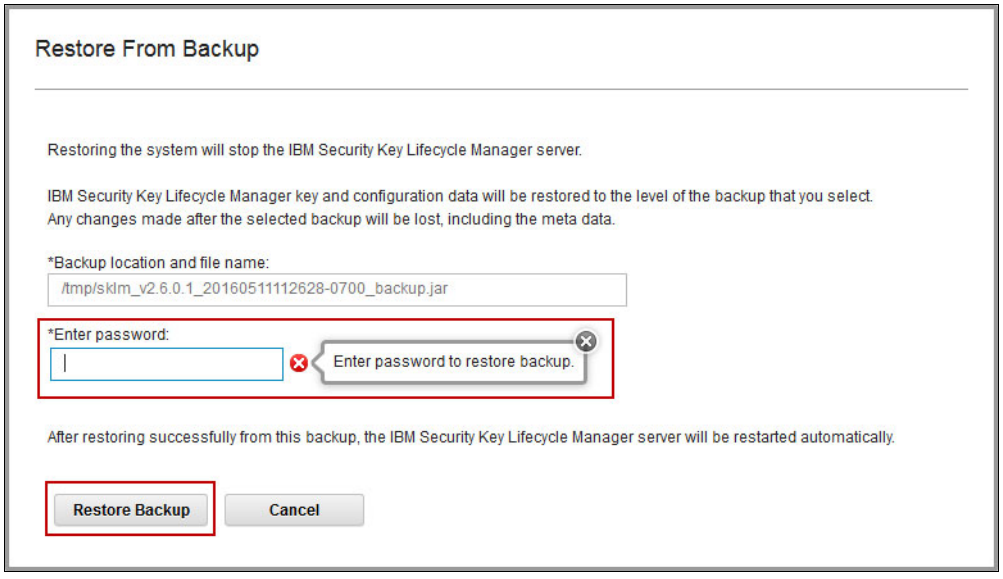


Figure 4-14 Restore backup

6. The following dialog must be confirmed by clicking:

   “The IBM Security Key Lifecycle Manager key and configuration data will be restored to the level of the backup that you select. Any changes that are made after the selected backup will be lost, including the metadata. After restoring successfully from this backup, the IBM Security Key Lifecycle Manager server will be restarted automatically as the autoRestartAfterRestore variable for auto restart after restore is set to True (default is True). The server will not be available during the server restart. After the server is restarted, restart your browser session (log in again to use the product UI).”

7. The backup is successfully restored if you receive the message that is shown in Figure 4-15.

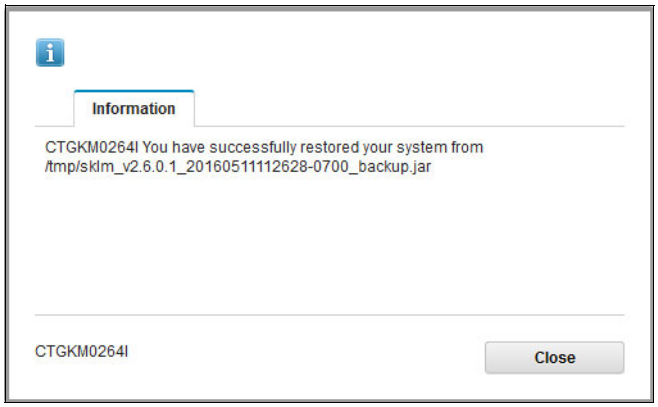


Figure 4-15 Successful

4.2.5 Setting up remote replication between IBM Security Key Lifecycle Manager key servers

To set up the IBM Security Key Lifecycle Manager automated clone replication process, you must configure the replication parameters for the *master* and *clone* servers.

IBM Security Key Lifecycle Manager provides a set of operations to replicate current active files and data across systems. This replication enables cloning of IBM Security Key Lifecycle Manager environments to multiple servers in a manner that is independent of operating systems and directory structure of the server. For example, you can replicate data from a master server on a Windows system to a clone server on a Linux system.

Master server configuration

Master server is the primary system that is being replicated. The replication process is triggered only when the new keys are added to the master server. You can replicate the master server with a maximum of 20 clone servers. Each clone server is identified through an IP address or host name, and a port number. The server uses properties in the `ReplicationSKLMConfig.properties` file to control the replication process.

You can also use the IBM Security Key Lifecycle Manager replication program to schedule automatic backup operation. You must configure properties only for the master server to back up data at regular intervals. For more information about automatic backup operations, see the IBM Security Key Lifecycle Manager IBM Knowledge Center:

`http://ibm.biz/SKLMv26KC`

Clone server configuration

The replication process enables cloning of IBM Security Key Lifecycle Manager environments from master server to multiple clone servers. The clone server uses properties in the `ReplicationSKLMConfig.properties` file to control the replication process. When the replication process is triggered, the following data is replicated to the clone server:

- Data in the IBM Security Key Lifecycle Manager database tables
- Truststore and keystore with the master key
- IBM Security Key Lifecycle Manager configuration files

Specifying replication parameters for a master server

Complete the following steps:

1. Log in to the IBM Security Key Lifecycle Manager that is going to become your master key server and click **Configuration** → **Replication**, as shown in Figure 4-16.

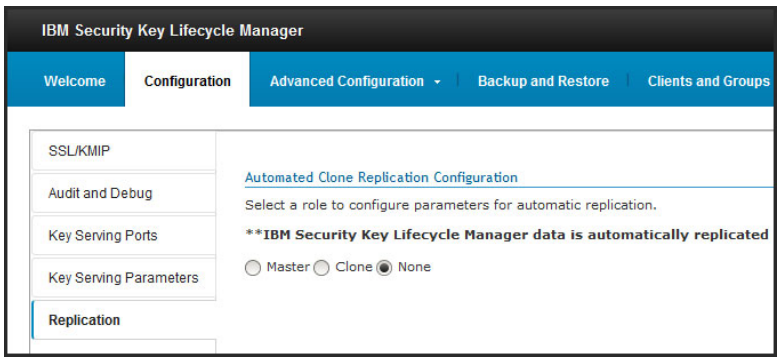


Figure 4-16 Start replication configuration on the master key server

2. Change the value for one or more settings of the master server. Select **Master** and confirm “Are you sure to set up this SKLM as Master?” by clicking **OK**, as shown in Figure 4-17.

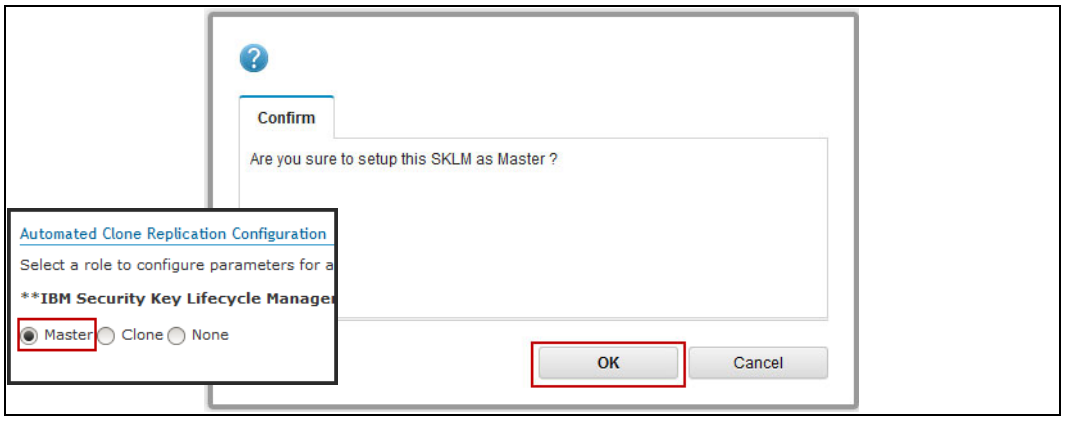


Figure 4-17 Confirm the master

3. Specify the following appropriate Basic Properties settings for the master, and then click **OK**, as shown in Figure 4-18:
   - Select a certificate from the list. The SSL/TLS certificate must exist on the master and all clone systems that you configure for replication.
   - The encryption password for the backup file to ensure data security. You need the same password to decrypt and restore the file.
   - The port number for communication when non-serialized or delayed replications take place. The default master listen port is 1111.

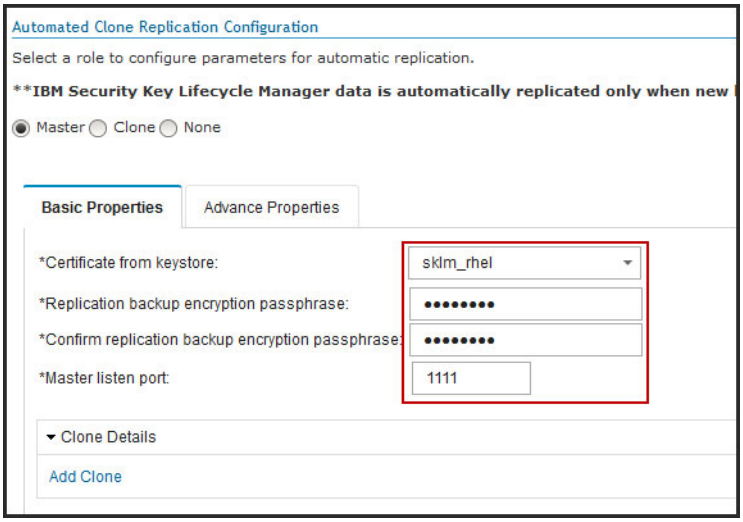


Figure 4-18 Basic Properties (Master)

Note: There is no need to touch the default parameters under Advanced Properties.

Specifying replication parameters for a clone server

Complete the following steps:

1. Log in to the IBM Security Key Lifecycle Manager that is going to become your master key server and click **Configuration** → **Replication**, as shown in Figure 4-19.

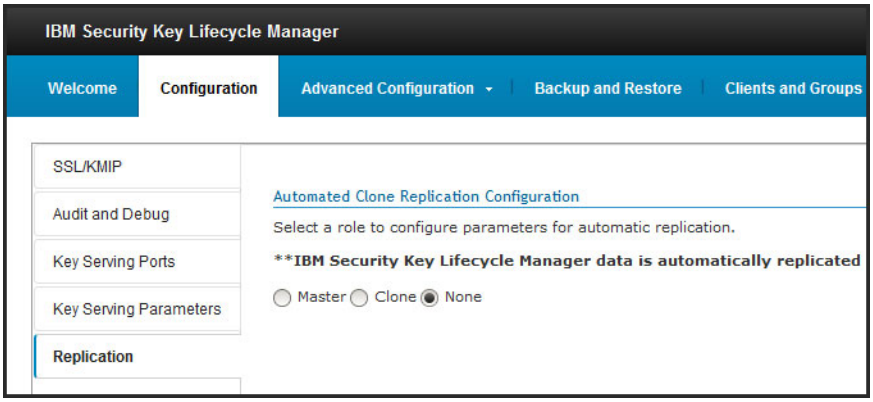


Figure 4-19 Start Replication Configuration on the clone key server

2. Change the value for one or more settings of the clone server. Select **Clone** and change the values under **Basic Properties**, if required. Click **OK**, as shown in Figure 4-20.

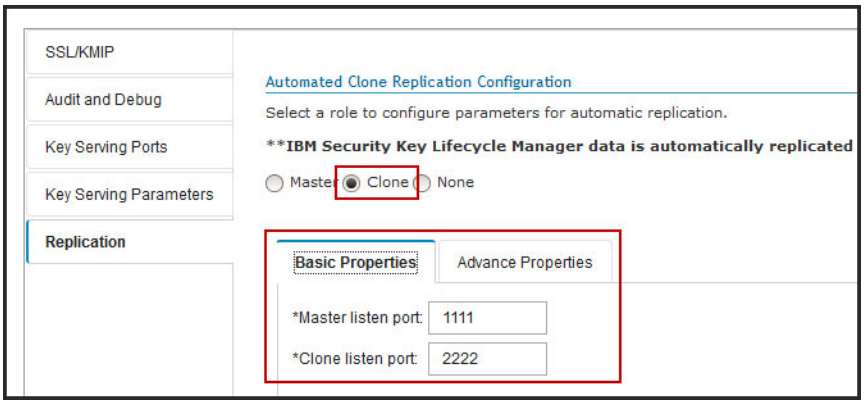


Figure 4-20 Basic Properties (Clone)

Note: There is no need to touch the default parameters under Advanced Properties.

Configuring the clone in the master key server

To use replication, you must have one master and at least one clone IBM Security Key Lifecycle Manager server available. The clone must be known to the master. Complete the following steps:

1. Log in to the master IBM Security Key Lifecycle Manager and click **Configuration** → **Replication** → **Add Clone**, as shown in Figure 4-21.

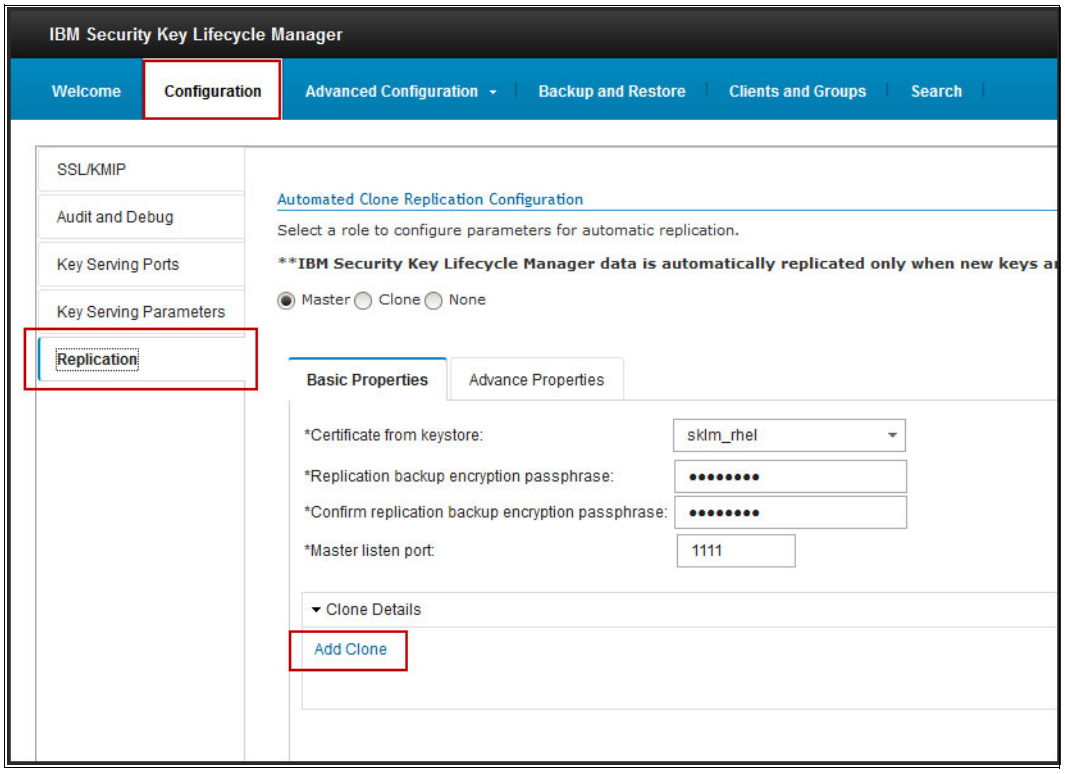


Figure 4-21 Add Clone

2. Add the full qualified host name or IP address of the clone and click **OK**, as shown in Figure 4-22.

Figure 4-22 Insert IP address / host name

3. Restart the master IBM Security Key Lifecycle Manager server and then all clone servers. Then, verify whether the replications service is running. Log in to all IBM Security Key Lifecycle Manager servers and look for the replication status in the lower right corner, as shown in Figure 4-23.

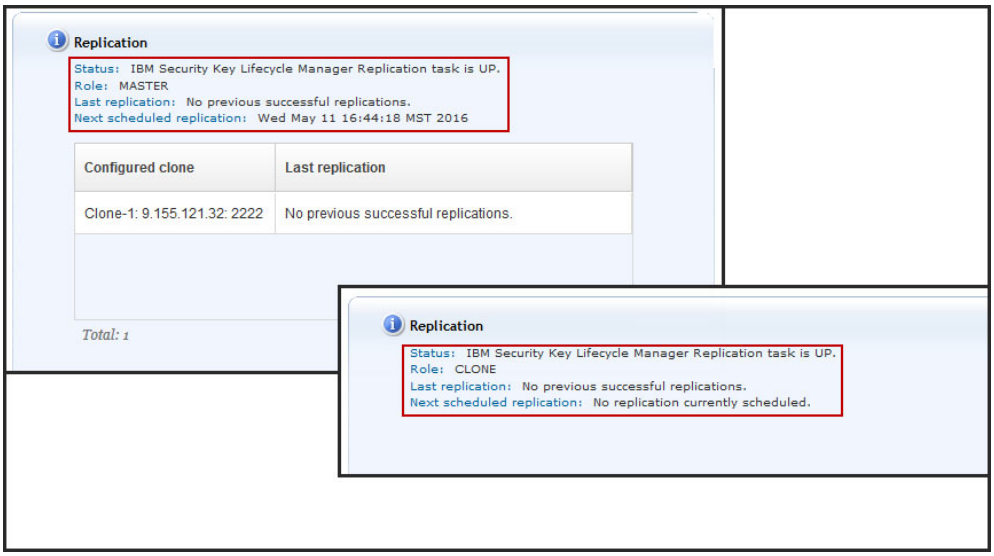


Figure 4-23 Verify the replication status

4.2.6 Defining the DS8000 storage facility image to use with IBM Security Key Lifecycle Manager

Create the image certificates that you need to associate the DS8000 storage facility images with IBM Security Key Lifecycle Manager by completing the following steps:

1. From the Welcome window, in the Key and Device Management pane, select **DS8000**. Then, from the **Go to** drop-down menu, select **Guided key and device creation**.

   These actions provide the guidance for steps that are required to add a storage facility image that the IBM Security Key Lifecycle Manager serves. Figure 4-24 shows these selections.

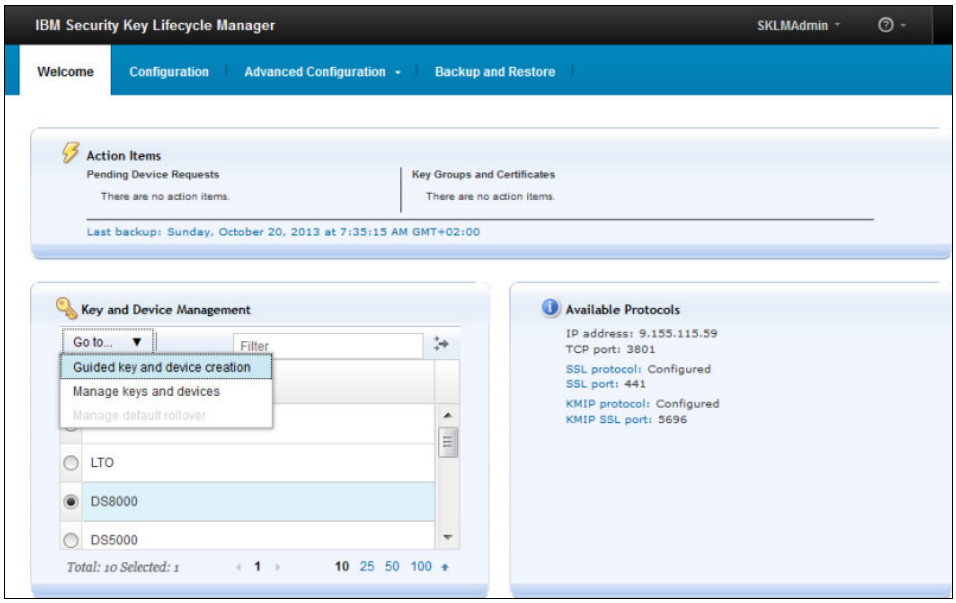


Figure 4-24 Guided key and device creation that is selected with DS8000

2. Click the **Advanced Configuration** tab that is shown in Figure 4-25 to create the certificate that will be associated with the storage facility image.

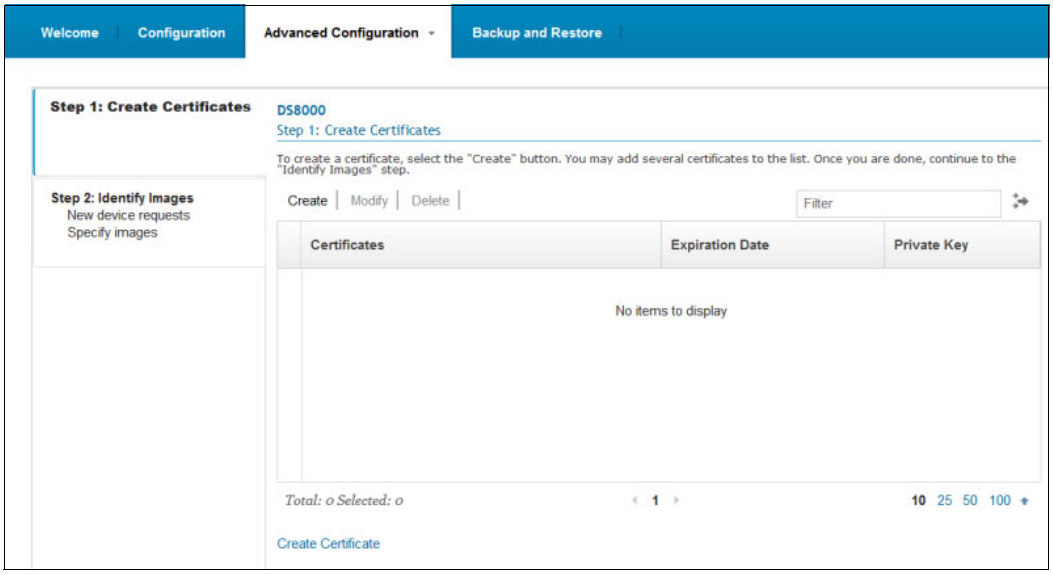


Figure 4-25 Step 1: Create Certificates window

3. After filling in the required fields (see Figure 4-26), click **Create Certificate**. Do not confuse this certificate with the one that is created in 4.2.2, “Creating the SSL certificate” on page 62 for network communication. This certificate is used with the storage image. The maximum validity period of a certificate is 9000 days (more than 24 years).

   Both self-signed certificates and third-party certificates are supported.

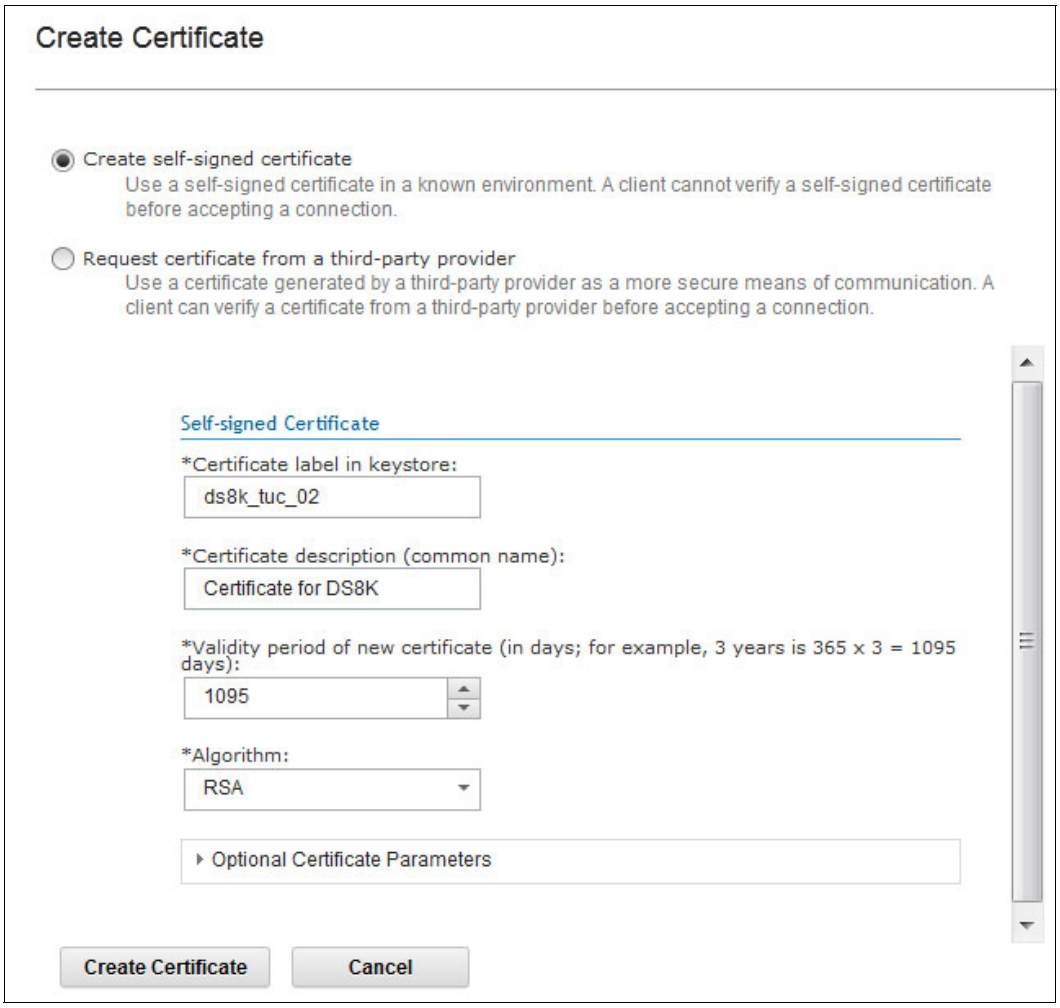


Figure 4-26 Create Certificate window to use for the storage facility image

After the certificate is created, a warning to create a backup displays (see Figure 4-27). Creating a backup includes the two certificates that were created: one for network communication and one that is going to be associated with the storage image. You can wait until after all storage images are defined to create the backup. It takes about 2 minutes to create the backup, and no progress indicator displays.

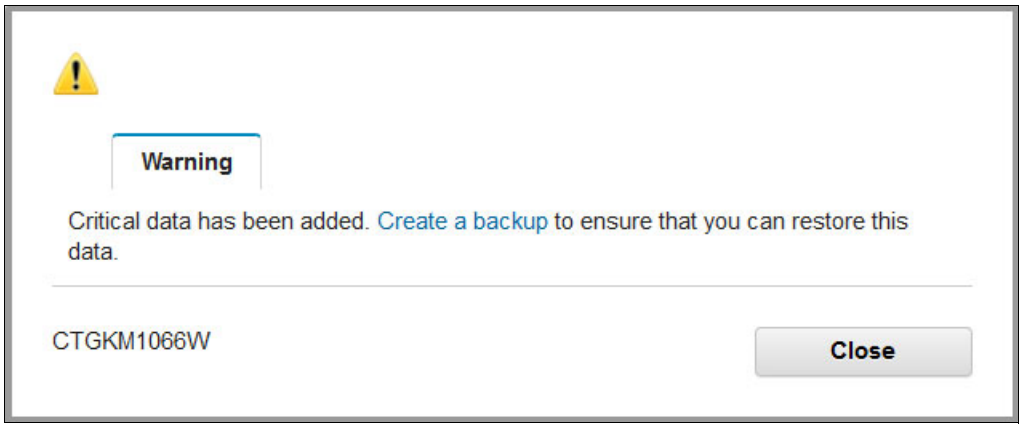


Figure 4-27 Warning to ensure the creation of a backup

4. If you want to create the backup now, click **Create a Backup**. If the backup will be created after the storage images are defined, click **Close**.

   The new certificate is available to associate with the storage image that you define next.

5. Click **Go to Next Step** at the bottom of the window, as shown in Figure 4-28.

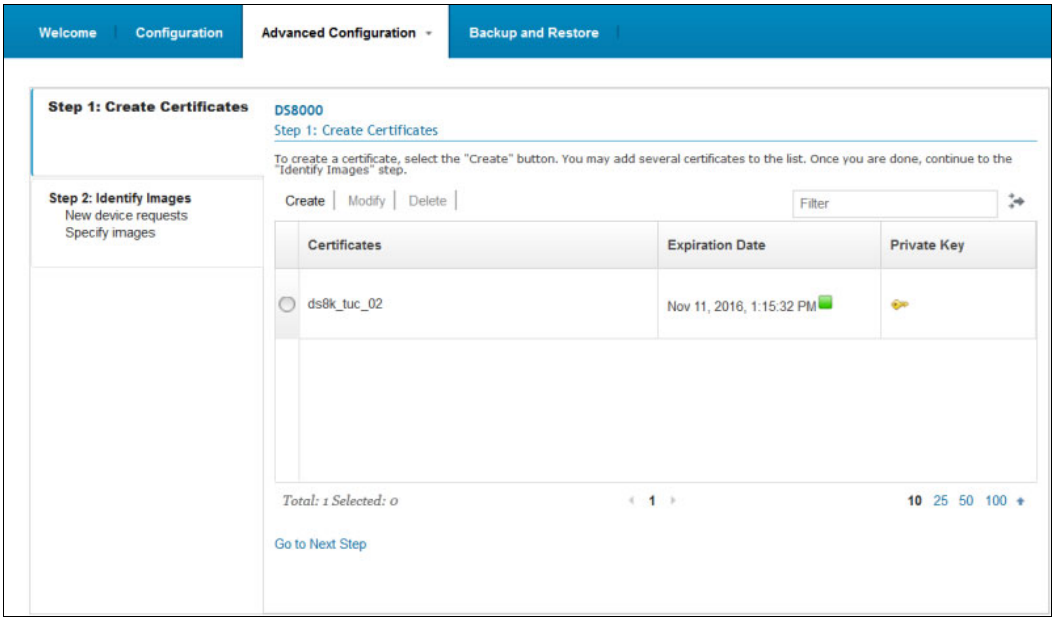


Figure 4-28 Create Certificates window under Advanced Configuration

6. When the window for Step 2: Identify Images opens (see Figure 4-29), add the storage image that is to be managed by the key server. This might seem confusing because it indicates that you are adding drives rather than defining storage images.

   If all the key servers are on the same platform, for example, Linux, only the primary certificate is used. If the key servers use multiple platforms, the secondary certificate also is created on the alternative platform.

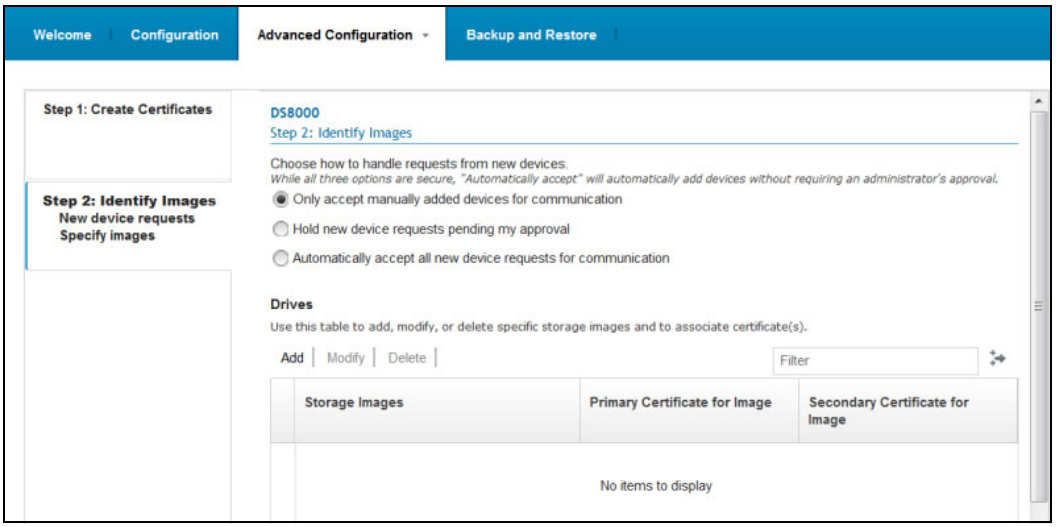


Figure 4-29 Step 2: Identify Images window

7. Under the Drives section, click **Add** to define the storage facility image. Enter the storage image and enter the certificate label that you created. The example in Figure 4-30 shows the correct format of the information that is required. All DS8000 models use the 2107-xxxxxxx format for the serial number.

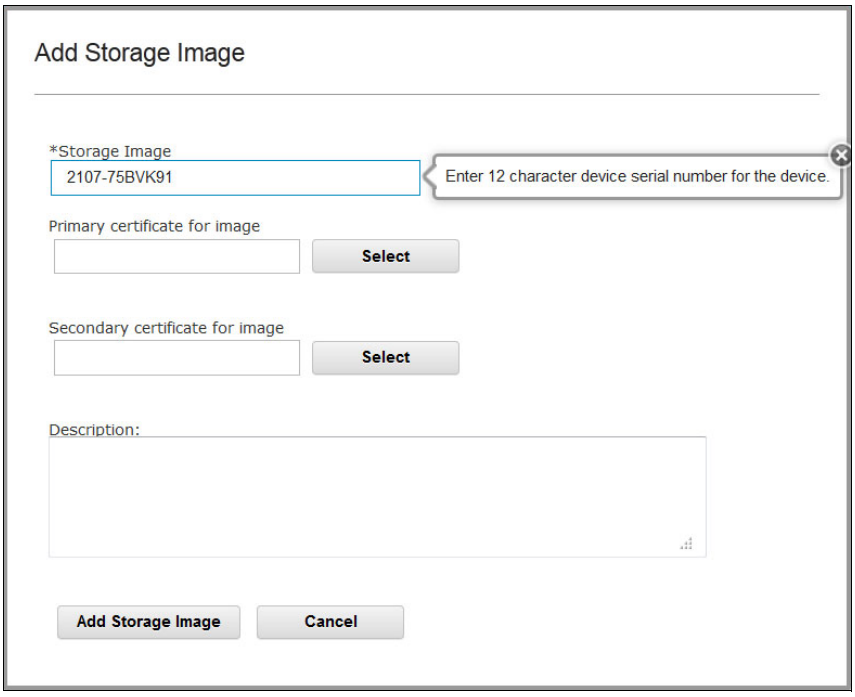


Figure 4-30 Add Storage Image window

In that window, if you click **Select** after the “Primary certificate for image” field, you get the window that is shown in Figure 4-31, where all certificate names are listed.

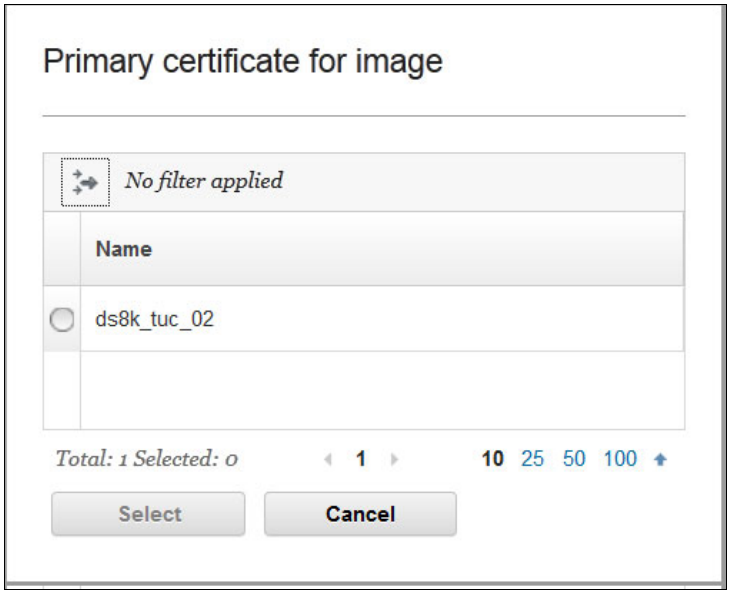


Figure 4-31 Certificate selection window

8. Select the radio button to select the certificate (**ds8k_tuc_02** for this example).

   After selecting the primary certificate for image, you return to the Add Storage Image window that is shown in Figure 4-32. Click **Add Storage Image** to complete the task.

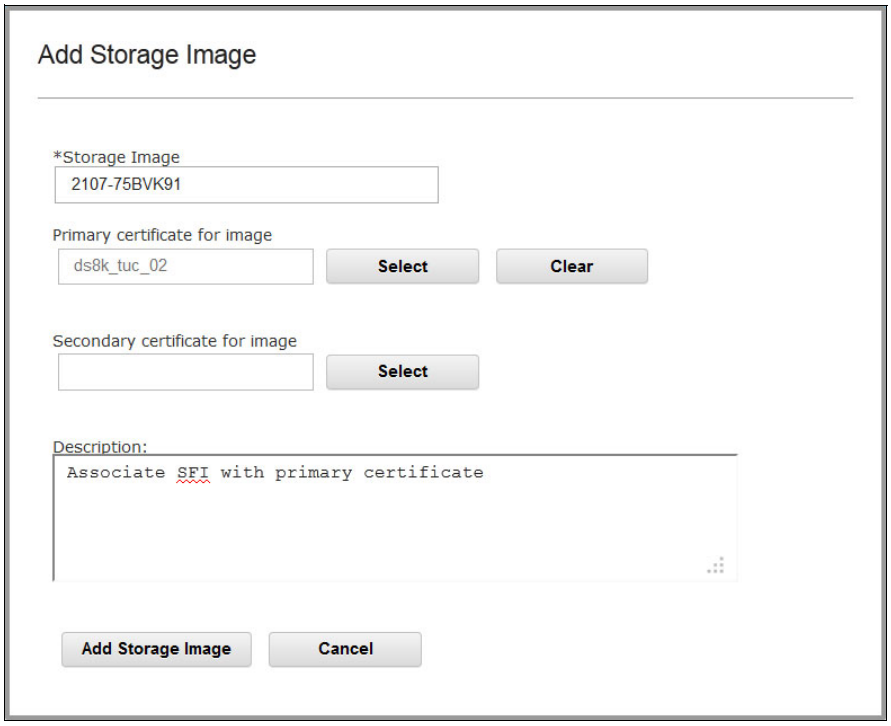


Figure 4-32 Add Storage Image window

You now have completed the task to add a storage image, as shown in Figure 4-33.

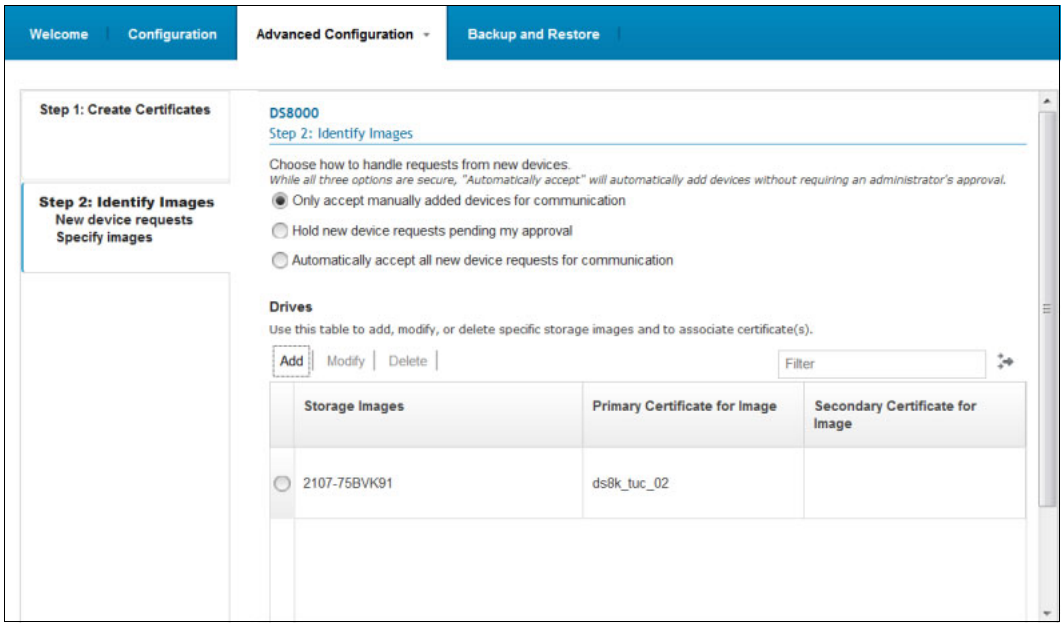


Figure 4-33 Identify Images window: Shows a machine that is already defined

Verifying that all information is correct

Complete the following steps:

1. Verify that the storage image and certificate information is correct. From the Welcome window, select **DS8000** and, under Key and Device Management, select **Manage keys and devices**, as shown in Figure 4-34. It is a good idea to verify all changes when adding storage images.

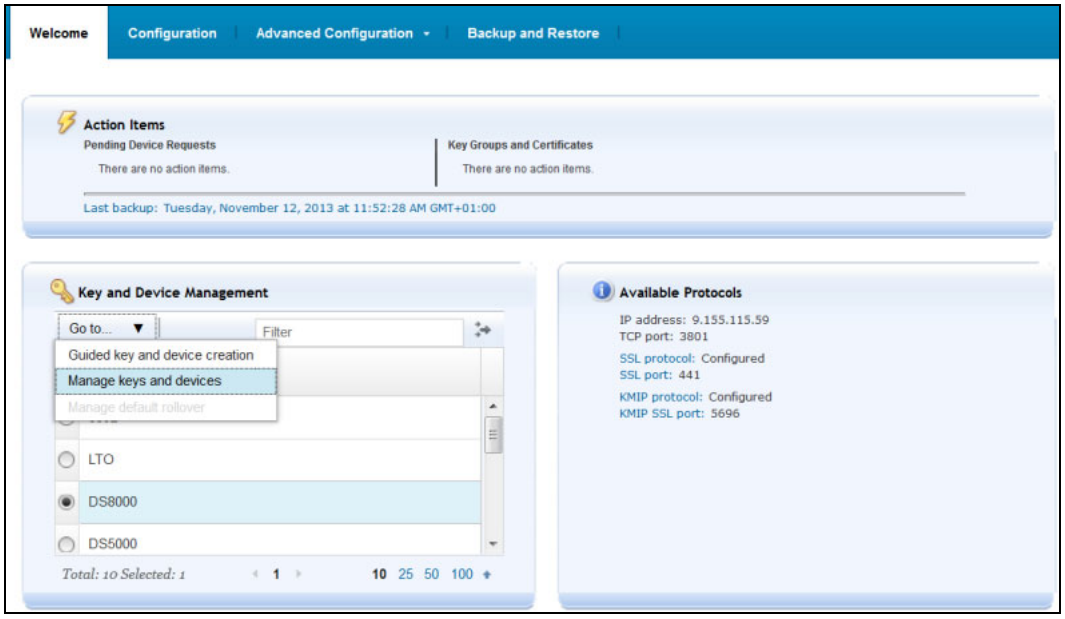


Figure 4-34 Manage keys and devices and DS8000 selected

2. Select the **Advanced Configuration** tab. Check all keys and storage images to verify that the information is correct, as shown in Figure 4-35.

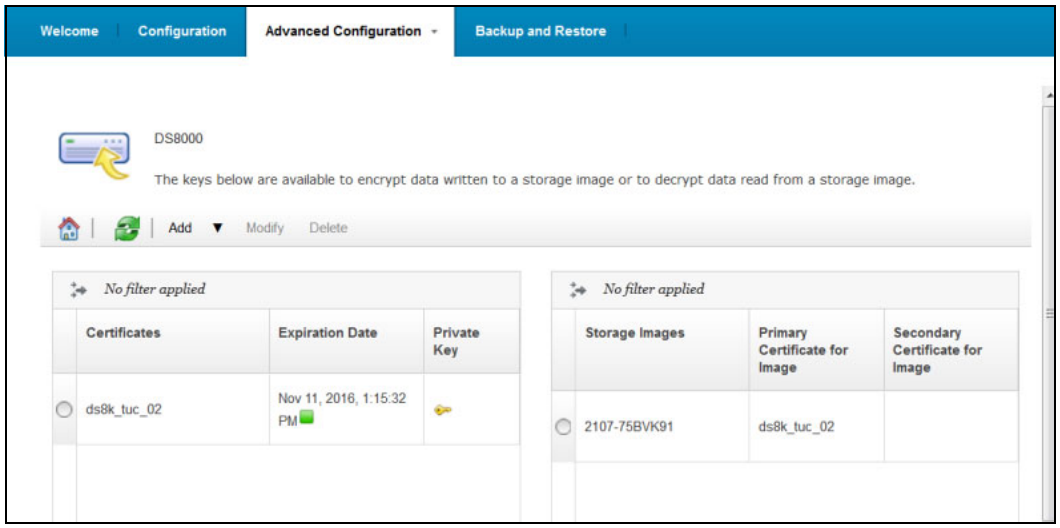


Figure 4-35 Advanced Configuration tab: Check keys and storage images

3. If no backup was created, create one now. If remote replication is not enabled, restore the backup now on all IBM Security Key Lifecycle Manager key servers.

Note: The IBM Security Key Lifecycle Manager key server is now ready to serve keys to the DS8000. The next task is to create the logical configuration. This task is not different when using encryption except for the Create Ranks and Create Pools tasks, which require **Encryption Group 1** to be selected.

4.3 Configuring Gemalto SafeNet KeySecure with KMIP

Gemalto SafeNet Key Secure is a third-party centralized key management platform, such as the IBM Security Key Lifecycle Manager, that is fully supported by DS8000 Release 8.1 and later.

Gemalto SafeNet provides KeySecure as hardware and virtual software appliance.

At the time of writing, the current version is 8.3.2 RevA. It supports KMIP Version 1.1 (used with DS8000 Release 8.1). LDAP and Active Directory authentication, and multiple network management protocols.

Like IBM Security Key Lifecycle Manager, Gemalto SafeNet KeySecure provides a GUI, which is named *Gemalto SafeNet KeySecure Management Console*. It supports 128-bit encryption and an SSH CLI.

Gemalto SafeNet KeySecure can manage up to 1,000,000 keys and 1,000 devices. It supports the hardware security module (HSM) to store the master key.

For more information about Gemalto SafeNet KeySecure, see the following website:

`http://www.safenet-inc.com/data-encryption/enterprise-key-management/key-secure/`

KMIP is an industry standard that strives to be a common language for key management systems and encryption systems of all varieties. There are many commercial key server vendors that support KMIP, and encryption systems as widely ranging as email databases and storage subsystems can communicate by using it.

The DS8000 supports KMIP to provide customers more flexibility and choice in key management. DS8000 customers now may take advantage of encryption if KMIP is a requirement in their infrastructure.

This section describes the procedure to configure Gemalto SafeNet KeySecure to serve keys to an encryption-enabled DS8000. The instructions are based on the assumption that the Gemalto SafeNet KeySecure servers are installed, clustered, and ready for configuration. The system clocks of all key server must be relatively synchronized.

Steps by step installation instructions are documented in the *KeySecure User Guide*, found at:

https://ibm.biz/Bd4JEm

4.3.1 Preparation

When using Gemalto SafeNet KeySecure KMIP Compatible Key Servers, KMIP must be configured with the necessary client certificate authentication policy. Three policies are supported by the DS8880:

- Client Certificate Authentication Not Used

  Not using Client Certificate Authentication when connecting to the DS8000 is not a preferred practice because this does not meet KMIP standards.

- Client Certificate Authentication used for SSL session only

  This policy applies to DS8000 disk storage systems that are shipped with Release 8.1 and to DS8000 disk storage systems that are upgraded from Release 8.0 to Release 8.1 and later.

- Client Certificate Authentication for SSL Session and User ID

  This policy applies to DS8000 disk storage systems that are shipped from manufacturing with Release 8.1. A user ID (UID) is added to the Gen-2 certificate in the DS8000 by manufacturing, thus connecting the DS8000 to the KMIP capable key server by using Client Certificate Authentication for SSL Session and User ID. It is the most secure way.

Not using Client Certificate Authentication (policy 1) is not a preferred practice, so it is not covered in this paper. Both the Client Certificate Authentication used for SSL session only (policy 2) and the two-factor authentication by enabling Client Certificate Authentication for SSL and configuring the Username (policy 3) are preferred and covered in this paper.

Policy 2 and 3 prerequisites

Before starting the Gemalto SafeNet KeySecure Configuration, make sure to satisfy the following prerequisites:

- Make sure that you configured two independent key servers in a cluster.
- Have the recovery key configured, as described in “Creating the recovery key” on page 99.
- Have the DS8000 certificate updated from Gen-1 to Gen-2, as described in “Migration from a Gen-1 to a Gen-2 certificate for encryption” on page 141.
- Have the root certificate for DS8000 downloaded to the client computer from the IBM DS8000 IBM Knowledge Center.

- Policy 3 only: Have the Gen-2 certificate exported to the client computer to extract the UID. For more information about how to export it and how to extract the UID from it, see “Exporting the DS8000 Gen-2 certificate” on page 81 and “Extracting the UID field from this certificate” on page 81.

Note: DS8000 Release 8.1 comes with a Gen-2 certificate from manufacturing by default.

Exporting the DS8000 Gen-2 certificate

To export the Gen-2 certificate, run the DSCLI command **`managekeygrp`**, as shown in Example 4-10.

Example 4-10 Export the Gen-2 certificate

```
dscli> managekeygrp -action exportcert -certType GEN2 -loc
c:\temp\smoker1h_gen2_cert.pem 1
Date/Time: 17. Mai 2016 11:45:26 MST IBM DSCLI Version: 7.8.10.321 DS:
IBM.2107-75LR811
CMUC00490I managekeygrp: The certificate for encryption group 1 has been exported.
dscli>
```

In the DS8000 GUI, click **Settings** → **Security** → **Certificate** and click **Export Public Key**, as shown in Figure 4-36.

Figure 4-36 Export Certificate from GUI

Figure 4-37 shows how the exported certificate looks.

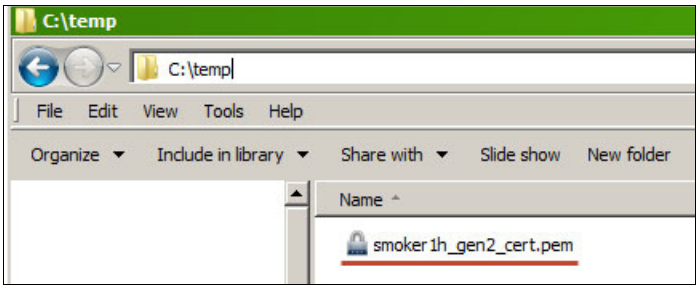


Figure 4-37 Gen-2 Certificate in c:\temp

Extracting the UID field from this certificate

In UNIX based operating systems, to extract the UID field from this certificate, run **`openssl`**, as shown in Example 4-11 on page 82.

Example 4-11 Extract the UID

```
[root@sklm-rehl64 tmp]# openssl x509 -in smoker1h_gen2_cert.pem -text | grep
Subject:
        Subject: UID=DS8K-2107-75LR811, C=US, O=ibmDisk, CN=2107-75LR811
[root@sklm-rehl64 tmp]#
```

Important: Save the UID. It is required in a later step.

In Windows, you can use the Certificate Manager to read the UID by completing the following steps:

1. First, run CertManager, as shown in Figure 4-38.

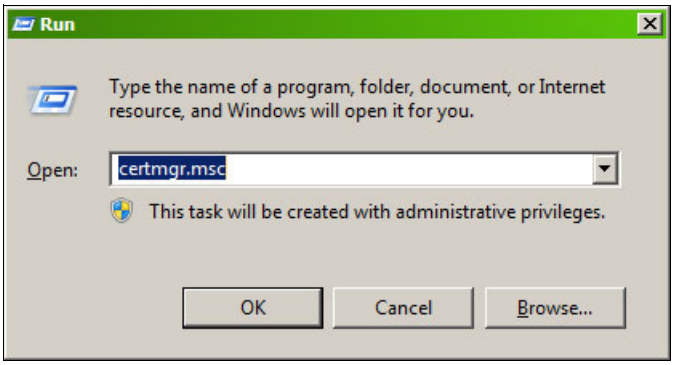


Figure 4-38 Run CertMgr

2. Click **Personal** → **Certificates** and click **All Tasks** → **Import**, as shown in Figure 4-39.

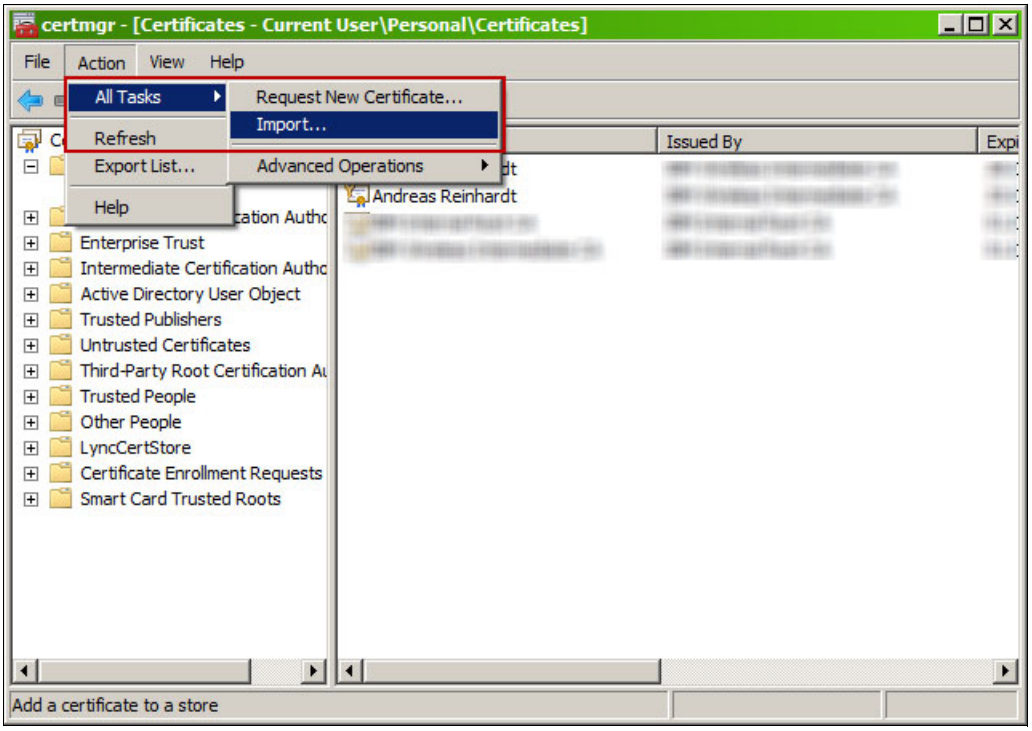


Figure 4-39 Navigate to Import

3. Follow the wizard to import the certificate. Make sure to select **“All Files (*.*)”** to see the certificate in .pem format, as shown in Figure 4-40.

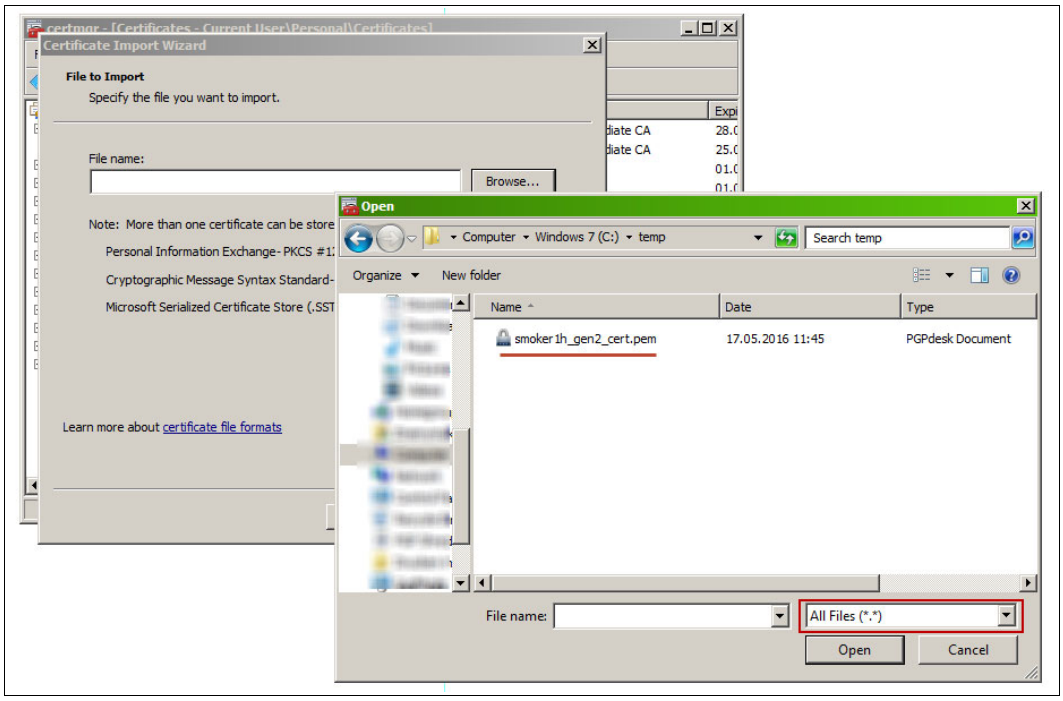


Figure 4-40 Select “All Files (.*)”*

4. The certificate is now imported, as shown in Figure 4-41.

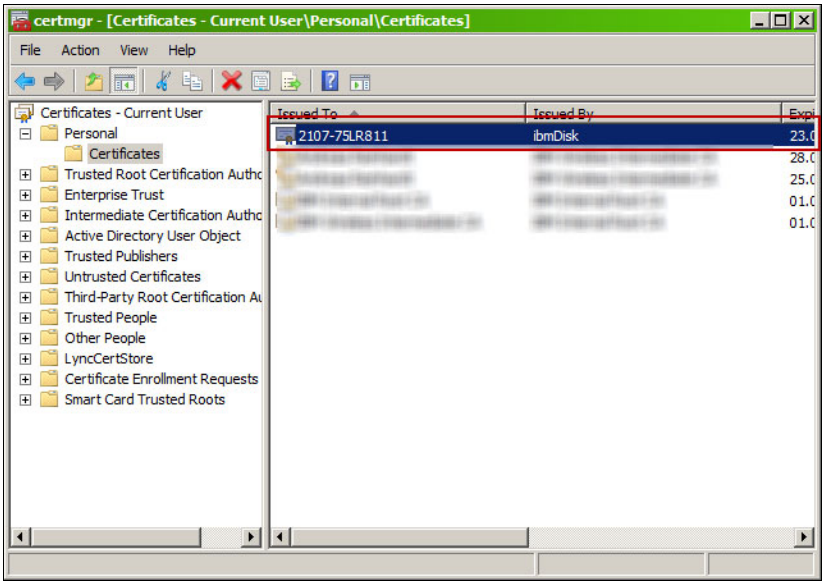


Figure 4-41 The imported certificate

5. Open the certificate, navigate to the Details tab, and select **Subject**. Figure 4-42 shows the UID.

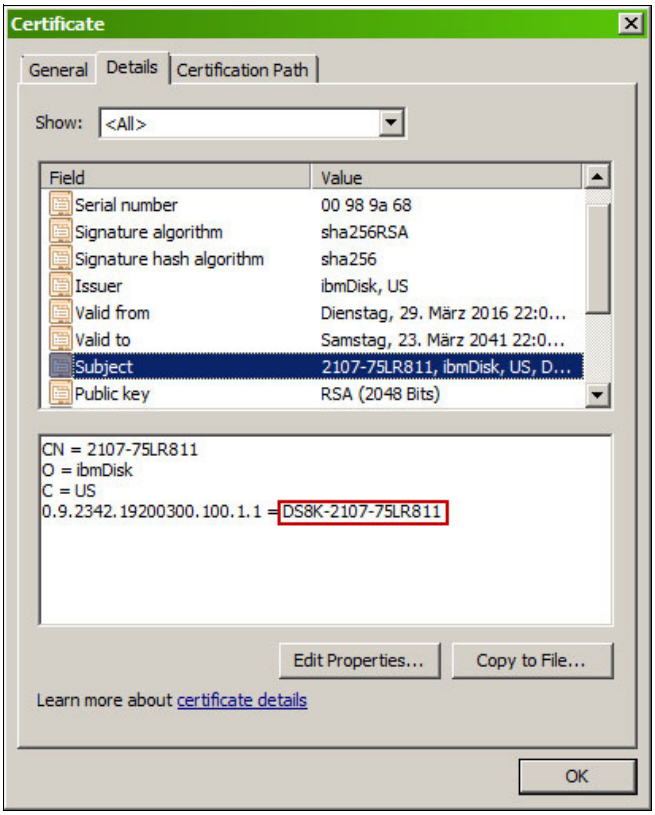


Figure 4-42 Show the UID

Important: Save the UID. It is required in a later step.

You can delete the Gen-2 certificate from the Windows keystore and Windows / UNIX hard disk drive (HDD) now.

Getting the Gen-1 or Gen-2 root certificate

If you upgraded your DS8880, you might still have a Gen-1 certificate. A system at Release 8.1 that is delivered from manufacturing has a Gen-2 certificate. To update your Gen-1 certificate, see 4.8, “Migration from a Gen-1 to a Gen-2 certificate for encryption” on page 141.

If you do not have access to the DS8000 IBM Knowledge Center, use the root certificates from the Example 4-12 (Gen-1) and Example 4-13 (Gen-2). Make sure to copy everything, including `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.

Example 4-12 Gen-1 root certificate

```
-----BEGIN CERTIFICATE-----
MIIDNzCCAh+gAwIBAgIBZzANBgkqhkiG9w0BAQUFADArMQswCQYDVQQGEwJVUzEM
MAoGA1UEChMDSUJNMQ4wDAYDVQQDEwVzdWJjYTAeFw0xNjA1MTExMzAzMzJaFw0z
NjA1MDYxMzAzMzJaMCsxCzAJBgNVBAYTAlVTMQwwCgYDVQQKEwNJQk0xDjAMBgNV
BAMTBXN1YmNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt7qPZB1y
O4pRaLSq3gB5nU+IbDLn7tWaTtBul1j+4EOZIuARUejobqFTASHz3+pxW8CKFY/p
9aw5p9JGSdIruSmpPidtYSPCFvs9PNRKlNzdzAOKgj89DNYS4InucUp5hAOMEzVm
gtbNOTMGUpaqYZeq3oMmOw2GNfM1YQcrrlm+LRUXiigz39DianpAWLXHWG9KVPAk
V9DGlHuMEk3FtaExKCop+w2DUfPzoSSRKXYabWjICErcj5G+xjMXWXLkdjXtt8lV
TbcU46RwkdzFsQMhrlsgu/brTG4xAxlDrD8R+QI95pfoF4/nx4rJECGGPyM1YnIJ
zbj93hXv2sM99QIDAQABo2YwZDAdBgNVHQ4EFgQUuMVQ/SlMpgJADhXMI2IHlEHf
BTUwHwYDVR0jBBgwFoAUuMVQ/SlMpgJADhXMI2IHlEHfBTUwEgYDVR0TAQH/BAgw
BgEB/wIBADAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQEFBQADggEBABsxKXgc
8QkWmj3ujLq1oniTW8Oqx1K9OGtJ/YAzYLSVb7CE+jrKRD7W18cDGdLA5od7NaQt
r6yX/eeOVAeQeagmo7CD9PndiBTNoShmybkFXIJOcfRUvvboE61umhQ5A9ht3rZI
8vrUGhWfXpOAhyDQkX3m6XotbplezeE3R41ZtH7ANS1zXXlZJqFpqR4m2/FBS/yg
Q2mU46AoOG6VgcrFaRXjHMvaPg3PlABzQ6Mm7M/Rc2BkRaWDNMK3DJAt9osMqfwu
TZ8PhRvROMKX1MvrG8n+waOFPvjv1hpIVF+JJoaH6hZje+lu46J+4+1QkoVMOedO
KnJ2fx7l1IS5JSY=
-----END CERTIFICATE-----
```

Example 4-13 Gen-2 root certificate

```
-----BEGIN CERTIFICATE-----
MIIDHzCCAgegAwIBAgIBADANBgkqhkiG9w0BAQsFADAfMQswCQYDVQQGEwJVUzEQ
MA4GA1UEChMHaWJtRGlzazAeFw0xMjA4MTcwMDM2NTVaFw0zMjA4MTIwMDM2NTVa
MB8xCzAJBgNVBAYTAlVTMRAwDgYDVQQKEwdpYm1EaXNrMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAptHo9ET6dtKKUqWHzqSOlWJlQI7lKBTn5niD/XwH
mBkZhRWCYkoM/GZvlk4YGvYvoxHmsvuADkO+/Sj2ZqlCOr5mWNdX4xqSuPO7xNT7
jUDt6E/39TpQS+2svseHrO7XCmf9qCncYW29Klyv5UQtvoszOvlGmwlz2WnF7qwu
CiMsqd6WoYsmRgStXGGGCuKHbPMJi7jKBgRf1QKmpIDw1NM/dA4lZElg/Nu+EHKq
KcbKzAcrx4PBY/7rLE9nkpqzzooH5Z/3s5tq7MOgHEuxJD86mLzug9kQOuPOVP5K
j/Oq+/CYVHqOAOwo3KuN8Ft78u2+c21RimoDW3kUhepizQIDAQABo2YwZDAdBgNV
HQ4EFgQUN4o9G4sMHvLCxe3qjGWpB/3CqeQwDgYDVR0PAQH/BAQDAgIEMBIGA1Ud
EwEB/wQIMAYBAf8CAQEwHwYDVR0jBBgwFoAUN4o9G4sMHvLCxe3qjGWpB/3CqeQw
DQYJKoZIhvcNAQELBQADggEBAGq6kM7n7IRVZS32uj1FYuB4PjWTYGRqm7HCYmIX
8zFpszPOBg9DWbtntQSXrVJV5u8lIyoU3m5ARgGWNKKEttHGLpF2M91ZxkkNyyhu
v1q+bPWt+jv1A7TfnvzxXpTx9jKrkSApuANP5AjMXZzVpem/pVM8DND8GFewSfKc
/CQacdGvE1SuXoaxUNWjC11RErvoEB2ty3B6Sf+snOecnD/iSRvOAR5q/2qY/vIM
7AURXz+XyrB1OLHRKOCHOwY+3AVKcQJUOulC9/qnof8c1gtKL+mc896vSRsGBaxR
hj8BbJAfD+xMf7Y4Ch9O4fjisSFWL9NX464wIjbaJhdqQWo=
-----END CERTIFICATE-----
```

The Gen-1 and Gen-2 root certificates can also be downloaded from the DS8000 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/ST5GLJ_8.1.0/com.ibm.storage.ssic.help.doc/ds8_gen1_gen2_cert.html

4.3.2 Configuration

There are five steps to configuring the KMIP server immediately after installation. SSL is mandatory for KMIP and must be configured. You must complete the following steps in order:

1. Create a self-signed SSL server certificate, or use a public CA with CSR on every Gemalto SafeNet KeySecure server.
2. Install the DS8000 root certificate from the DS8000 IBM Knowledge Center.
3. Create a Trusted CA List and add the known CA.
4. Add a KMIP device and edit it.
5. Add a user to key server, based on the UID from the Gen-2 certificate (policy 3 only).

The Gemalto SafeNet KeySecure installation secures HTTPS transport with a self-signed certificate by default. Depending on the browser and version that is used, an exception can occur. In that case, you must accept the certificate as a trusted certificate.

Creating a self-signed SSL server certificate

The self-signed SSL server certificate must be created on every Gemalto SafeNet KeySecure server.

Important: SSL server certificates are not replicated between the servers. Every SSL server certificate must have the exact same name.

Complete the following steps:

1. Log in as Admin to the Gemalto SafeNet KeySecure GUI by pointing your browser to the address of the key server by using the format `https://(ip address>:<ip port>`. The default port is 9443.
2. Navigate to **Security** → **SSL Certificates** and create a certificate request, as shown in Figure 4-43.

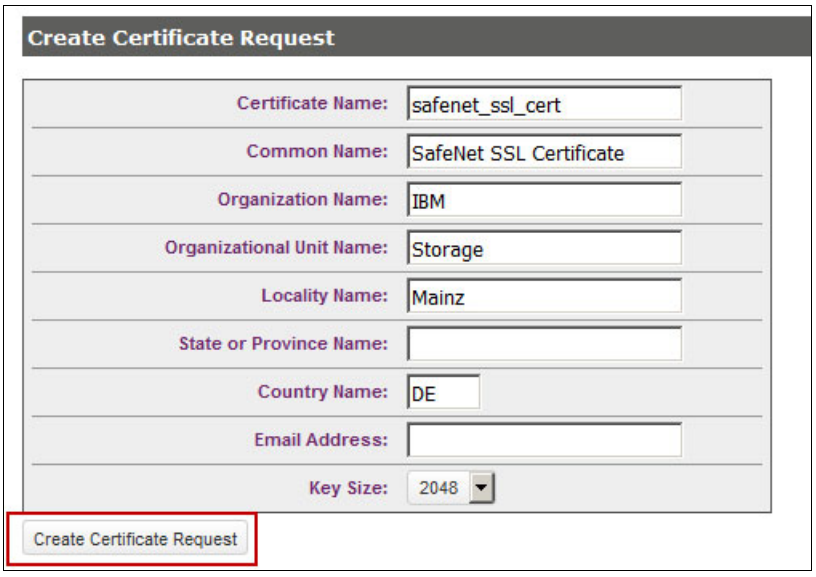


Figure 4-43 Create a certificate

3. After the SSL certificate is created, select it and click **Properties**, as shown in Figure 4-44.

Figure 4-44 SSL certificate properties

4. Within the properties, select **Create Self-Signed Certificate**, as shown in Figure 4-45.

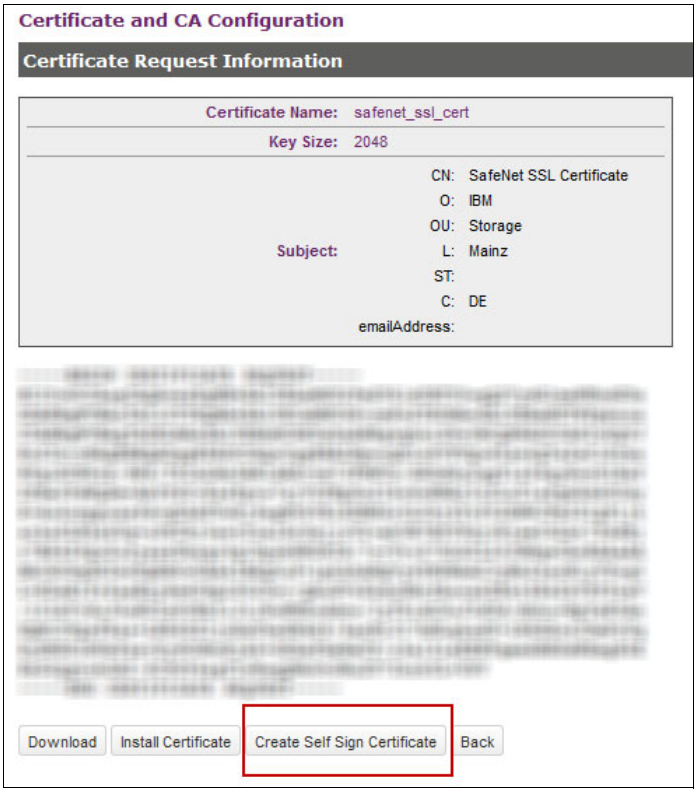


Figure 4-45 Self-sign the certificate

5. You can modify the certificate duration in days, as shown in Figure 4-46. Although the system lets you specify a maximum of 7300 days (20 years), it is advised, as a cryptographic preferred practice, to use smaller durations, such as 365 or 730 days (1 or 2 years).

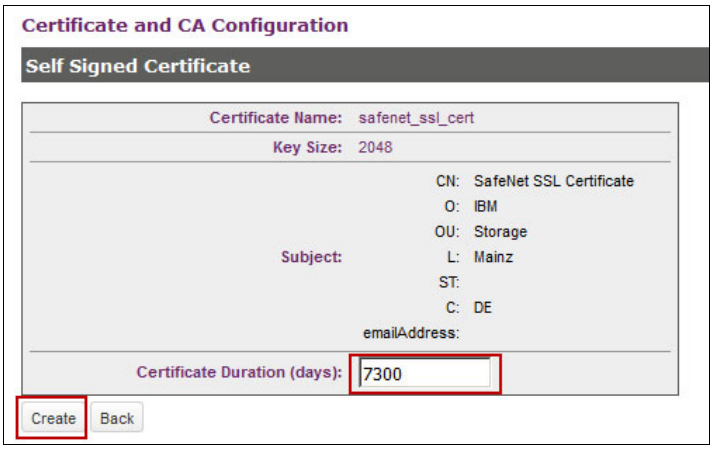


Figure 4-46 Maximum certificate duration

6. The SSL Certificate is now active, as shown in Figure 4-47.

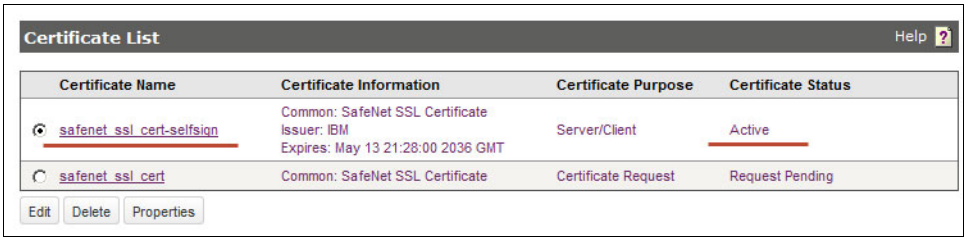


Figure 4-47 SSL Cert Ready

7. Select the self-signed SSL certificate again and select **Properties**. Then, click **Download**, as shown in Figure 4-48.

Figure 4-48 Download the CCL certificate

8. Save it to your local hard drive and do not rename it, as shown in Figure 4-49.

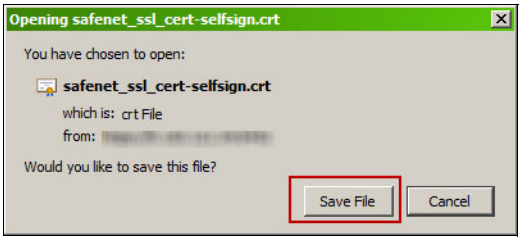


Figure 4-49 Save SSL Certificate to the hard disk drive

Important: Repeat these steps on all Gemalto SafeNet KeySecure servers.

Installing the DS8000 root certificate from the DS8000 IBM Knowledge Center

This section shows how to install the DS8000 root certificate to the Gemalto SafeNet KeySecure key server cluster. There is only one DS8000 root certificate for all machines in your environment.

From now, all steps can be performed on just one Gemalto SafeNet KeySecure key server, independently from which one you choose.

Complete the following steps:

1. Still logged in to the Gemalto SafeNet KeySecure GUI, click **Security** → **Known CAs**, as shown in Figure 4-50, paste the certificate text, and click **Install**. The name should clearly identify the DS8000 Gen-2 root certificate.

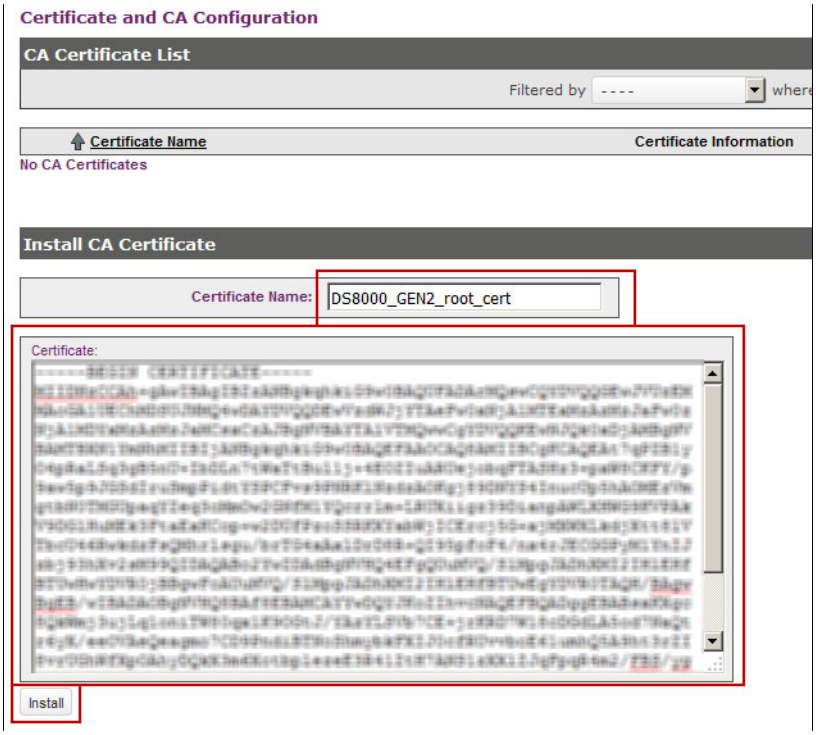


Figure 4-50 Install the DS8000 Gen-2 root certificate

2. The root certificate is now installed and active. As shown in Figure 4-51, the certificates must be added to a trusted CA list to be recognized by the KMIP server.

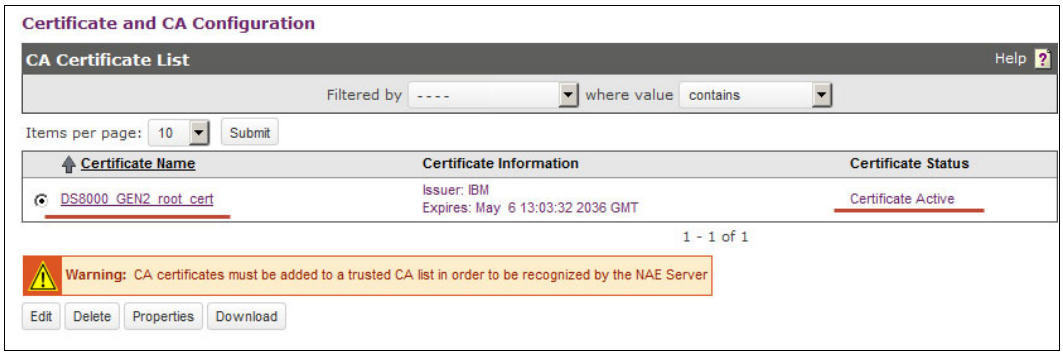


Figure 4-51 Root certificate is now installed

Creating a Trusted Certificate Authority List Profile and adding the known CA

The next step is to create a Trusted Certificate Authority List Profile for your environment and add the previously created known Certificate Authority to the profile list.

Complete the following list:

1. Still logged in to the Gemalto SafeNet KeySecure GUI, click **Security** → **Trusted CA Lists**, as shown in Figure 4-52, and click **Add**.

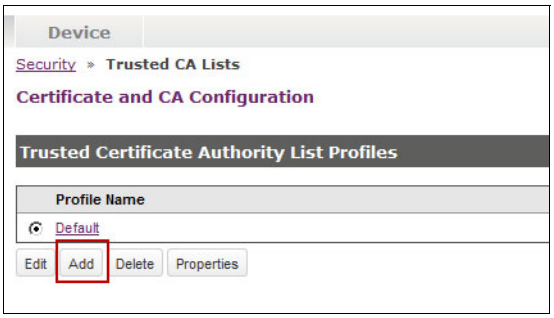


Figure 4-52 Add a profile

2. The name should clearly identify the DS8000 profile, as shown in Figure 4-53.

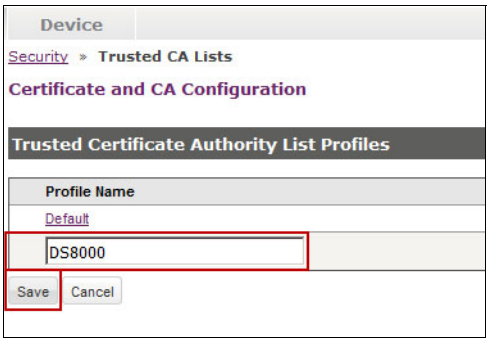


Figure 4-53 Name the profile

3. The new profile is now in the list of profiles, as shown in Figure 4-54. Select it and open the properties.

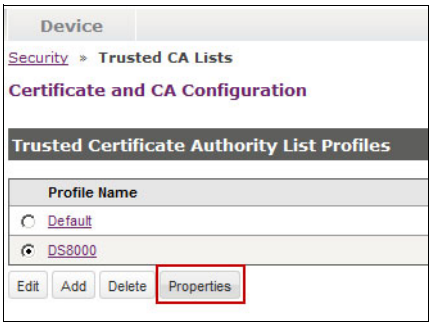


Figure 4-54 Profile properties

4. In the profile properties (Figure 4-55), click **Edit**.

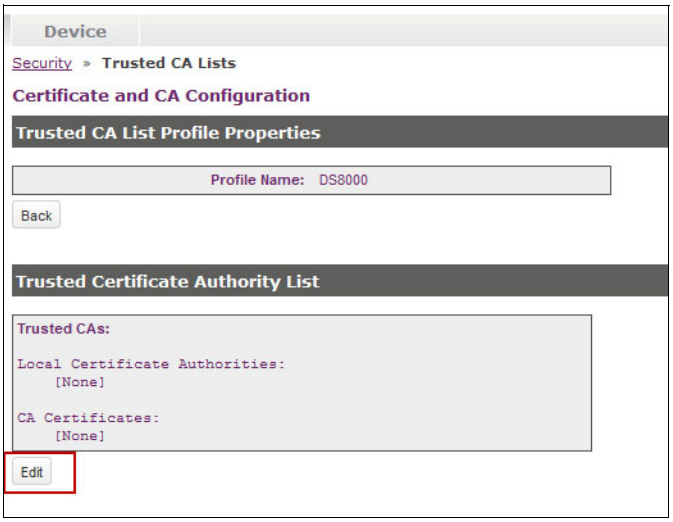


Figure 4-55 Edit the profile

5. Add the previously created DS8000 root CA to the list of trusted CAs and save it, as shown in Figure 4-56.

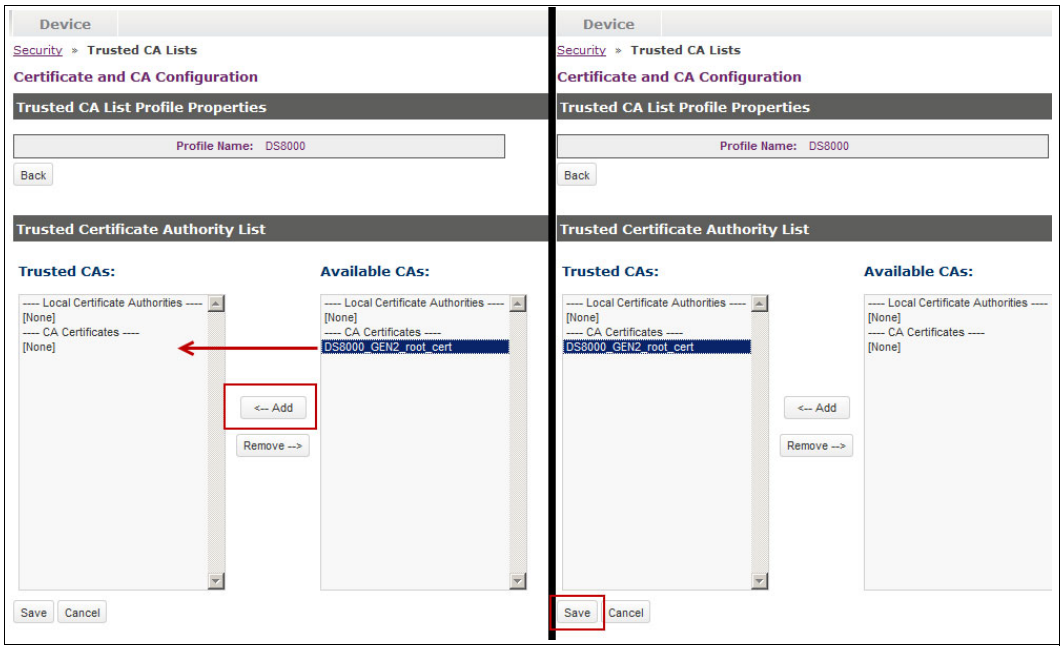


Figure 4-56 Add the DS8000 root CA to the list of trusted CAs

Adding a KMIP device to Gemalto SafeNet KeySecure

Now, a KMIP Cryptographic Key Server protocol must be configured in your environment to serve the keys to the DS8000. Complete the following steps:

1. Still logged in to the Gemalto SafeNet KeySecure GUI, click **Device** → **Key Server** and click **Add** to create a protocol, as shown in Figure 4-57. There already is one NAE-XML protocol that is preconfigured, which can be ignored.

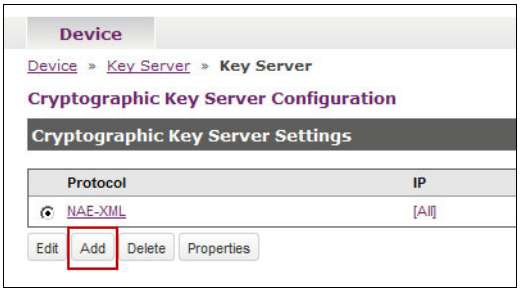


Figure 4-57 Add KMIP protocol

2. Figure 4-58 shows the settings for the KMIP protocol that must be selected:
   - Protocol: KMIP.
   - IP: All.
   - Port: 5696 (This is the default KMIP port for the DS8000.)
   - Use SSL: Select it.
   - Server Certificate: Select the Servers SSL certificate that you created in “Creating a self-signed SSL server certificate” on page 86.

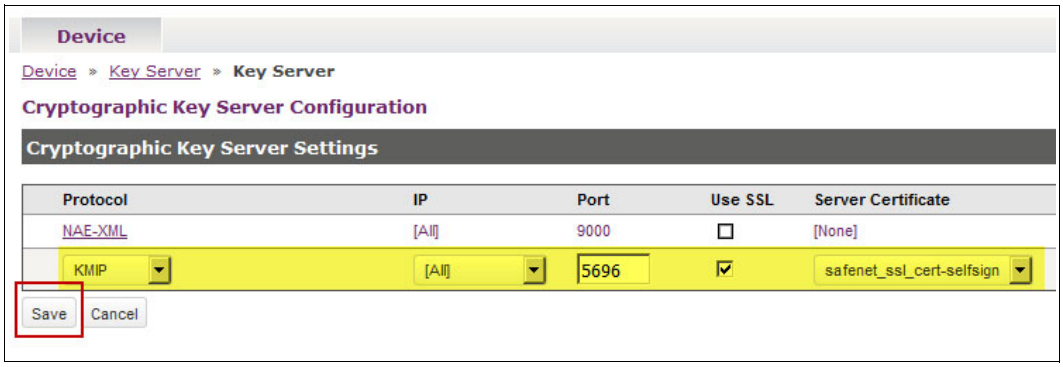


Figure 4-58 Define KMIP protocol settings

3. The KMIP protocol appears in the list, as shown in Figure 4-59. Select it and click **Properties**.

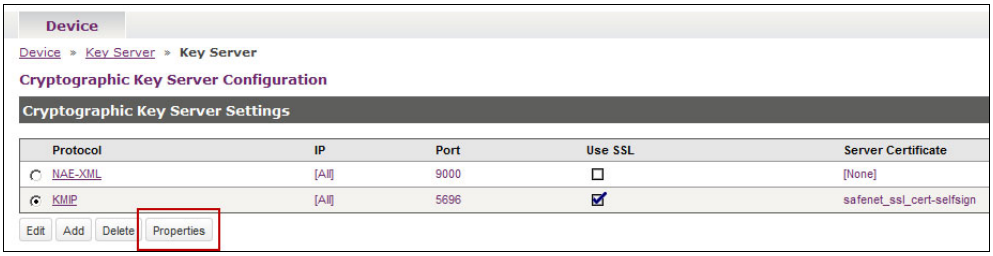


Figure 4-59 KMIP protocol created

4. In the KMIP Protocol properties, click **Edit** in the Authentication Settings area, as shown in Figure 4-60.

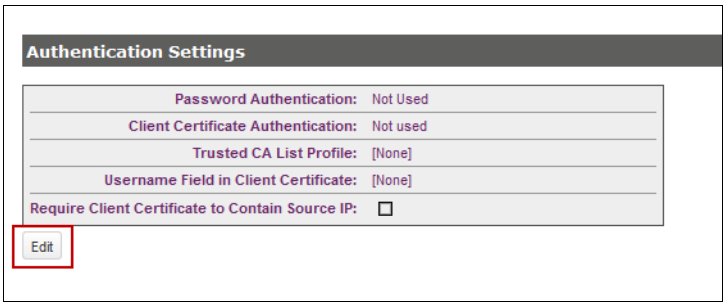


Figure 4-60 KMIP Authentication Settings

5. The Authentication Settings must be modified as follows:
   - Password Authentication: “Not Used”.
   - Client Certificate Authentication:
     - If you have at least one DS8000 that was upgraded from Release 8.0 to release 8.1 in your encryption environment, select **Used for SSL session only**. DS8000 Release 8.0 systems do not have an UID field in the Gen-2 certificate.
     - If you have one or more DS8000 disk storage systems that were delivered with Release 8.1 or later, select **Used for SSL session and username (most secure)**. DS8000 Release 8.1 and later systems from manufacturing do have an UID field in the Gen-2 certificate. However, it is possible to use “Used for SSL session only”.

- Trusted CA List Profile: Select the previously created Trusted CA List Profile.
- Username Field in Client Certificate: Select **UID (User ID)** for DS8000 disk storage systems that were delivered with Release 8.1 and **None** for DS8000 disk storage systems that were upgraded from Release 8.0 to Release 8.1.
- Require Client Certificate to Contain Source IP: Do not select it.

6. A full configured KMIP profile is shown in Figure 4-61. Click **Save** when you are ready.

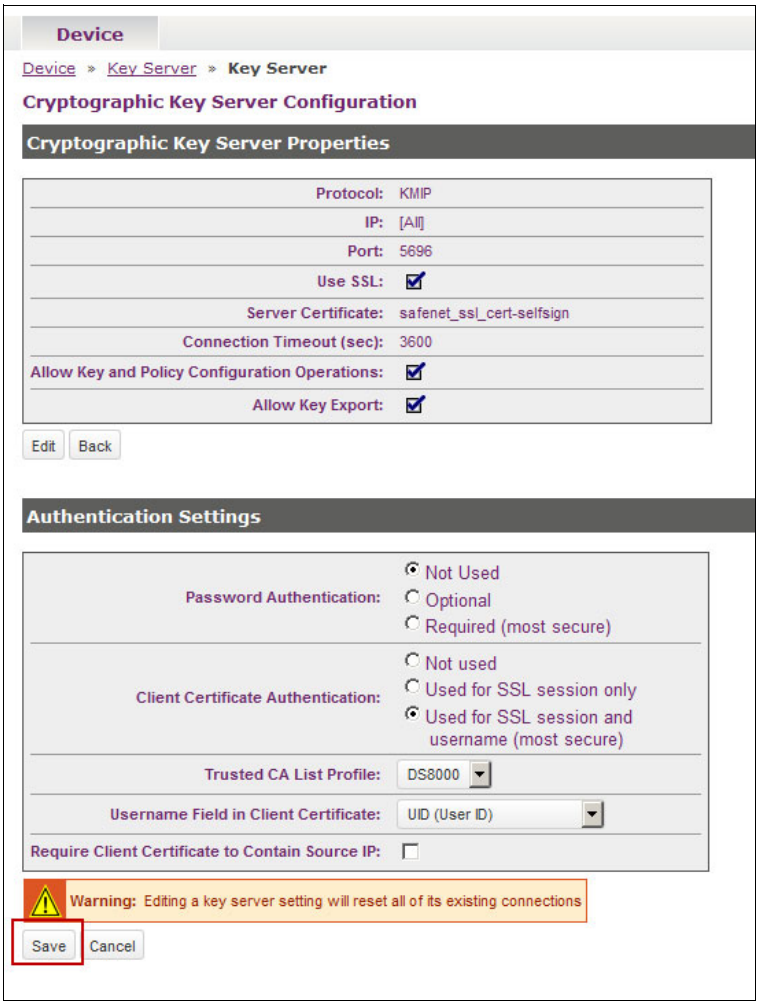


Figure 4-61 Configured KMIP profile

Adding a user to a key server based on the UID from the Gen-2 certificate

The final step is to create a user account for each DS8000 Release 8.1 in your environment that will be encrypted.

Complete the following steps:

1. Still logged in to the Gemalto SafeNet KeySecure GUI, click **Security** → **Local Authentication** → **Local Users & Groups** and click **Add** to add a user, as shown in Figure 4-62.

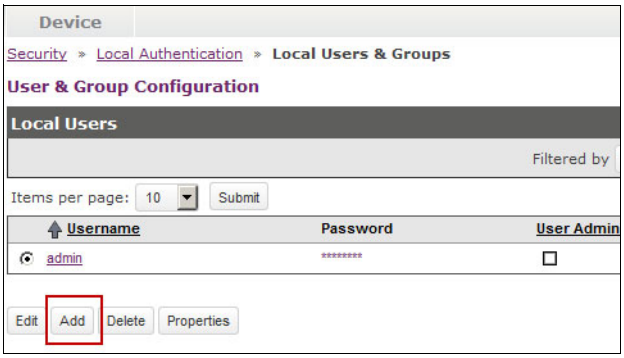


Figure 4-62 Add a DS8000 user

2. The user ID equals the UID that you extracted from the DS8000 Gen-2 certificate in “Extracting the UID field from this certificate” on page 81. In Figure 4-63, the user ID is DS8K-2107-75LR811. Add as many different user IDs as you require in your environment and click **Save**. Complete the following fields:
   - Password: This can be anything. The password is not supported by the DS8000.
   - User Administration Permission: Do not check it.
   - Change Password Permission: Do not check it.

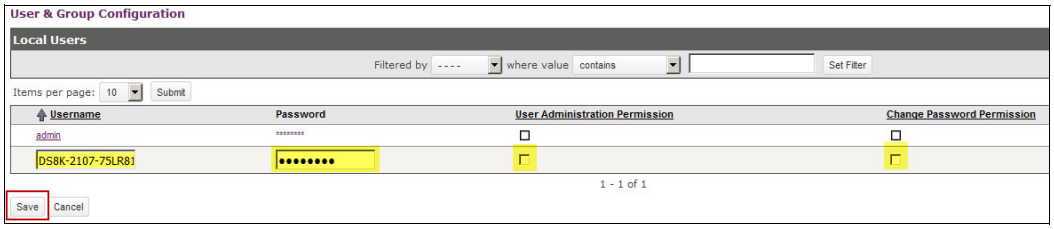


Figure 4-63 User details

The Gemalto SafeNet KeySecure server is now ready to serve keys to the DS8000.

4.4 DS8000 GUI configuration for encryption

This section explains how to configure the disk encryption on the DS8000 by using the Storage Manager GUI. The high-level configuration sequence includes these steps:

1. Apply the *drive encryption authorization* license key.
2. Assign additional storage and Security Administrators.
3. Create the recovery key.
4. Enable encryption with IBM Security Key Lifecycle Manager:
   a. Define the key labels.
   b. Configure the key server connection to the DS8000.
   c. Authorize the recovery key.
5. Enable encryption with Gemalto SafeNet KeySecure:
   a. Configure the key server connection to the DS8000.
   b. Authorize the recovery key.
6. Configure and administer encrypted arrays, ranks, and extent pools.

For information about enabling NIST SP 800-131a-compliant encryption certificates and TLS v1.2 communication, see 4.7, “NIST SP 800-131a requirements for key servers” on page 136.

4.4.1 Applying the drive encryption authorization license key

The *drive encryption authorization* license key is one of the main DS8000 encryption requirements. The key can be obtained from the IBM Data Storage Feature Activation (DFSA) website. For a new DS8000 installation, the GUI starts the System Setup wizard. The setup wizard is started automatically after the first logon. One of the tasks is to activate all available license keys.

The DS8000 System Setup wizard process, including key activation, is explained in *IBM DS8880 Architecture and Implementation (Release 8)*, SG24-8323.

To check whether the drive encryption authorization license key is installed, click **Settings** → **System**, as shown in the Figure 4-64 on page 96.

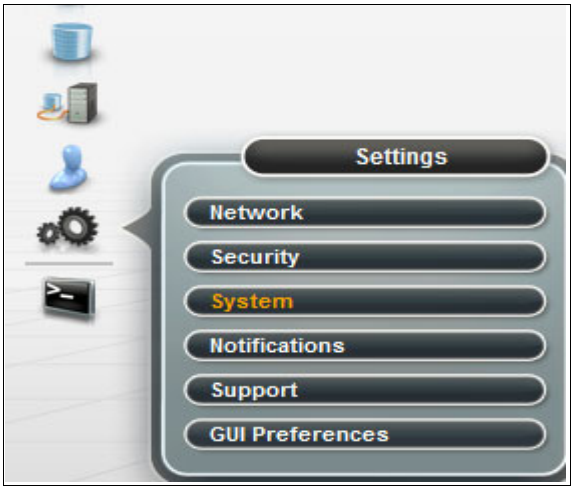


Figure 4-64 Navigate to the Licensed Functions window

The example in Figure 4-65 highlights the enabled drive encryption authorization license for all storage types (CKD and FB).

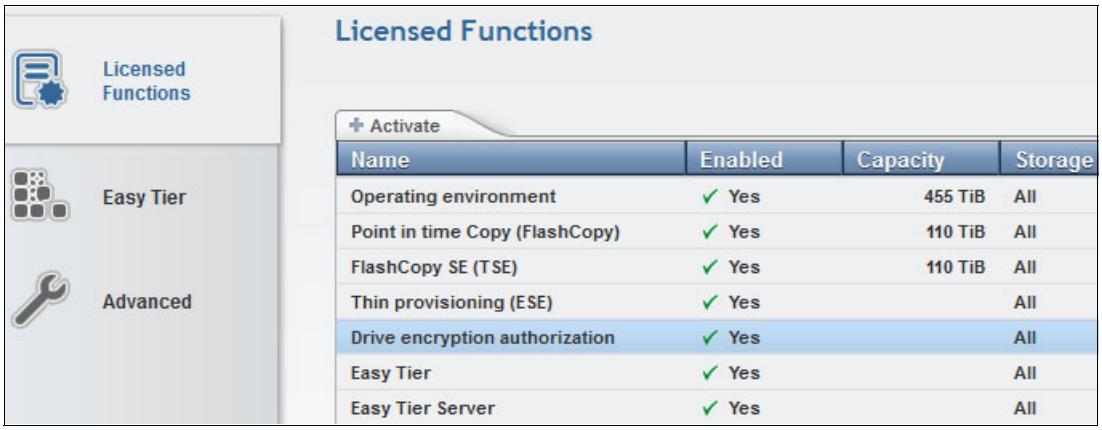


Figure 4-65 Licensed Functions

4.4.2 Assigning additional storage and Security Administrators

The DS8000 includes an internal authentication and authorization service that is called the *basic authentication service*. This service also provides local user management. The DS8000 allows you to also use an external authentication service, such as an LDAP server, but still use the internal authorization service to grant access to resources as defined by these DS8000 user group roles:

- admin
- secadmin
- op_storage
- op_volume
- op_copy_services
- service
- monitor

With the introduction of the encryption recovery key on DS8000, a *dual control* security process is required to prevent unauthorized use of the recovery key. This dual control process requires two separate user accounts to process most recovery commands. If these accounts are owned by two separate people, the recovery key cannot be used by any one person to gain access to encrypted data.

The first user role is in the *admin* user group and is called *Storage Administrator*. The second user role, *secadmin*, is called *Security Administrator*. Both users are created on the DS8000 by default, and you must assign these roles to two individuals.

To define new user IDs or modify the default user names, complete the following steps:

1. Sign on to the DS8000 GUI with Storage Administrator or Security Administrator privileges, depending on which role is needed.
2. From the left pane of the Welcome window, select the **User Access** icon and select **Users**. as shown Figure 4-66.

Passwords: The initial admin password is *admin* and the secadmin password is *secadmin*. The first time that you log in, you must change the password to a new one. Because these users are owned by different people, the admin and secadmin passwords must not be stored in one place.

Figure 4-66 Navigate to the User Administration window

The User Administration window (Figure 4-67) lists all defined users. The default users are admin and secadmin.

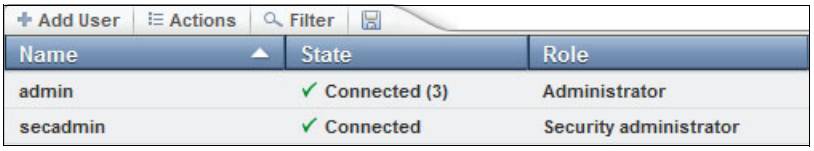


Figure 4-67 User Administration window

3. From the menu, click **Add User** to create users. The Add User window opens, as shown in Figure 4-68.

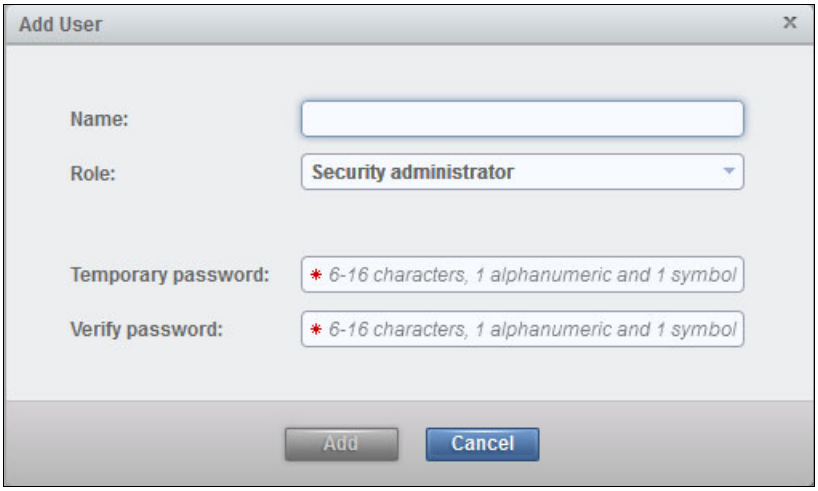


Figure 4-68 Add user

A user who has the Security Administrator authority cannot have the authority of any other user role. Also, a user with any other user role cannot have the Security Administrator

authority concurrently. The secadmin user can create only users with the Security Administrator authority. All other authorities are disabled if you are logged in as the user belonging to the secadmin group authority. Any attempt to select any other role results in an error message, as shown in Figure 4-69.

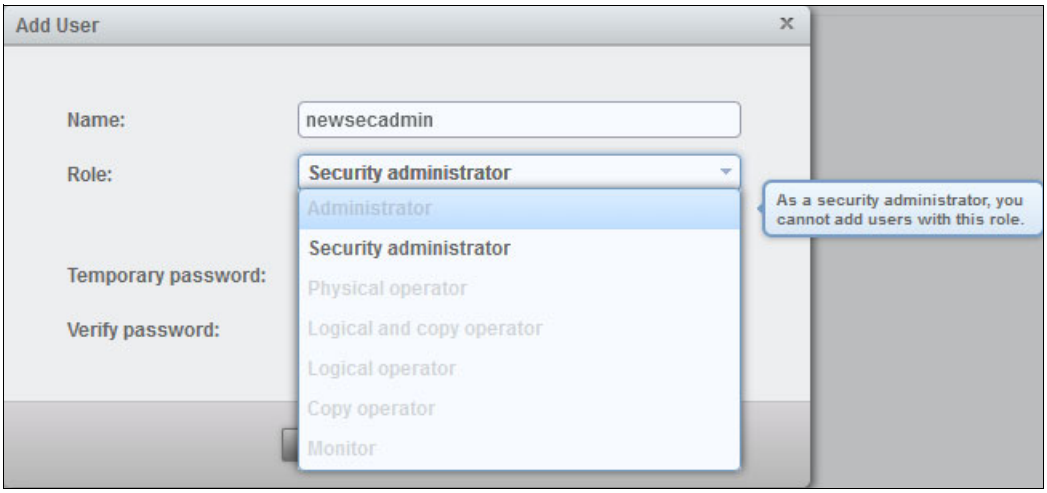


Figure 4-69 Add user by using the Security administrator role

4. After you input the new user name, Security Administrator role, and password, click **Add** to complete this task.

4.4.3 Creating the recovery key

Whenever an encryption technology is applied, a new type of risk appears: *deadlock*. This situation happens, for example, when a DS8870 cannot obtain a required data key from the key server because no key server can communicate with the DS8000. As a consequence, all data on the DS8000 becomes inaccessible because without the keys the data cannot be decrypted anymore.

The risk of a deadlock can be substantially minimized by maintaining redundant (dual-platform) key servers, but it cannot be eliminated. The *recovery key* feature provides a way to get out of a deadlocked state.

Without the recovery key, the DS8000 data becomes unrecoverable if you permanently loose access to the key severs (typically if the key servers are corrupted and not recoverable).

The decision about whether to create the recovery key must be made at this stage. Creation enables the recovery key automatically. If you decide to proceed without creating the recovery key, you can enable it later. However, all DS8000 logical configuration, including volumes, ranks, and extent pools, must be removed (deleted) to do so. The same disruptive process happens if you create the recovery key during initial configuration and later decide to disable it.

To create a recovery key, complete the following steps:

1. Log in to the DS8000 GUI as a user with Security Administrator privileges. If the DS8000 does not have any extent pool that is defined yet, the window in Figure 4-70 opens.

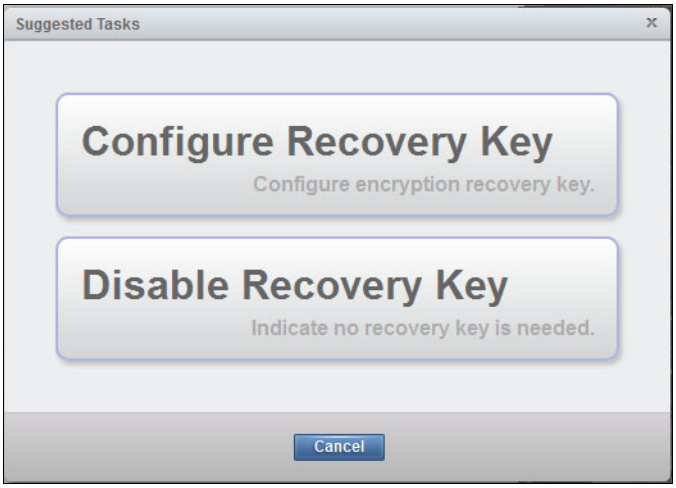


Figure 4-70 Configure Recovery Key

2. Because the DS8000 does not have any extent pools that are configured, the only choice is to select the **Configure Recovery Key** option, which automatically generates a new recovery key. It is displayed as a 64-hexadecimal character key with dashes between every four characters. The Security Administrator must record and protect this key because there is no way to view the key from the key server. You can select and copy the key. Click **Enable** to continue (see Figure 4-71).

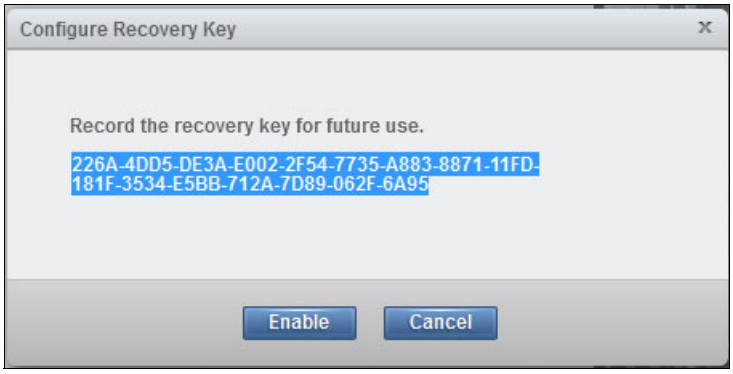


Figure 4-71 Create Recovery Key (or disable recovery key) window

Saving the key text: The Security Administrator is responsible for recording and storing the recovery key in a safe place. This key is critical to recovery of a deadlock condition. Any person that knows the recovery key can unlock the DS8000.

3. To ensure that the recovery key was recorded correctly, type it into the **Verify Recovery Key** field (Figure 4-72) and click **Verify**.

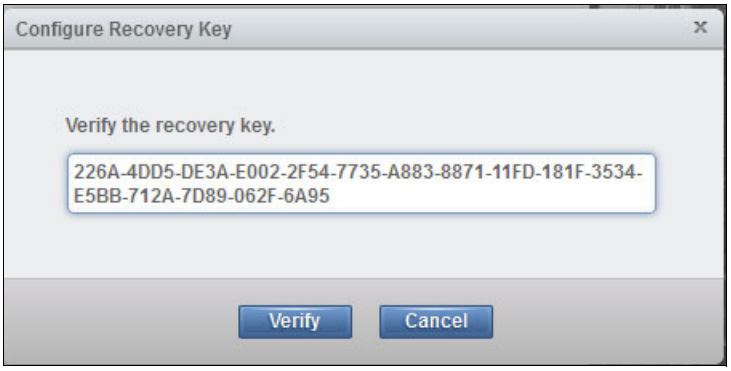


Figure 4-72 Recovery key verification

After the key is verified, it is still not active because it is waiting for the Storage Administrator to authorize the newly created recovery key. At this stage, you can log off as the Security Administrator user.

4.4.4 GUI configuration for DS8000 encryption

The data within the DS8000 that is encrypted is partitioned in one *encryption group*. The encryption group that contains encrypted data is enabled to access data through one data key that is obtained from a key server.

Note: Currently, the DS8000 supports only one encryption group. It must either be configured for IBM Proprietary Protocol (IPP) or KMIP.

An *encryption group* contains a set of *extent pools*, each of which has a set of associated *ranks* and *volumes*.

Configuring IBM Security Key Lifecycle Manager (TCP)

To configure IBM Security Key Lifecycle Manager on the TCP port, complete the following steps:

1. After the recovery key is created, you must log on to the DS8000 GUI as a user who has the Administrator role to enable the encryption and authorize the previously generated recovery key. From the DS8000 GUI Welcome window, click **Settings** and then **Security**, as shown in the Figure 4-73.

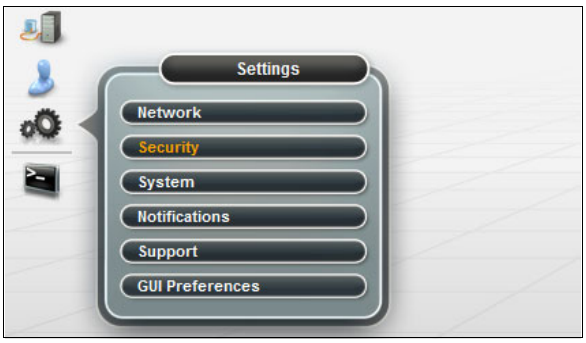


Figure 4-73 Navigate to the Encryption window

2. The encryption wizard is displayed in Figure 4-74. This wizard is started only when you enable the encryption for the first time. Click **Enable Encryption** to continue.

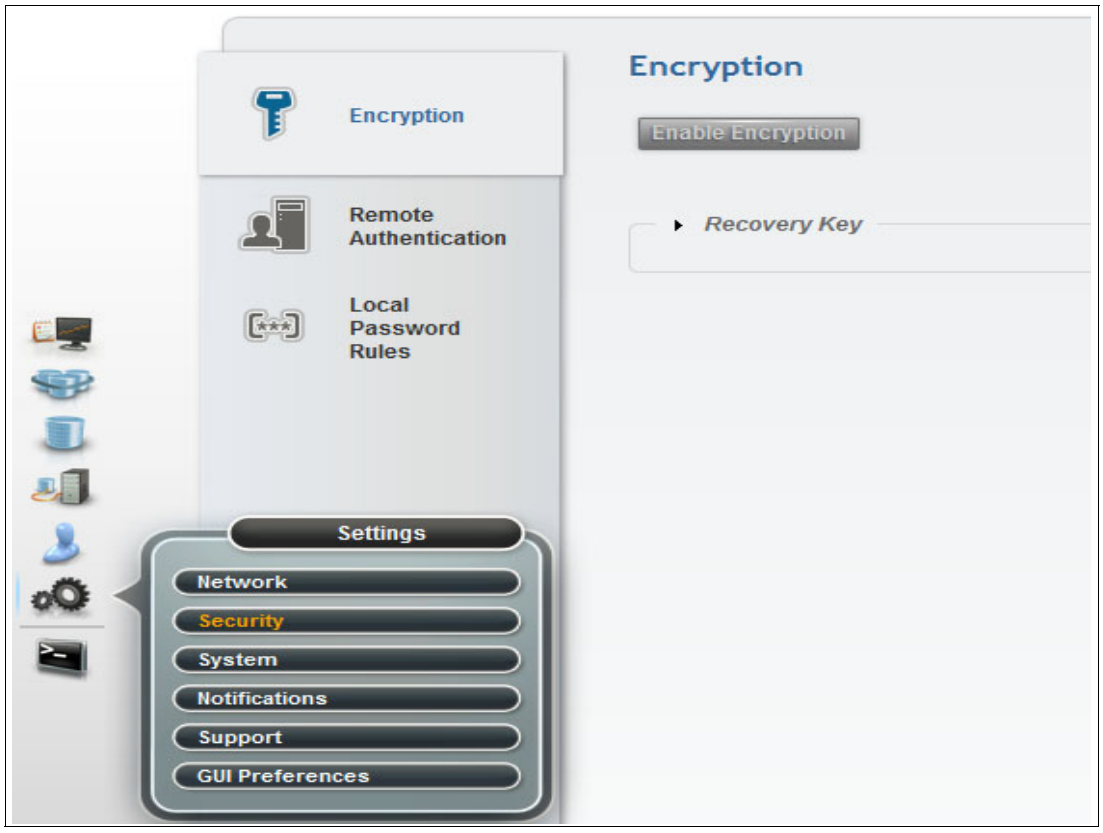


Figure 4-74 Encryption wizard

3. The welcome window opens with the basic information that is related to the prerequisites for the next steps, such as at least two key servers should be already configured and online (connected to the DS8000). Click **Next** to continue.

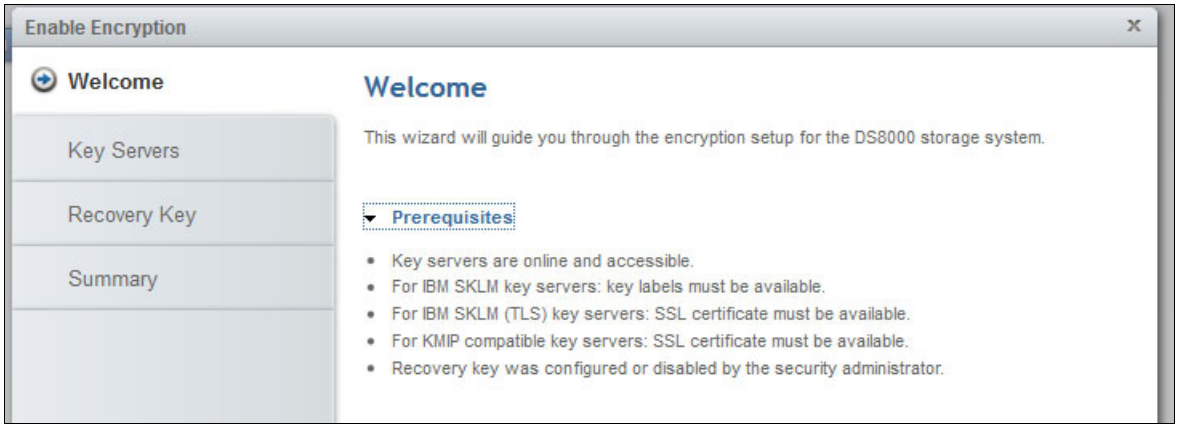


Figure 4-75 Encryption wizard: Welcome window

4. The next step is to select the Key server type. Select **IBM Security Key Lifecycle Manager** to use TCP port 3801 to communicate with IBM Security Key Lifecycle Manager.

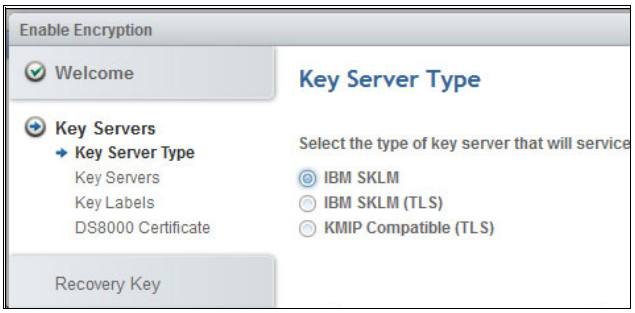


Figure 4-76 Key Server Type

5. Configure the key servers. The DS8000 supports up to four key servers. The following considerations apply to configurations:
   - In multiple site configurations, at least two of the key server ports should be assigned to isolated key servers at separate physical sites. The remaining ports can be connected to general key servers.
   - In single-site configurations, at least two of the key server ports should be assigned to isolated key servers at the same site.

   The DS8000 configuration for encryption also requires that at least two active key servers be connected and defined at the DS8000 instance.

   The DS8000 monitors all configured key servers. Client notification is provided for loss of access to key servers and other key server-related errors through DS8000 client notification mechanisms (SNMP traps and email, if configured) in the following ways:
   - Loss of access to key servers is reported at 5-minute intervals.
   - Loss of the ability for at least two key servers to provide key services that can prevent access to the data on the DS8000 is reported at 8-hour intervals.
   - The inability of any one key server to provide key services that can prevent access to data on the DS8000 is also reported at 8-hour intervals.

   In the example (see Figure 4-77), two key servers are defined. You can add or remove key servers by clicking the **+** or **-** sign next to each key server field. Specify the host address (IP address or the full qualified host name of the key server). Click **Next**.

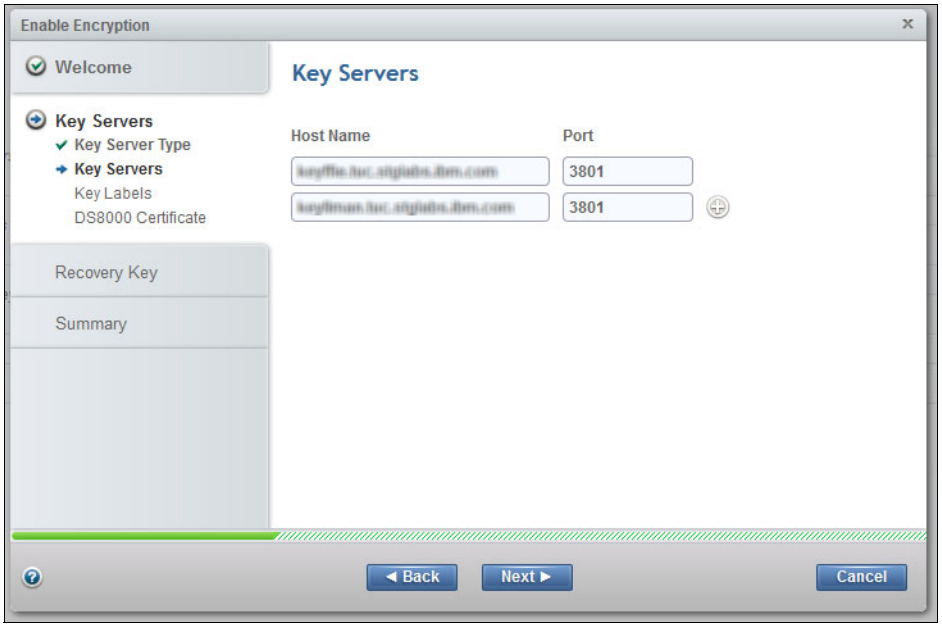


Figure 4-77 Encryption wizard: Define key servers

6. Each key server connection is tested. The message in Figure 4-78 is displayed if all the key servers that you defined in step 5 on page 103 are accessible. Click **OK**.

Note: The ports can also be changed and they should match the setting on the IBM Security Key Lifecycle Manager key server. The default TCP port is 3801.

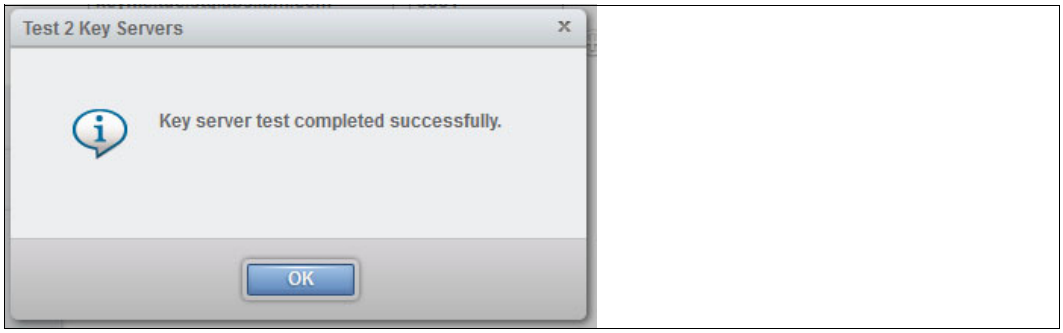


Figure 4-78 Encryption wizard: Test key servers

7. Define the key label for the data key that is generated by the IBM Security Key Lifecycle Manager server during the certificate creation step on the IBM Security Key Lifecycle Manager server. It is not required when using the KMIP protocol. This key label should match the one defined in Figure 4-4 on page 62. In this example, this key label is named `smoker1`.

   Only one key label is required in case the IBM Security Key Lifecycle Manager key servers are all installed on the open systems platforms with the same keystore type. A dual-key label option is applicable only if at least one IBM Security Key Lifecycle Manager key server is installed on a z Systems server (z/OS) and the other on the open systems platform. This is due to the different keystore type that is used on z Systems servers.

In the example in Figure 4-79, only one label is defined because all IBM Security Key Lifecycle Manager key servers are installed on the same platform with the same keystore type. Click the **+** sign to add a key label for the dual platform support. You can add the key label even after the encryption is enabled. The encryption wizard lets you continue even though the key label you provided does not match the key label you that specified in the IBM Security Key Lifecycle Manager server. The label verification is done as the last step of the encryption enablement process. Click **Next** to continue.

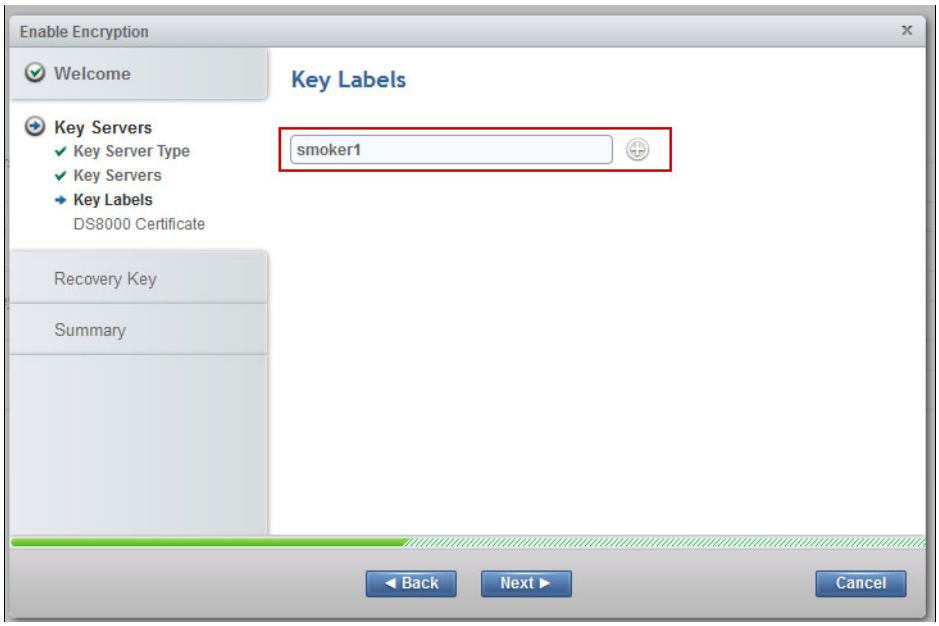


Figure 4-79 Encryption wizard: Define the key label

8. Authorize the pending request for the recovery key from the Security Administrator (if not already done). Click **Authorize**, as shown in the Figure 4-80.

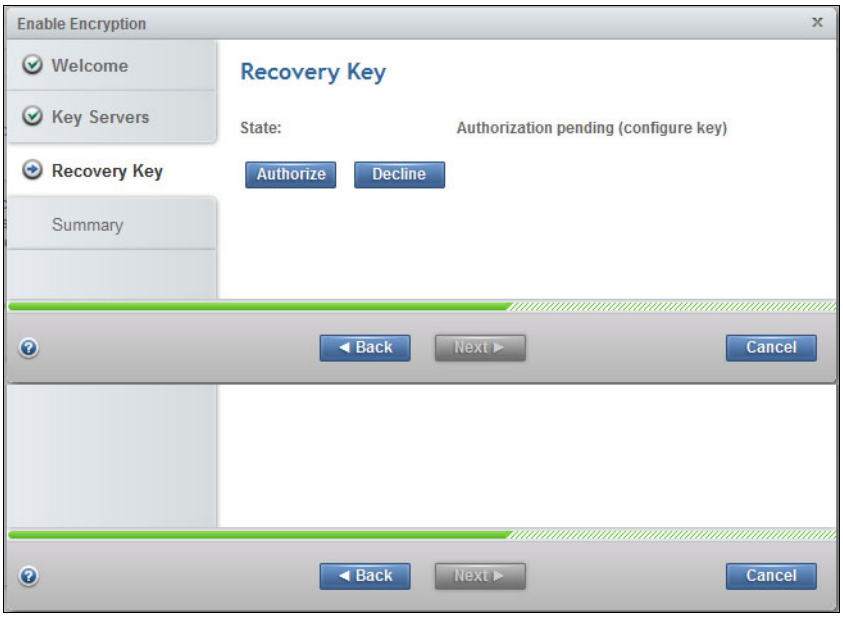


Figure 4-80 Encryption wizard: Authorize recovery key

9. The confirmation message window opens (Figure 4-81). Click **Yes** to continue.

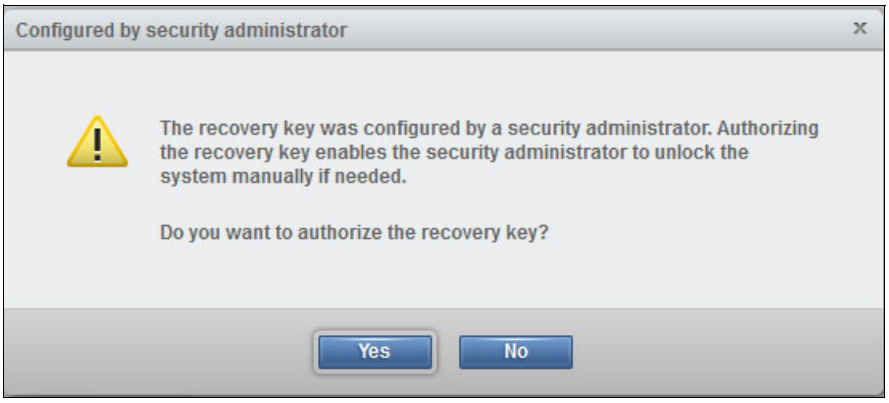


Figure 4-81 Encryption wizard: Confirm recovery key authorization

The recovery key state changes to *Configured* when the recovery key authorization is confirmed (see Figure 4-82).

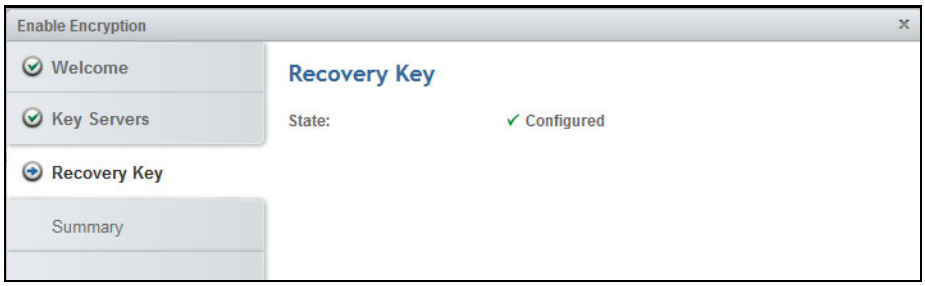


Figure 4-82 Encryption wizard: Recovery key configured

10. Figure 4-83 provides a summary of the configuration. Click **Finish** to initiate all tasks that are required to enable the encryption.

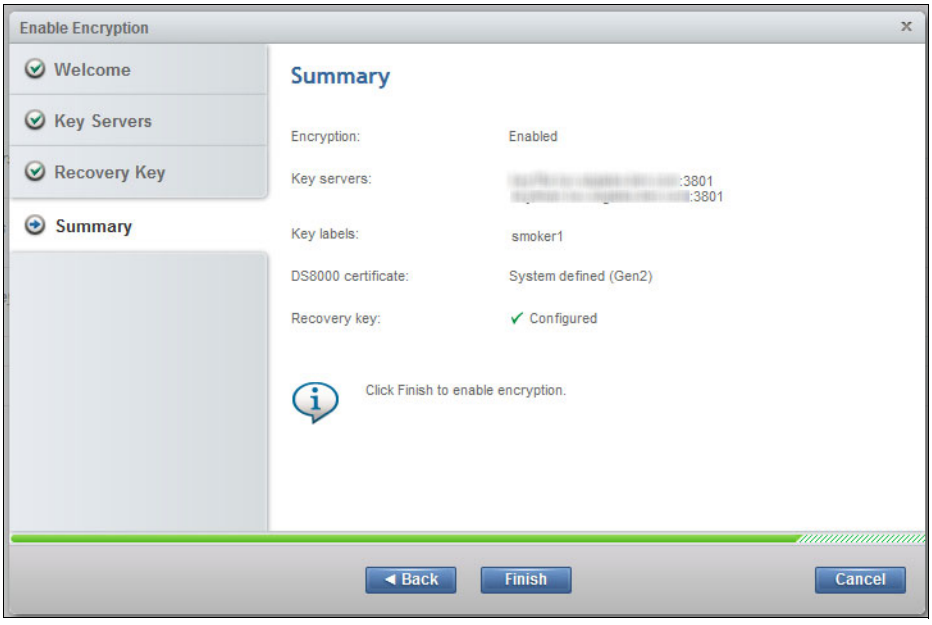


Figure 4-83 Encryption wizard: Summary

11. Encryption enablement tasks take approximately 1 minute. Expand the **View more details** section to see the task list. The overall progress is displayed as a percentage. When the Completed message is displayed, click **Close** (see Figure 4-84).

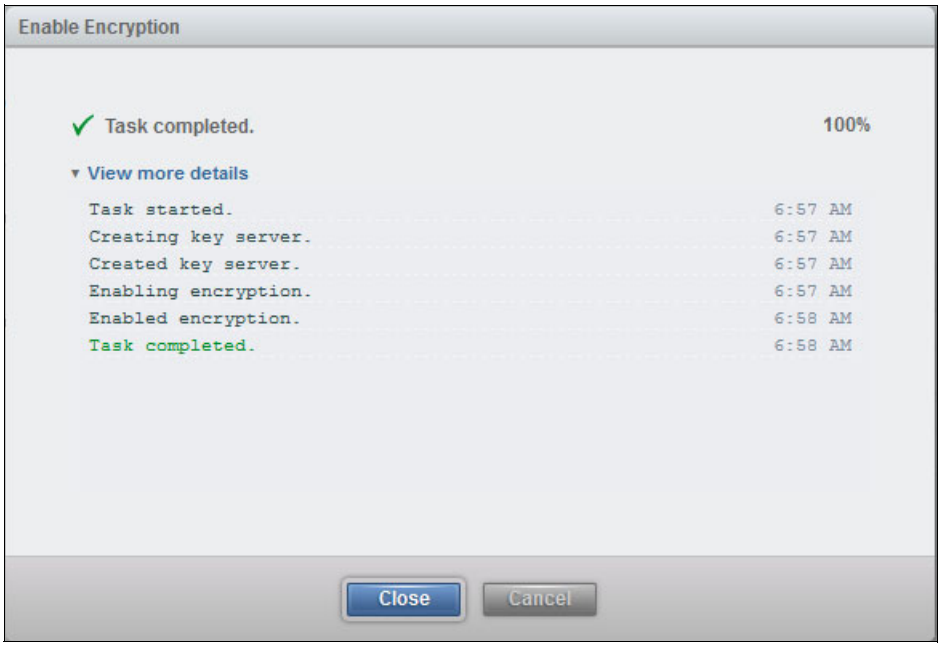


Figure 4-84 Encryption wizard: Encryption enabled

In the Encryption window (Figure 4-85), the encryption state is Enabled and the encryption key is Accessible. By expanding each section, you get more information. In this example, there is one key label and four key servers that are accessible and online.

Figure 4-85 Encryption enabled and accessible

The overall process to enable encryption on a DS8000 by using the GUI interface is simple. It takes approximately 5 minutes to complete all these steps. Now, you are ready for logical configuration, that is, to create ranks, extent pools, and volumes. From the moment you create the extent pools, you cannot disable the encryption unless you delete all volumes, ranks, and extent pools.

There are a few options that are available to manage the encryption environment. You can rekey the data key and recovery key. For more information, see 5.1, “Rekeying the data key” on page 150 and 5.2, “Recovery key use and maintenance” on page 151.

Steps for IBM Security Key Lifecycle Manager (TLS)

The steps in the configuration wizard for encrypting the DS8000 with IBM Security Key Lifecycle Manager by using TLS are similar to the steps for encrypting with IBM Security Key Lifecycle Manager by using TCP. However, before configuring IBM Security Key Lifecycle Manager with TLS, make sure to upgrade the security level of it, as shown in 4.7.1, “Configuration steps for changing IBM Security Key Lifecycle Manager V2.6 to use TLS 1.2” on page 137.

Complete the following steps:

1. After the recovery key is created, you must log on to the DS8000 GUI as a user who has the Administrator role to enable the encryption and authorize the previously generated recovery key. From the DS8000 GUI Welcome window, click **Settings** and then **Security**, as shown in the Figure 4-86.

Figure 4-86 Navigate to the Encryption window

2. The encryption wizard is shown in Figure 4-87. This wizard is started only when you enable the encryption for the first time. Click **Enable Encryption** to continue.

Figure 4-87 Encryption wizard: Enable Encryption

3. The Welcome window opens with the basic information that is related to the prerequisites for the next steps, such as at least two key servers should be already configured and online (connected to the DS8000). See Figure 4-88. Click **Next** to continue.

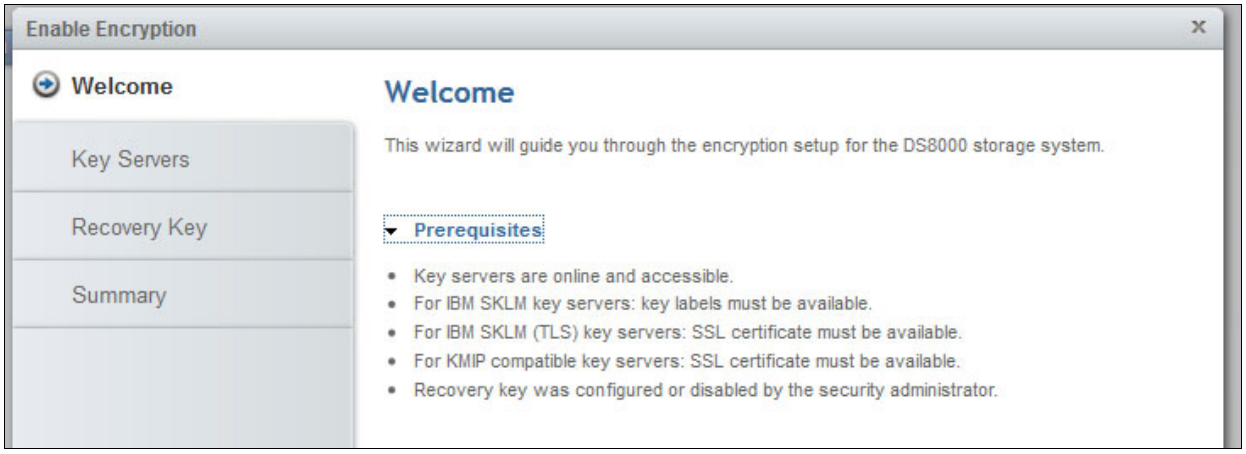


Figure 4-88 Encryption wizard: Welcome window

4. Select the Key Server Type. Select **IBM SKLM (TLS)** to use TLS port 441 to communicate with IBM Security Key Lifecycle Manager, as shown in Figure 4-89.

Figure 4-89 Encryption wizard: Key Server Type

5. Configure the key servers. The DS8000 supports up to four key servers. The following considerations apply to configurations:
   - In multiple site configurations, at least two of the key server ports should be assigned to isolated key servers at separate physical sites. The remaining ports can be connected to general key servers.
   - In single-site configurations, at least two of the key server ports should be assigned to isolated key servers at the same site.

   The DS8000 configuration for encryption also requires that at least two active key servers be connected and defined at the DS8000 instance.

   The DS8000 monitors all configured key servers. Client notification is provided for loss of access to key servers and other key server-related errors through DS8000 client notification mechanisms (SNMP traps and email, if configured) in the following ways:
   - Loss of access to key servers is reported at 5-minute intervals.
   - Loss of the ability for at least two key servers to provide key services that can prevent access to the data on the DS8000 is reported at 8-hour intervals.
   - The inability of any one key server to provide key services that can prevent access to data on the DS8000 is also reported at 8-hour intervals.

In Figure 4-90, two key servers are defined. You can add or remove key servers by clicking the + or - sign next to each key server field. Specify the host address (IP address or full qualified host name of the key server). Click **Next**.

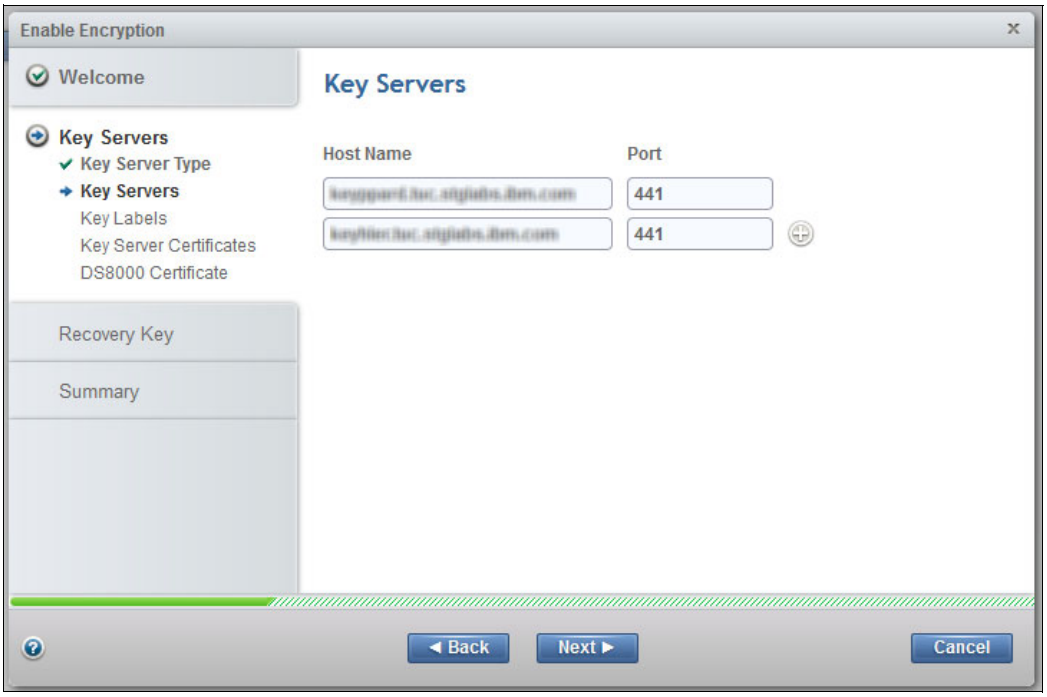


Figure 4-90 Encryption wizard: Define key servers

6. Each key server connection is tested. The message in Figure 4-91 is displayed if all the key servers that you defined in step 5 on page 110 are accessible. Click **OK**.

Note: The ports can also be changed and should match the setting on the IBM Security Key Lifecycle Manager key server. The default TLS port in 441.

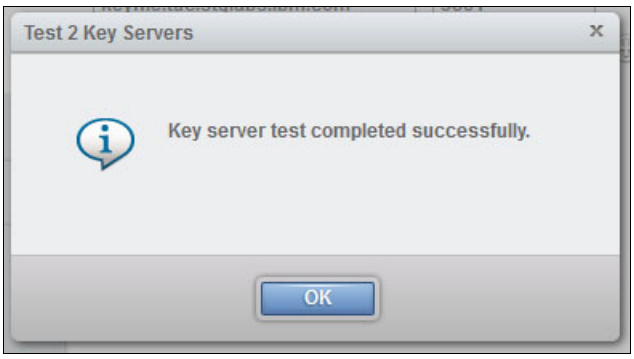


Figure 4-91 Encryption wizard: Test Key Servers

7. Define the key label for the data key that is generated by the IBM Security Key Lifecycle Manager server during the certificate creation step on the IBM Security Key Lifecycle Manager server. It is not required when using the KMIP protocol. This key label must match the one that is defined in 4.2.6, “Defining the DS8000 storage facility image to use with IBM Security Key Lifecycle Manager” on page 73. In this example, this key label is named `smoker1`.

Only one key label is required when the IBM Security Key Lifecycle Manager key servers are all installed on the open systems platforms with the same keystore type. A dual key label option is applicable only if at least one IBM Security Key Lifecycle Manager key server is installed on a z Systems server (z/OS) and the other on the open systems platform. This is due to the different keystore type that is used on z Systems servers.

In Figure 4-92, only one label is defined because all IBM Security Key Lifecycle Manager key servers are installed on the same platform with the same keystore type. Click the **+** sign to add a key label for the dual platform support. You can add the key label even after the encryption is enabled. The encryption wizard lets you to continue even though the key label that you provided does not match the key label that you specified in the IBM Security Key Lifecycle Manager server. The label verification is done as the last step of the encryption enablement process. Click **Next** to continue.

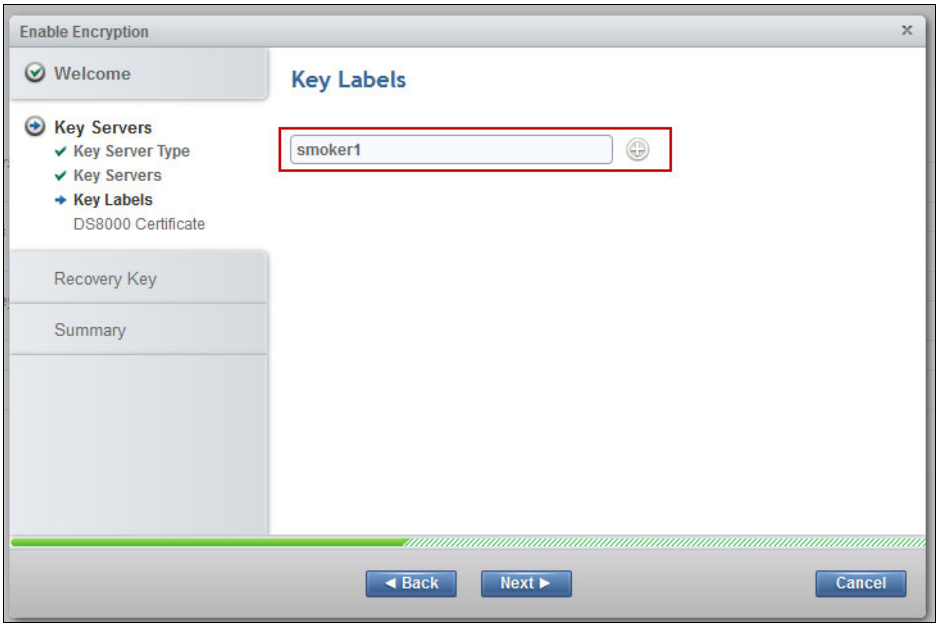


Figure 4-92 Encryption wizard: Define the key label

8. Transfer the SSL certificates from the key servers to the DS8000, as shown in Figure 4-93. Exporting the SSL certificates is shown in 4.7.1, “Configuration steps for changing IBM Security Key Lifecycle Manager V2.6 to use TLS 1.2” on page 137.

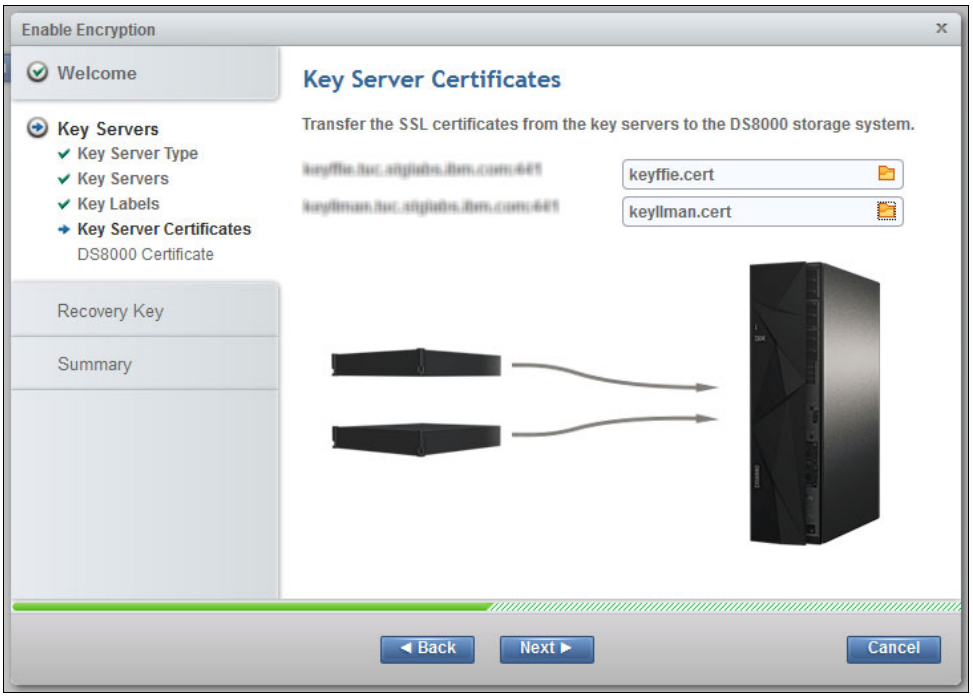


Figure 4-93 Transfer SSL certificates to DS8000

In Figure 4-94, the DS8000 public key is now transferred to the key server.

Figure 4-94 Key transferred

9. Authorize the pending request for the recovery key from the Security Administrator (if not already done). Click **Authorize**, as shown in the Figure 4-95.

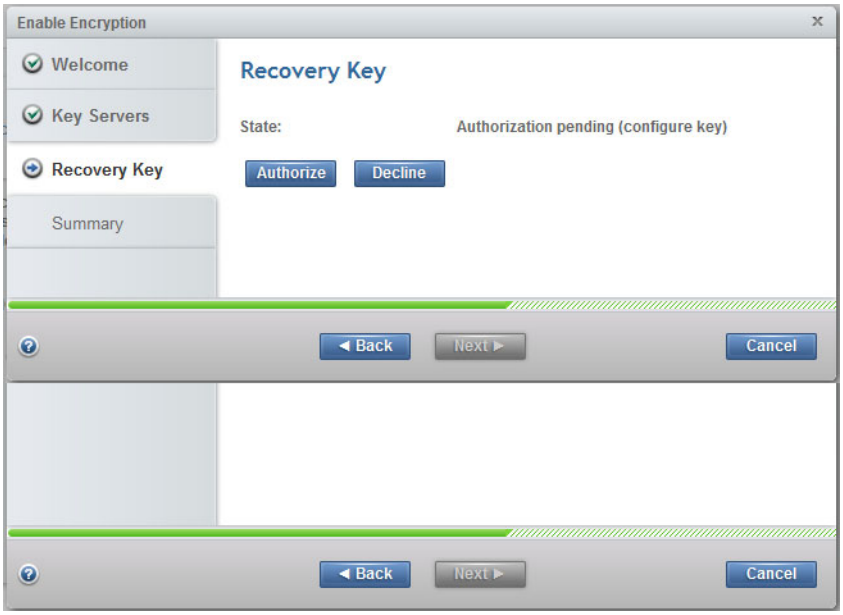


Figure 4-95 Encryption wizard: Authorize the recovery key

10. The confirmation message window opens (Figure 4-96). Click **Yes** to continue.

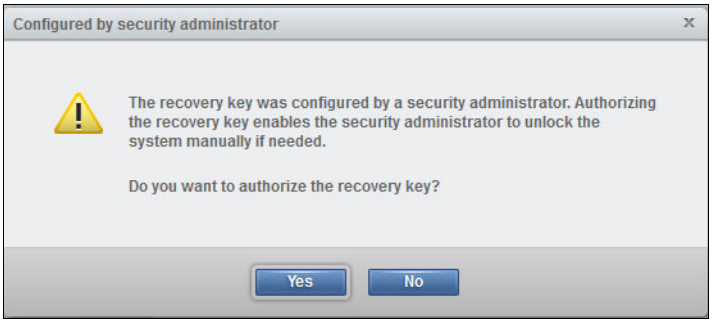


Figure 4-96 Encryption wizard: Confirm recovery key authorization

The recovery key state changes to Configured when the recovery key authorization is confirmed (see Figure 4-97). If the recovery key is disabled, the state shows as disabled.

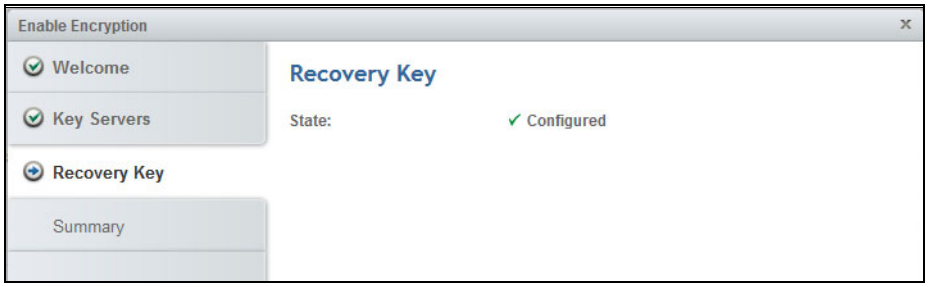


Figure 4-97 Encryption wizard: Recovery key configured

11. Figure 4-98 provides a summary of the configuration. Click **Finish** to initiate all the tasks that are required to enable the encryption.

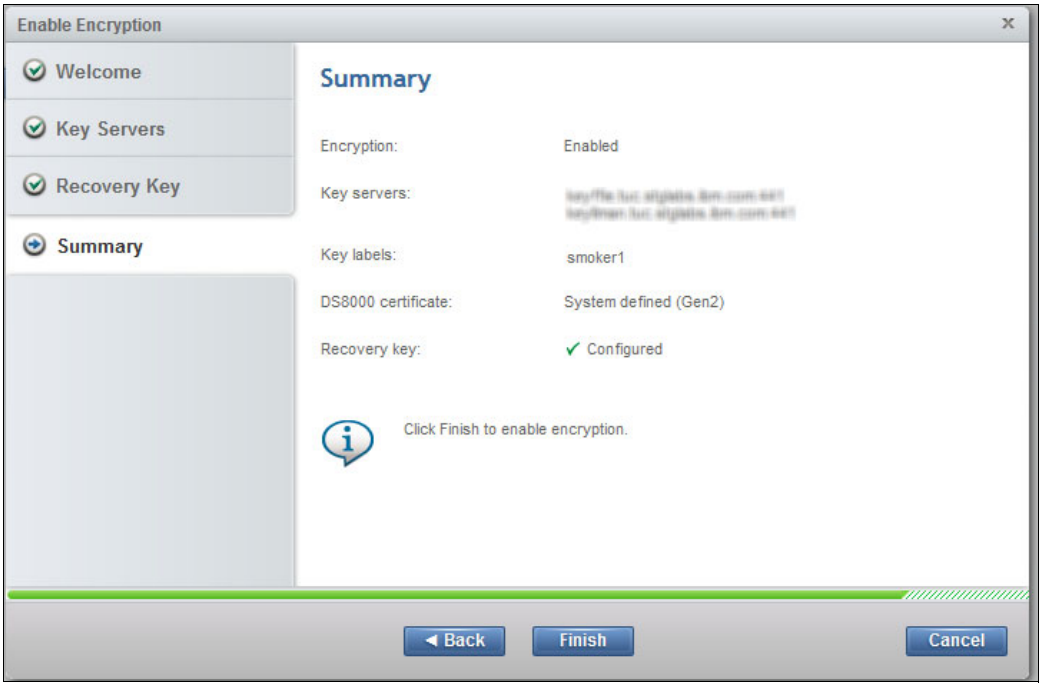


Figure 4-98 Encryption wizard: Summary

12. Encryption enablement tasks take approximately 1 minute. Expand the **View more details** section to see the task list. The overall progress is displayed as a percentage. When the Completed message displays, click **Close** (see Figure 4-99).

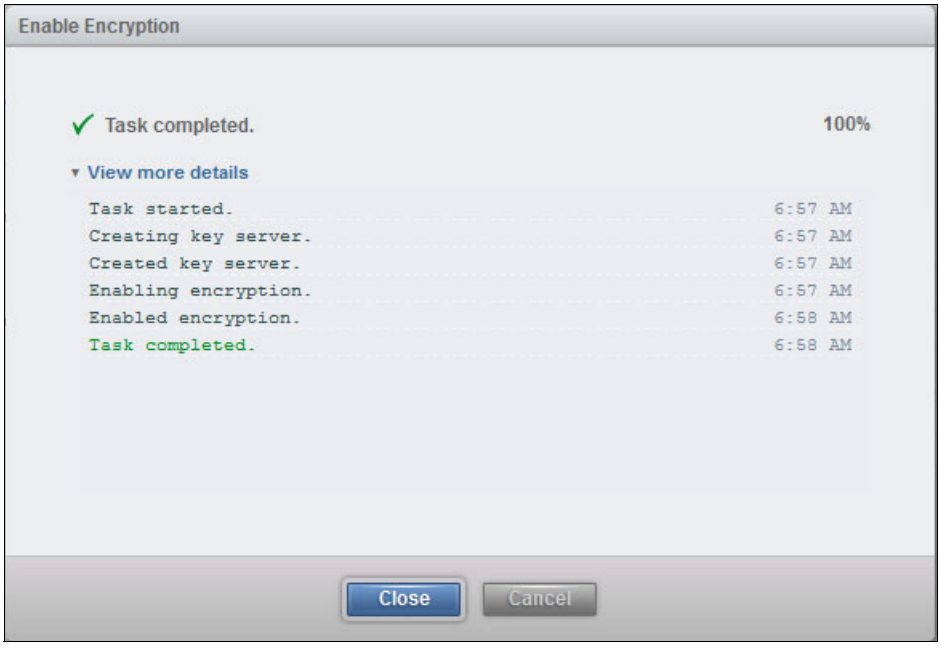


Figure 4-99 Encryption wizard: Encryption enabled

In the Encryption window (Figure 4-100), the encryption state is Enabled and the encryption key is Accessible. By expanding each section, you get more information. In this example, there is one key label and, in this case, four key servers that are accessible and online.

Figure 4-100 Encryption enabled and accessible

The overall process to enable encryption on a DS8000 by using the GUI interface is simple. It takes approximately 5 minutes to complete all these steps. Now, you are ready for logical configuration, that is, create ranks, extent pools, and volumes. From the moment you create the extent pools, you cannot disable the encryption unless you delete all the volumes, ranks, and extent pools.

There are a few options that are available to manage the encryption environment. You can rekey the data key and recovery key. For more information, see 5.1, “Rekeying the data key” on page 150 and 5.2, “Recovery key use and maintenance” on page 151.

Steps for Gemalto SafeNet Key Secure with KMIP

The steps in the configuration wizard for encrypting the DS8000 with Gemalto SafeNet KeySecure are similar to the steps that are used to encrypt the DS8000 with IBM Security Key Lifecycle Manager.

Complete the following steps:

1. After the recovery key is created, log on to the DS8000 GUI as a user who has the Administrator role to enable the encryption and authorize the previously generated recovery key. From the DS8000 GUI Welcome window, click **Settings** and then **Security**, as shown in the Figure 4-101.

Figure 4-101 Navigate to the Encryption window

The Encryption wizard is started only when you enable the encryption for the first time. Click **Enable Encryption** to continue as shown in Figure 4-102 on page 117.

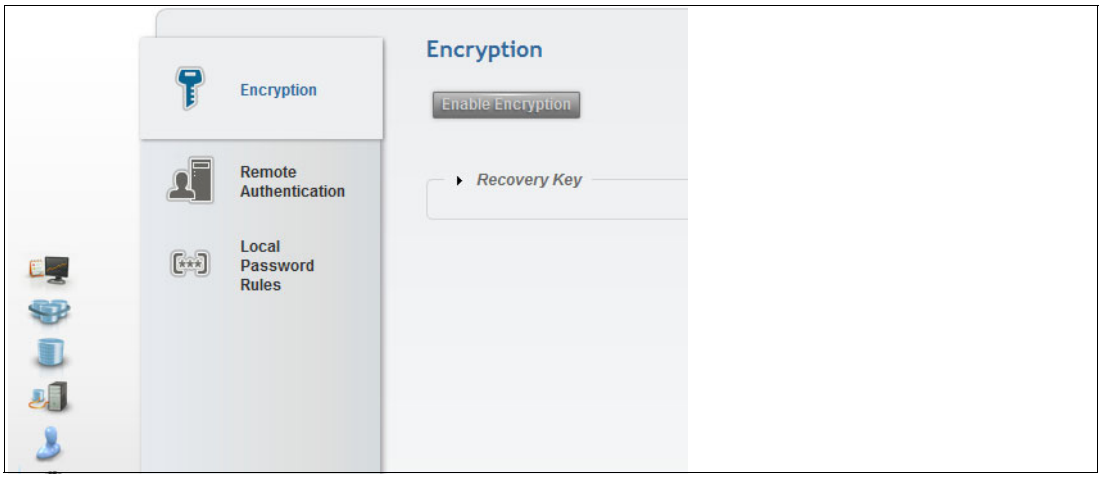


Figure 4-102 Enable Encryption

2. The Welcome window opens with the basic information that is related to the prerequisites for the next steps, such as at least two key servers should be already configured and online (connected to the DS8000). See Figure 4-103. Click **Next** to continue.

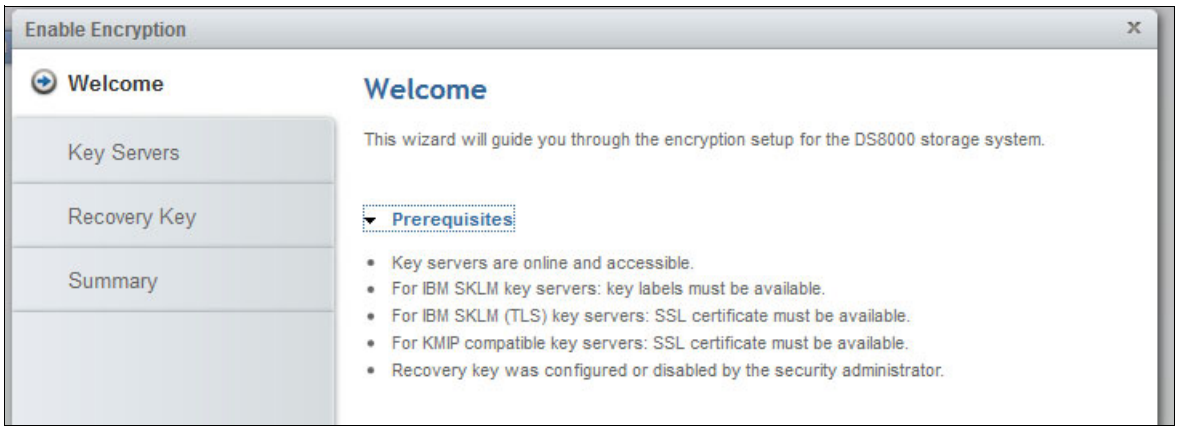


Figure 4-103 Encryption wizard: Welcome window

3. As shown in Figure 4-104, select the Key Server Type. Select **KMIP Compatible (TLS)** to use TLS port 5696 with KMIP to communicate with Gemalto SafeNet KeySecure.

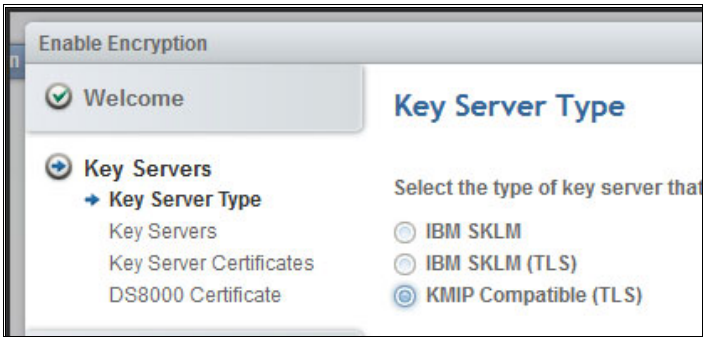


Figure 4-104 Key Server Type

4. The next step is to configure the key servers. The DS8000 supports up to four key servers. The following considerations apply to configurations:
   - In multiple site configurations, at least two of the key server ports should be assigned to isolated key servers at separate physical sites. The remaining ports can be connected to general key servers.
   - In single-site configurations, at least two of the key server ports should be assigned to isolated key servers at the same site.

   The DS8000 configuration for encryption also requires that at least two active key servers be connected and defined at the DS8000 instance.

The DS8000 monitors all the configured key servers. Client notification is provided for loss of access to key servers and other key server-related errors through DS8000 client notification mechanisms (SNMP traps and email, if configured) in the following ways:

- Loss of access to key servers is reported at 5-minute intervals.
- Loss of the ability for at least two key servers to provide key services that can prevent access to the data on the DS8000 is reported at 8-hour intervals.
- The inability of any one key server to provide key services that can prevent access to data on the DS8000 is also reported at 8-hour intervals.

In the example (Figure 4-105), two key servers are defined. You can add or remove key servers by clicking the + or - sign next to each key server field. Specify the host address (IP address or the full qualified host name of the key server). Click **Next**.

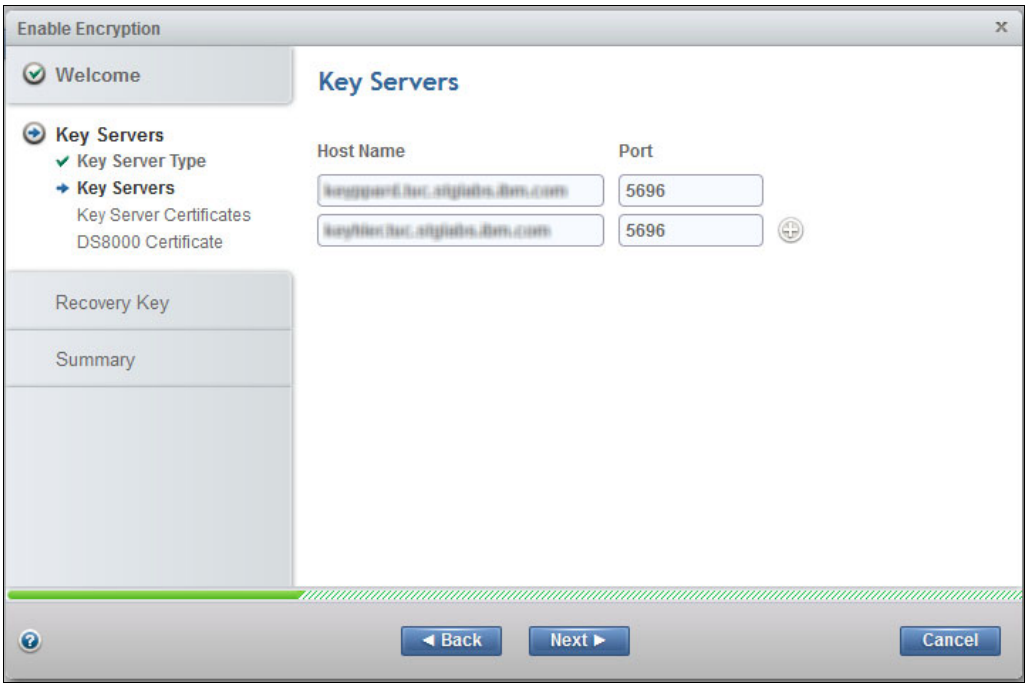


Figure 4-105 Encryption wizard: Define key servers

5. Each key server connection is tested. The message in Figure 4-107 is displayed if all the key servers that you defined in step 4 on page 118 are accessible. Click **OK**.

Note: The ports can also be changed and they should match the setting on the Gemalto SafeNet KeySecure key server. The default KMIP port is 5696.

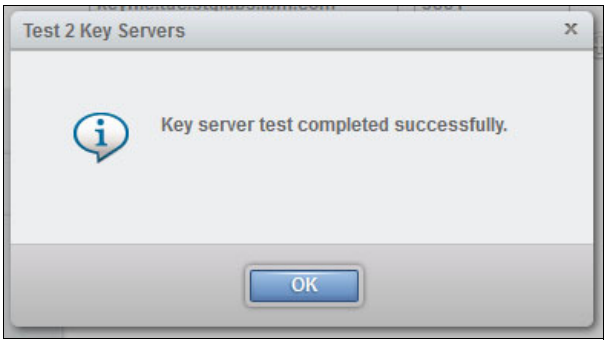


Figure 4-106 Encryption wizard: Test key servers

6. Now, transfer the SSL certificates from the key servers to the DS8000 that were exported during the Gemalto SafeNet KeySecure setup. Make sure to assign the correct SSL certificate to each server because they have the same filename. See Figure 4-107.

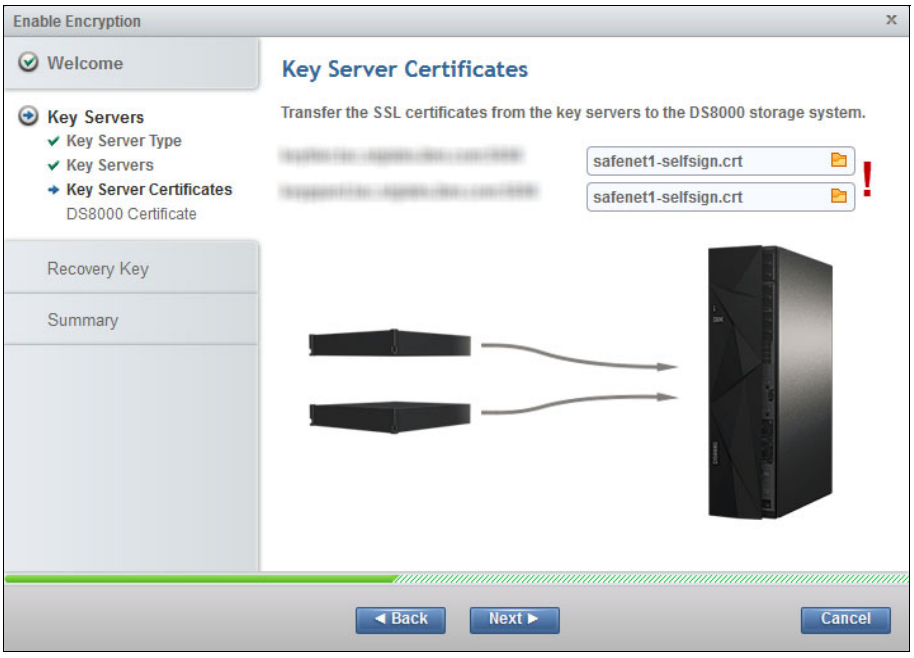


Figure 4-107 Transfer the SSL server certificate

The DS8000 public key is transferred to the key server, as shown in Figure 4-108.

Figure 4-108 Key transferred

7. Authorize the pending request for the recovery key from the Security Administrator (if not already done). Click **Authorize**, as shown in Figure 4-109.

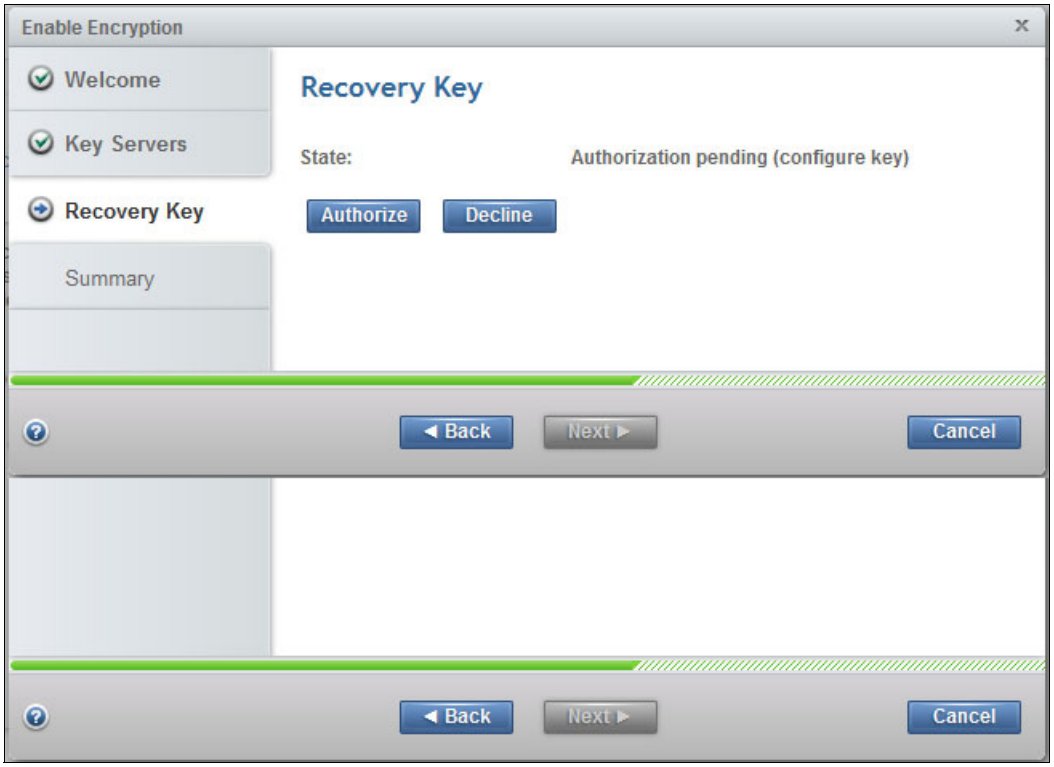


Figure 4-109 Encryption wizard: Authorize the recovery key

8. The confirmation message window opens (see Figure 4-110). Click **Yes** to continue.

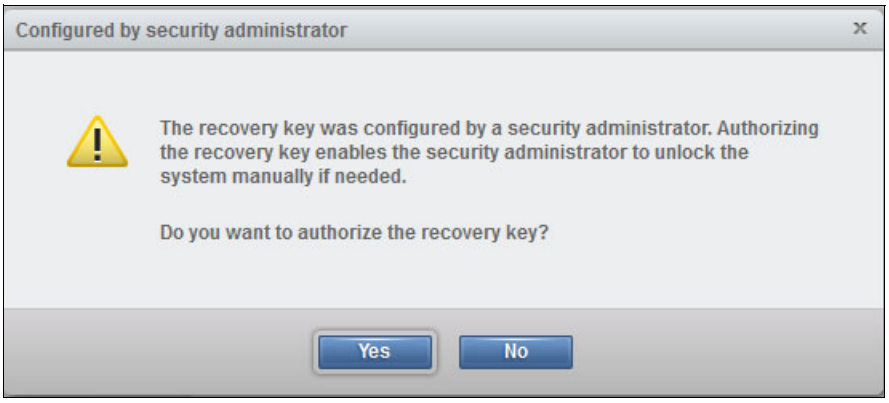


Figure 4-110 Encryption wizard: Confirm the recovery key authorization

The recovery key state changes to Configured when the recovery key authorization is confirmed (see Figure 4-111).

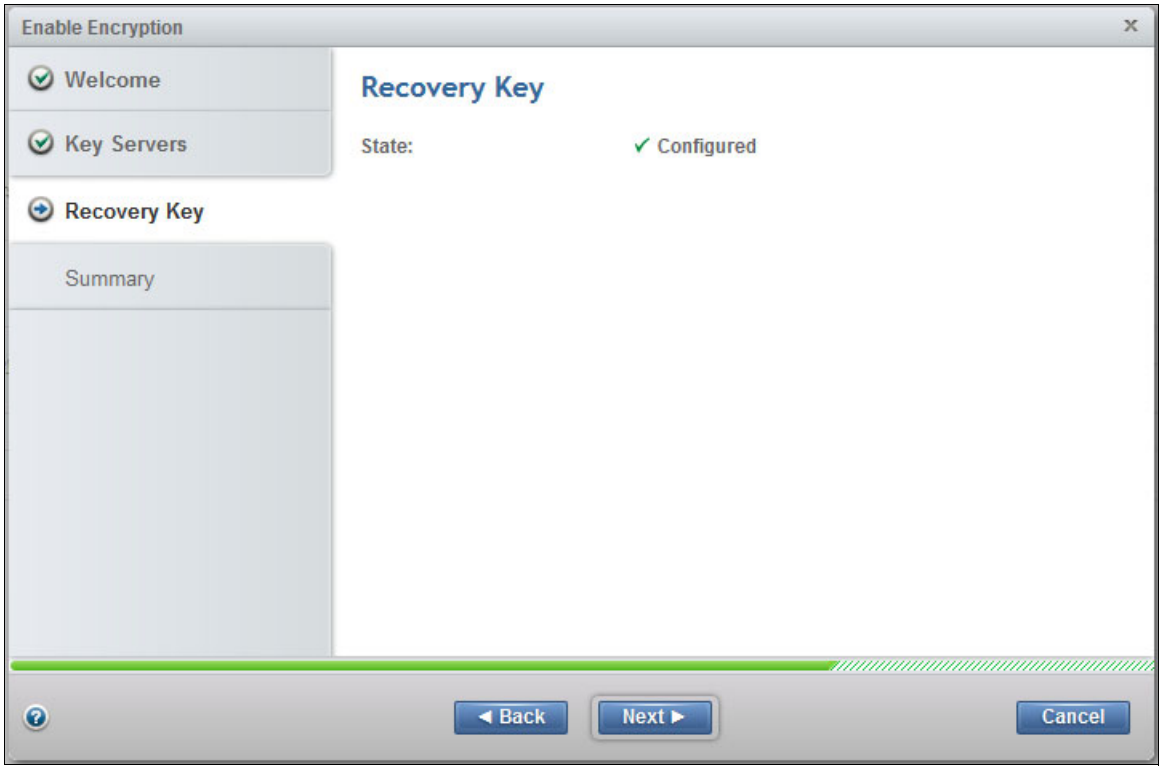


Figure 4-111 Encryption wizard: Recovery key configured

9. Figure 4-112 provides a summary of the configuration. Click **Finish** to initiate all the tasks that are required to enable encryption.

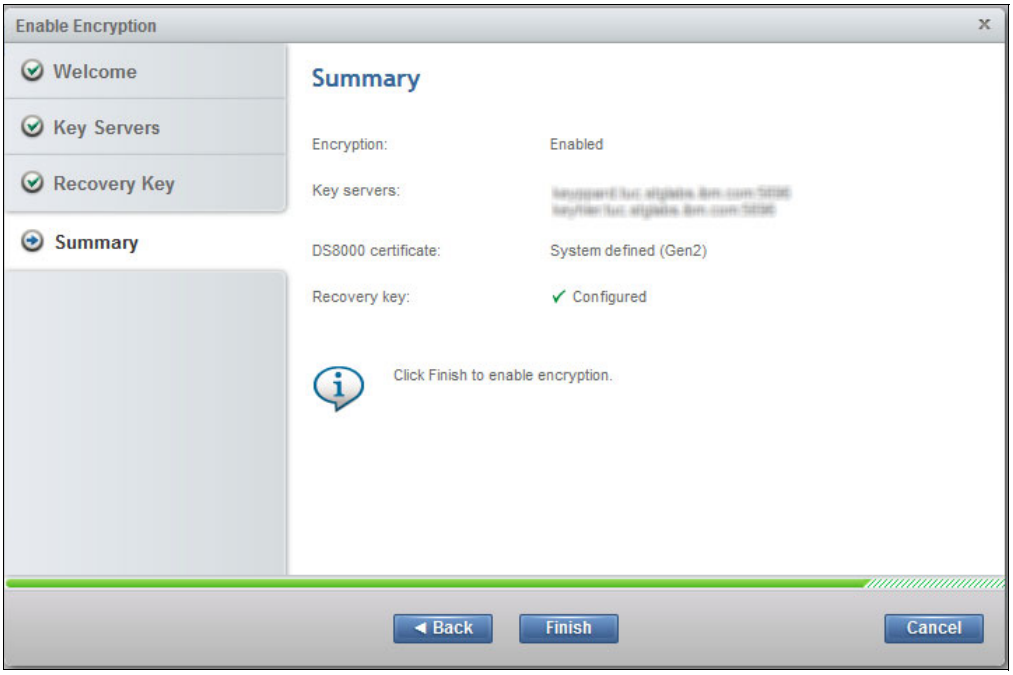


Figure 4-112 Encryption wizard: Summary

10. Encryption enablement tasks take approximately 1 minute. Expand the **View more details** section to see the task list. The overall progress is displayed as a percentage. When the Completed message displays, click **Close** (see Figure 4-113).

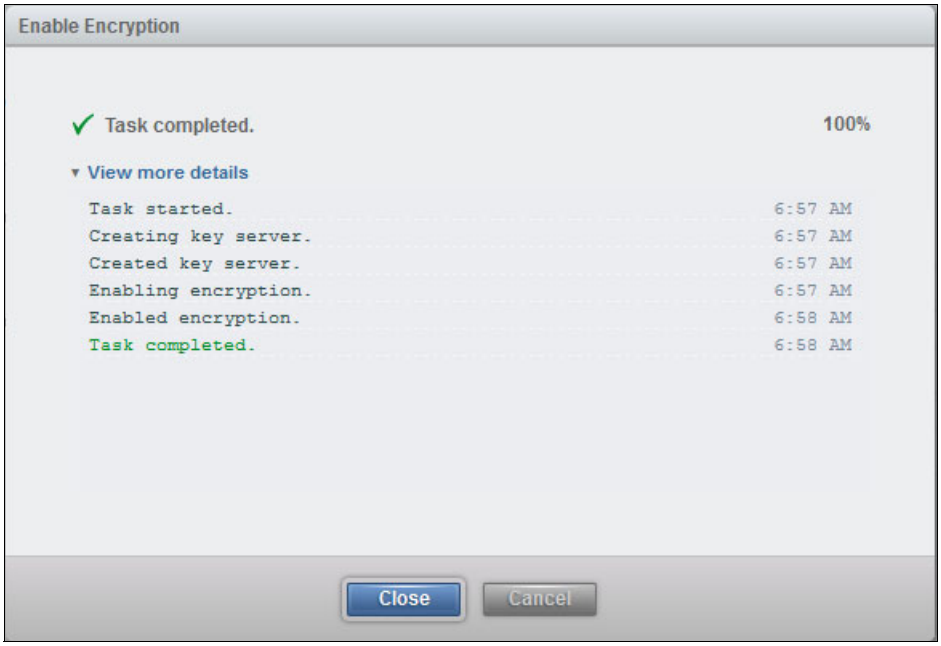


Figure 4-113 Encryption wizard: Encryption enabled

In the Encryption window (Figure 4-114), the encryption state is Enabled and the encryption key is Accessible. By expanding each section, you get more information. In this example, there are two KMIP compatible Gemalto SafeNet Key Servers that are configured and the UUID was generated.

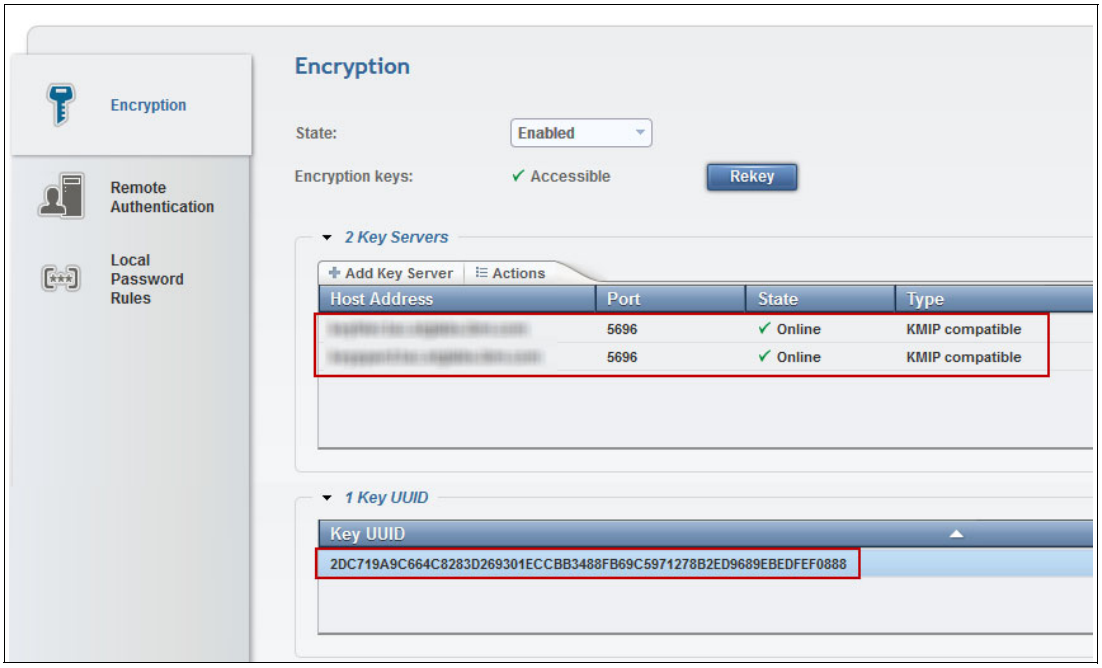


Figure 4-114 Encryption enabled and accessible

The overall process to enable the encryption on DS8000 by using the GUI interface is simple. It takes approximately 5 minutes to complete all the above documented steps. Now, you are ready for logical configuration, that is, create ranks, extent pools, and volumes. From the moment you create the extent pools, you cannot disable the encryption unless you delete all the volumes, ranks, and extent pools.

There are a few options that are available to manage the encryption environment. You can rekey the data key and recovery key. For more information, see 5.1, “Rekeying the data key” on page 150 and 5.2, “Recovery key use and maintenance” on page 151.

4.4.5 Configuring and administering encrypted arrays, ranks, and extent pools

When you enable encryption, you can start creating the extent pools. You do not need to specify or enable any additional parameters to start using the encrypted DS8000 disks. When you select arrays, the encryption status for each array is displayed. As shown in Figure 4-115, selected arrays are encrypted.

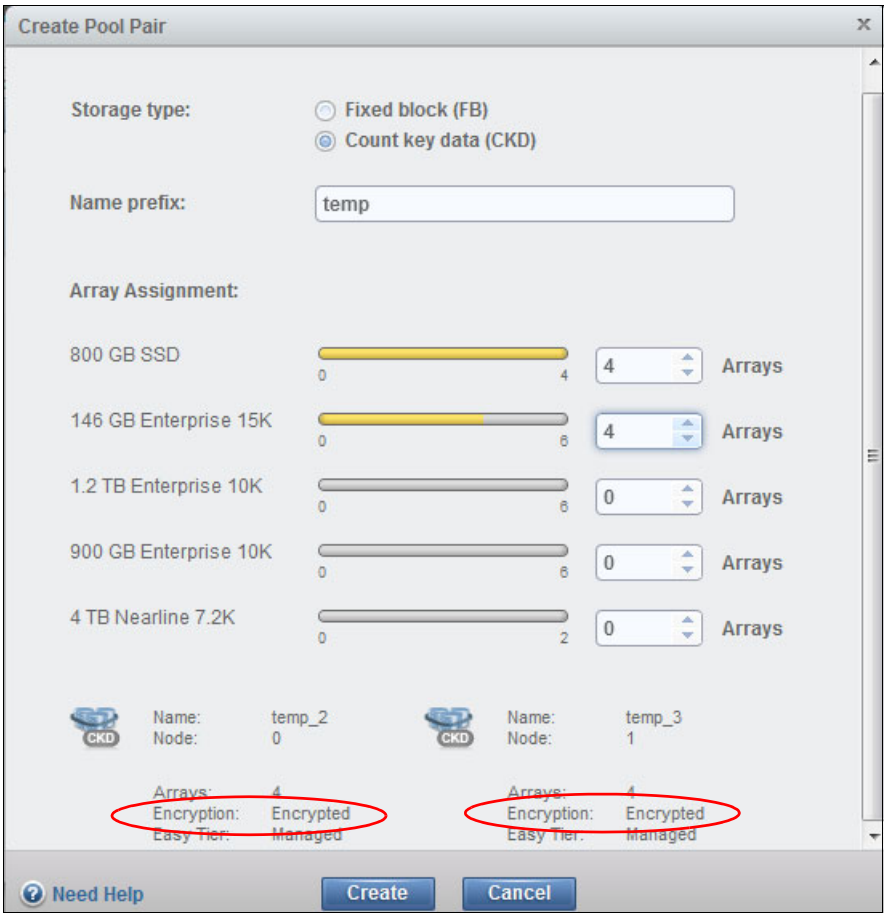


Figure 4-115 Create extent pools

For more information about how to configure DS8000 logically, see *ACI Worldwide's BASE24-eps V6.2: A Supplement to SG24-7268*, REDP-4338.

4.5 Command-line configuration for DS8000 encryption

You can configure disk encryption on the DS8000 by using the DS command-line interface (DS CLI). The high-level configuration sequence is as follows:

1. Configure IBM Security Key Lifecycle Manager or Gemalto SafeNet KeySecure server connection to the DS8000.
2. Configure the recovery key.
3. Configure the encryption group.
4. Apply the activation key.
5. Configure and administer the encrypted arrays.

6. Configure and administer the encrypted ranks.
7. Configure and administer the encrypted extent pools.

For information about enabling NIST SP 800-131a compliant encryption certificates and TLS 1.2 communication, see 4.7, “NIST SP 800-131a requirements for key servers” on page 136.

Details: For more information about the CLI and commands, see *IBM DS8000 Series Command-Line Interface User's Guide*, SC27-8526.

4.5.1 Configuring the key server connection

The DS8000 supports up to four IBM Security Key Lifecycle Manager key servers or four Gemalto SafeNet KeySecure connections. Intermixing is not allowed.

The following suggestions apply to configurations:

- In multiple site configurations, at least two of the key server ports should be assigned to isolated key servers at separate physical sites. The remaining ports can be connected to general key servers.
- In single-site configurations, at least two of the key server ports should be assigned to isolated key servers at the same site.

The DS8000 configuration for encryption also requires that at least two active key servers be connected and defined at the DS8000 installation. The following procedure describes how to connect the key servers by using TCP port 3801 for IBM Security Key Lifecycle Manager, SSL/TLS port 441 for IBM Security Key Lifecycle Manager, and KMIP port 5697 for Gemalto SafeNet KeySecure.

If you must connect the IBM Security Key Lifecycle Manager by using SSL/TLS v1.2 for NIST SP 800-131a compliance, see 4.7, “NIST SP 800-131a requirements for key servers” on page 136, then come back and continue with “Configuring IBM Security Key Lifecycle Manager with TLS/SSL” on page 127.

Configuring IBM Security Key Lifecycle Manager with IPP

To configure the IBM Security Key Lifecycle Manager server connection, use the **lskeymgr** and **mkkeymgr** commands in the following steps:

1. Run the **lskeymgr** command to view the list of registered IBM Security Key Lifecycle Manager servers, if any.

   Run the **lskeymgr** command at the **dscli** command prompt with the parameters and variables that are shown in Example 4-14. In this example, one IBM Security Key Lifecycle Manager server, named `popcorn`, is already registered with the DS8000.

   Example 4-14 List existing IBM Security Key Lifecycle Manager servers

```
dscli> lskeymgr -l
-------------------------------------------
ID  state   status keyprotocol addr      port
===========================================
1   active  normal IPP         popcorn   3801
```

2. Run the **mkkeymgr** command to add an IBM Security Key Lifecycle Manager server.

   Run the **mkkeymgr** command at the **dscli** command prompt, with the parameters and variables that are shown in Example 4-15. In this example, a second IBM Security Key Lifecycle Manager server, named `peanuts`, is added to the DS8000 configuration.

   The introduction of KMIP requires you to specify the protocol being that is used during the key server connection. It is IPP for IBM Security Key Lifecycle Manager.

   Example 4-15 Add new IBM Security Key Lifecycle Manager server

```
dscli> mkkeymgr -addr -keyprotocol IPP peanuts 2
-------------------------------------------------------
CMUC00354I mkkeymgr: The key server 2 has been created.
```

3. Verify that the new IBM Security Key Lifecycle Manager server was added successfully, the state is active, and the status is normal. Run the **lskeymgr** command again with the **-l** parameter, as shown in Example 4-16.

   Example 4-16 Verify the IBM Security Key Lifecycle Manager servers

```
dscli> lskeymgr -l
-----------------------------------------
ID  state   status keyprotocol addr      port
=========================================
1   active  normal IPP         popcorn   3801
2   active  normal IPP         peanuts   3801
```

4. To delete an IBM Security Key Lifecycle Manager server, you can run the **rmkeymgr** command with the parameters and variables that are shown in Example 4-17.

   Example 4-17 Remove the IBM Security Key Lifecycle Manager server from the configuration

```
dscli> rmkeymgr 2
-------------------------------------------------------------------------------
CMUC00356I rmkeymgr: Are you sure you want to delete the key server 2? [y/n]:y
CMUC00357I rmkeymgr: The key server 2 has been deleted.
```

Configuring IBM Security Key Lifecycle Manager with TLS/SSL

To configure the IBM Security Key Lifecycle Manager server connection, run the **lskeymgr** and **mkkeymgr** commands in the following steps:

1. Run the **lskeymgr** command to view the list of already registered IBM Security Key Lifecycle Manager servers, if any.

   Run the **lskeymgr** command at the **dscli** command prompt, with the parameters and variables that are shown in Example 4-18. In this example, one IBM Security Key Lifecycle Manager server, named `popcorn`, is already registered with the DS8000.

   Example 4-18 List the existing IBM Security Key Lifecycle Manager servers

```
dscli> lskeymgr -l
-----------------------------------------
ID  state   status keyprotocol addr      port
=========================================
1   active  normal IPP         popcorn   441
```

2. Run the **mkkeymgr** command to add an IBM Security Key Lifecycle Manager server.

   Run the **mkkeymgr** command at the **dscli** command prompt, with the parameters and variables that are shown in Example 4-19. In this example, a second IBM Security Key Lifecycle Manager server, named `peanuts`, with its SSL certificate `sklm6_ssl_cert.crt`, is added to the DS8000 configuration.

   The introduction of KMIP requires you to specify the protocol that is being used during the key server connection. It is IPP for IBM Security Key Lifecycle Manager.

   Example 4-19 Add an IBM Security Key Lifecycle Manager server

```
dscli> mkkeymgr -protocol IPP -port 441 -addr peanuts -cert
/tmp/sklm6_ssl_cert.crt 2
-------------------------------------------------------
CMUC00354I mkkeymgr: The key server 2 has been created.
```

3. Verify that the new IBM Security Key Lifecycle Manager server was added successfully, the state is active, and the status is normal. Run the **lskeymgr** command again with the **-l** parameter, as shown in Example 4-20.

   Example 4-20 Verify the IBM Security Key Lifecycle Manager servers

```
dscli> lskeymgr -l
-----------------------------------------
ID  state   status keyprotocol addr      port
=========================================
1   active  normal IPP         popcorn   441
2   active  normal IPP         peanuts   441
```

4. To delete an IBM Security Key Lifecycle Manager server, you can run the **rmkeymgr** command with the parameters and variables that are shown in Example 4-21.

   Example 4-21 Remove the IBM Security Key Lifecycle Manager server from the configuration

```
dscli> rmkeymgr 2
-------------------------------------------------------------------------------
CMUC00356I rmkeymgr: Are you sure you want to delete the key server 2? [y/n]:y
CMUC00357I rmkeymgr: The key server 2 has been deleted.
```

Configuring Gemalto SafeNet KeySecure with KMIP

To configure the Gemalto SafeNet KeySecure server connection, run the **lskeymgr** and **mkkeymgr** commands in the following steps:

1. Run the **lskeymgr** command to view the list of already registered key servers, if any.

   Run the **lskeymgr** command at the **dscli** command prompt, with the parameters and variables that are shown in Example 4-22. In this example, no key servers are registered with the DS8000.

   Example 4-22 List existing key servers

```
dscli> lskeymgr -l
----------------------------------------
CMUC00234I lskeymgr: No Key Manager found.
```

2. Run the **mkkeymgr** command to add a Gemalto SafeNet KeySecure server.

   Run the **mkkeymgr** command at the **dscli** command prompt, with the parameters and variables that are shown in Example 4-23. In this example, the first Gemalto SafeNet KeySecure server is added to the DS8000 configuration.

   Example 4-23 Add a Gemalto SafeNet KeySecure server

```
dscli> mkkeymgr -addr 9.155.000.23 -keyprotocol KMIP -cert c:\cert\mycert.pem 1
-------------------------------------------------------
CMUC00354I mkkeymgr: The key server 1 has been created.
```

   Note: The **-cert** parameter specifies the location of the certificate file (SafeNet SSL public key) that was exported during the Gemalto SafeNet KeySecure server setup to use as a trust anchor to authenticate the certificate of the specified key server when using a TLS security protocol. If the parameter is not specified, then only non-TLS protocols that do not require a trust anchor certificate are allowed. The certificate is in the PEM or DER format. The TLS security protocol is required when using KMIP.

3. Verify that the new Gemalto SafeNet KeySecure server was added successfully, the state is active, and the status is normal. Run the **lskeymgr** command again with the **-l** parameter, as shown in Example 4-24.

   Example 4-24 Verify the Gemalto SafeNet KeySecure server

```
dscli> lskeymgr -l
-------------------------------------------------
ID  state   status keyprotocol addr           port
=================================================
1   active  normal KMIP        9.155.000.23   5697
```

4. To delete a Gemalto SafeNet KeySecure server, you can run the **rmkeymgr** command with the parameters and variables that are shown in Example 4-25.

   Example 4-25 Remove the Gemalto SafeNet KeySecure server from the configuration

```
dscli> rmkeymgr 1
-------------------------------------------------------------------------------
CMUC00356I rmkeymgr: Are you sure you want to delete the key server 1? [y/n]:y
CMUC00357I rmkeymgr: The key server 1has been deleted.
```

4.5.2 Managing the recovery key

As described in 4.4.5, “Configuring and administering encrypted arrays, ranks, and extent pools” on page 125, the risk of a deadlock situation can be substantially minimized by maintaining redundant (dual-platform) IBM Security Key Lifecycle Manager servers, but it cannot be eliminated. The *recovery key* feature provides a way to get out of a deadlock state. Starting with Licensed Machine Code (LMC) Level 6.5.1.xx, users can enable or disable recovery key management.

This choice must be made before you configure the encryption group.

Enabling the recovery key

To configure the recovery key by using CLI commands, complete the following steps:

1. The CLI command that is used to configure the recovery key must be entered by a user with Security Administrator (secadmin) authority. Run the **mkreckey** command, as shown in Example 4-26.

   Example 4-26 Configure the recovery key with the mkreckey command

   ```
   dscli> mkreckey –dev IBM.2107-1300861 1
   ----------------------------------------------------------------------------------
   CMUC00392I mkreckey: The access Recovery Key
   0123-4569-4443-3334-3334-0123-4569-3334-4443-3334-3334-0123-4569-4443-3334-3334 for
   encryption group 1 has been created, pending verification.
   ```

   You can copy the new recovery key text from the terminal and save it in a file, which can be used for printing. However, this is not preferable because the key can be discovered by a network sniffer. A better approach is to write the key on a piece of paper.

   The Security Administrator is responsible for writing the recovery key and storing the paper in a safe place.

2. The Security Administrator runs the **managereckey -action verify** command to ensure that the previously written key is correct, as shown in Example 4-27.

   Example 4-27 Verify the recovery key

   ```
   dscli> managereckey –dev IBM.2107-1300861 –action verify - key
   0123-4569-4443-3334-3334-0123-4569-3334-4443-3334-3334-0123-4569-4443-3334-3334 1
   ----------------------------------------------------------------------------------
   CMUC00393I managereckey: The access Recovery Key for encryption group 1 has been
   verified, pending authorization.
   ```

   The recovery key is now in the Authorization Pending status.

3. After the recovery key is verified, the Storage Administrator authorizes use of the recovery key that was created. You must log on as a user with Storage Administrator authority to run the **managereckey -action authorize** command, as shown in Example 4-28.

   Example 4-28 Authorize the recovery key

   ```
   dscli> managereckey –dev IBM.2107-1300861 –action authorize 1
   CMUC00406W managereckey: Are you sure that you want to authorize the creation of the
   access Recovery Key for encryption group 1? [Y/N]:Y
   ----------------------------------------------------------------------------------
   CMUC00395I managereckey: The pending Recovery Key management operation for encryption
   group 1 has been authorized.
   ```

4. Check the recovery key state by running the **lskeygrp** command, as shown in Example 4-29.

   Example 4-29 List the recovery key and its status

   ```
   dscli> lskeygrp -l
   ------------------------------------------------------
   ID  state        reckeystate reckeycreated label label2
   ======================================================
   1   unconfigured configured  10/10/2009    -     -
   ```

   The `reckeystate` is `configured`, indicating that a new recovery key was requested, verified, and authorized. `State` is still unconfigured because the encryption group is not configured yet.

Now, you can proceed with the configuration of the encryption group, as described in 4.5.3, “Configuring and administering the encryption group” on page 131. However, you can disable the recovery key instead, as described in , “Disabling the recovery key” on page 131.

Disabling the recovery key

To disable recovery key management by using CLI commands, complete the following steps:

1. The DS CLI command that is used to disable the recovery key must be entered by a user with Security Administrator authority. Run the **managereckey** command as shown in the Example 4-30.

Example 4-30 Disable recovery key: The managereckey command

```
dscli> managereckey -action disable 1
-------------------------------------------------------------------------------
CMUC00416I managereckey: The access recovery key for encryption group 1 has been
disabled, pending authorization.
```

The disable action is now in the Authorization Pending status.

2. The Storage Administrator authorizes the disabling of the recovery key. Log on as a user with Storage Administrator authority to run the **managereckey -action authorize** command, as shown in Example 4-31.

Example 4-31 Disable recovery key: The managereckey command

```
dscli> managereckey -action authorize 1
-------------------------------------------------------------------------------
CMUC00418W managereckey: Are you sure that you want to authorize the disable of the
access recovery key for encryption group 1? [Y/N]: y
CMUC00395I managereckey: The pending recovery key management operation for encryption
group 1 has been authorized.
```

3. Check the recovery key state by running the **lskeygrp** command, as shown in Example 4-32.

Example 4-32 List the recovery key and its status

```
dscli> lskeygrp
-----------------------------------------------------
ID  state        reckeystate reckeydate datakeydate keyprotocol
=====================================================================
1   unconfigured disabled    -          05/18/2010  IPP / KMIP
```

The `reckeystate` is `disabled`. `State` is still unconfigured because the encryption group is not configured yet.

4.5.3 Configuring and administering the encryption group

The client data within the storage facility image that is encrypted is contained in a single encryption group. The encryption group that contains encrypted data is enabled to access data through one data key that is obtained from a key server.

Note: Currently, the DS8000 supports only one encryption group. It must either be configured for IPP or KMIP.

An *encryption group* contains a set of *extent pools*, each of which has a set of associated ranks and volumes. The IBM FlashCopy® and remote mirror and copy functions can migrate data within or between encryption groups.

To configure the encryption key group by using DS CLI commands, complete the following steps:

1. Run the **lskeygrp** command to view the state of encryption group and recovery key. Run the **lskeygrp** command at the **dscli** command prompt with the parameters and variables that are shown in Example 4-33. In this example, the recovery key is configured and the encryption group is unconfigured.

   Example 4-33 List key groups

```
dscli> lskeygrp -l
-------------------------------------------------------
ID  state        reckeystate reckeycreated label label2
=======================================================
1   unconfigured configured  10/10/2009    -     -
```

2. Run the **mkkeygrp** command to configure the new encryption key group. Enter the **mkkeygrp** command at the **dscli** command prompt with the parameters and variables that are shown in Example 4-34. The **-label** parameter is required for IPP. It matches the one, that you defined on the IBM Security Key Lifecycle Manager server (see Figure 4-4 on page 62). Number 1 is the encryption group ID. The syntax for KMIP is slightly different, as shown in Example 4-35.

   Example 4-34 Create IPP key group: IPP

```
dscli> mkkeygrp -label itsokey -keyprotocol IPP 1
-------------------------------------------------------
The key server encryption key group 1 has been created.
```

   Example 4-35 Create KMIP key group: KMIP

```
dscli> mkkeygrp -keyprotocol KMIP 1
-------------------------------------------------
The encryption group entry 1 created successfully.
```

3. Run the **lskeygrp** command with the **-l** parameter, as shown in Example 4-36, to verify that the encryption group was added successfully and that its state is accessible.

   Example 4-36 List defined key groups

```
dscli> lskeygrp -l
-----------------------------------------------------------
ID  state      reckeystate reckeydate datakeydate keyprotocol
===========================================================
1   accessible configured  04/29/2016 05/04/2016  IPP or KMIP
```

4.5.4 Applying the encryption activation key

Now that the IBM Security Key Lifecycle Manager key servers are set up and registered with the DS8000, the encryption feature activation key must be applied. You must do this task before the logical configuration (defining ranks and extent pools).

The **applykey** command activates the licenses for your storage unit. The **lskey** command verifies which type of licensed features are activated for your storage unit.

Complete the following steps:

1. To apply the encryption activation key, run the following command:

   ```
   dscli> applykey -file c:\keys.xml -dev IBM.2107-75LY981
   ```

   This example is based on the assumption that your XML file is named `keys.xml` and it is in the root directory of your `C:` drive.

2. Run the **lskey** command to verify that the license codes are applied, as shown in Example 4-37.

Example 4-37 List all activated licenses

```
dscli> lskey IBM.2107-75LY981
------------------------------------------------------------------------
Activation Key                              Authorization Level (TB) Scope
========================================================================
Encryption Authorization                    on                       All
Global mirror (GM)                          80.3                     CKD
High Performance FICON for z Systems (zHPF) on                        CKD
IBM FlashCopy SE                            80.3                     CKD
IBM HyperPAV                                on                       CKD
IBM database protection                     on                       FB
Metro mirror (MM)                           80.3                     CKD
Metro/Global mirror (MGM)                   80.3                     CKD
Operating environment (OEL)                 80.3                     All
Parallel access volumes (PAV)               80.3                     CKD
Point in time copy (PTC)                    80.3                     CKD
RMZ Resync                                  80.3                     CKD
Remote mirror for z/OS (RMZ)                80.3                     CKD
```

4.5.5 Creating encrypted arrays

To create encrypted arrays by using the DS CLI commands, use the **lsarraysite** and **mkarray** commands. An array inherits the characteristics of its parent array sites and the RAID type attribute (5, 6, or 10). A DS8000 array of RAID type 5, 6, or 10 is made from one (eight disk drive modules (DDMs)) array site. The status of the array is *unassigned* until the array is assigned to a rank.

To create an encrypted array from unassigned array sites, complete the following steps:

1. Run the **lsarraysite** command to view a list of array site IDs for all installed array sites. Review those arrays that are designated with the state of unassigned.

   Run the **lsarraysite** command at the **dscli** command prompt with the following parameters and variables:

   ```
   dscli> lsarraysite -dev IBM.2107-75LY981 -state unassigned -l
   ```

2. Press Enter to get a report of unassigned array sites (see Example 4-38). Use the list to identify unassigned array site capacity, rpm, and device adapter (DA) pair attributes. Record the RAID type for each array site.

Example 4-38 List array sites

```
dscli> lsarraysite -dev IBM.2107-75LY981 -state unassigned -l
-------------------------------------------------------------------------------
arsite DA Pair dkcap (10^9B) diskrpm State      Array diskclass encrypt
===============================================================================
S8     0               450.0   15000 Unassigned -     ENT       supported
```

Unassigned state: Make sure that the array site supports encryption. If this is your first time creating fixed block volumes, all of the arrays are displayed with a state of `unassigned`.

3. Run the **mkarray** command to create an array from one array site with the status of `unassigned`. Run the **mkarray** command at the **dscli** command prompt with the following parameters and variables:

   ```
   dscli>mkarray -dev storage_image_ID -raidtype 6 -arsite array_site
   ```

   Consider the following information when you create the arrays:

   – Specify one array site with identical capacity, rpm, interface, and DA pair attributes.
   – The new array inherits the capacity, rpm, interface, and DA pair characteristics of its parent array site.
   – The state of the array remains `unassigned` until it is assigned to a rank.

4. Verify that the array-to-array site assignment is recognized and complete by running either the **lsarray** or **lsarraysite** command with the **-l** parameter. Example 4-39 gives an illustration.

Example 4-39 Verify the arrays

```
dscli> lsarray -dev IBM.2107-75LY981 -l
-------------------------------------------------------------------------------
Array State      Data   RAIDtype    arsite Rank DA Pair DDMcap (10^9B)
diskclass encrypt
===============================================================================
A0    Assigned   Normal 5 ( 6+P+S )   S8 R1   0 450.0 ENT supported
```

4.5.6 Creating encrypted ranks

RUn the **lsarray**, **mkrank**, and **lsrank** commands to assign a rank to each unassigned array.

To create ranks, complete the following steps:

1. Ensure that you have a list of the unassigned arrays for which ranks must be assigned. Run the **lsarray** command to get this list if you do not already have it. Enter the **lsarray** command at the **dscli** command prompt with the following parameters and variables:

   ```
   dscli>lsarray -dev IBM.2107-75LY981 -state unassigned
   ```

2. Run the **mkrank** command to assign a rank to rank group 0 or 1 according to the rank group number of the assigned extent pool ID.

   To create an encrypted rank, use the **-encryptgrp** variable.

Run the **mkrank** command at the **dscli** command prompt with the parameters and variables that are shown in Example 4-40 for fixed block storage type.

Example 4-40 Assign an array to a rank

```
dscli>mkrank -dev IBM.2107-75LY981 -array A1 -stgtype fb -wait -encryptgrp 1
-------------------------------------------------------------------------
Rank IBM.2107-75HT551/R0 successfully created.
```

Notes:

- The **-encryptgrp encryption_group_ID** parameter specifies the encryption group that this rank should use. The default is 0 (zero), which means that no encryption group is assigned to the rank.
- You can specify either the **-wait** or the **-extpool** parameter when you use the **mkrank** command. Either of these parameters allows you to be notified if the rank configuration fails for any reason.
- If you use the **-wait** parameter, you cannot run other commands until the entire transaction is processed.

3. Press Enter to display a report of rank assignments for your entire storage unit. Because the process of creating the rank involves formatting drives, the process can take a while before it finishes. If you want to check on the process, run the **lsrank** command from a different DS CLI session.
4. Verify that ranks and extent pools are assigned by running the **lsrank** command at the **dscli** command prompt. Use the parameters and variables that are shown in Example 4-41.

Example 4-41 Verify the ranks

```
dscli> lsrank -dev IBM.2107-75LY981 -l
-------------------------------------------------------------------------------
ID Group State  datastate Array RAIDtype extpoolID extpoolnam stgtype exts usede
xts encryptgrp
===============================================================================
R0     0 Normal Normal    A1          10 P0        raid10P0   fb      1186     1 152          1
```

5. Press Enter to display a report of the rank assignments for your entire storage unit.

4.5.7 Creating encrypted extent pools

Complete this task to create encrypted volume extent pools. This is the first step in configuring new encrypted fixed block storage.

1. Run the **mkextpool** command to create the encrypted extent pool for rank group 0 (zero).

   Run the **mkextpool** command at the **dscli** command prompt with the parameters and variables, as shown in Example 4-42, for fixed block extent pools, where **-encryptgrp 1** represents the encryption group ID and the **P0** represents the extent pool name that you assign. This name can be 16 double-byte characters.

Example 4-42 Create encrypted FB extent pool

```
dscli>mkextpool -dev IBM.2107-75LY981 -rankgrp 0 -stgtype -encryptgrp 1 fb P0
-------------------------------------------------------------------------
Extent pool P0 successfully created.
```

2. Verify the extent pool assignments by running the **lsextpool** command when you are done creating the extent pools. Use the **-l** parameter to display a full report for the extent pools that are assigned to the storage unit.

   Run the **lsextpool** command at the **dscli** command prompt with the parameters and variables that are shown in Example 4-43.

Example 4-43 Verify the extent pool assignments

```
dscli> lsextpool -dev IBM.2107-75LY981 -l
-----------------------------------------------------------------------------
Name       ID stgtype rankgrp status availstor (2^30B) %allocated available rese rved numvols numranks encryptgrp
=============================================================================
raid10P0   P0 fb            0  below               34         97        34 0      24        1          1
```

All ranks and extent pools on a particular encryption-capable DS8000 storage facility image must be configured with the same encryption group attribute. The first rank or encryption group that is configured determines with what the remaining objects must be configured. A value of 0 indicates encryption-disabled. A value of 1 indicates that encryption is enabled.

To change between encryption-enabled and encryption-disabled, all ranks and extent pools must be unconfigured. Unconfiguring an encryption-enabled rank causes any data that is stored on the rank to be cryptographically erased (the disk is instructed to reset its own encryption key) and subsequently overwritten to reinitialize the rank.

4.6 Encryption and Copy Services functions

Copy Services operations are not affected by encrypting drives. The encryption applies only to data at rest, which is the data that is physically written to the disk drives. If you are doing remote replication of the encrypted data, when the data is *read* from the source disk, it is decrypted, and sent across the network link. If the target storage system is also set up for encryption, when the data is written to disk at the target site, it is encrypted again. There is no relationship between the encryption that is done at the source and the encryption at the target. They are independent operations with their own sets of keys and potentially even their own key managers, depending on how the environment is configured.

This encryption strategy also holds true for FlashCopy. Although this copy is a T0 copy of data that resides only with the DS8000, when the source data is read and rewritten to the FlashCopy target volume, it is decrypted at *read* and reencrypted at *write*. The encryption is not intrusive in terms of performance because it is all done by the drives.

4.7 NIST SP 800-131a requirements for key servers

If one or more key servers are configured on the DS8000, the HMC initiates periodic connections to the key servers to monitor whether the key servers are accessible. When an encryption group or recovery key is configured on the DS8000, the HMC initiates connections to the key servers to request key services and periodically verifies that any active data keys are valid on all configured key servers.

Encryption Key Servers use a secure connection with the HMC with IPP. However, TLS 1.2 can also be enabled. IBM Security Key Lifecycle Manager V2.6 includes an NIST SP 800-131a security-compliant Java level to meet this requirement.

The periodic Key Server accessibility monitoring is implemented in the DS/NI server on the HMC. The key services requests and periodic data key validation are implemented in the key client in the storage facility image, and it communicates through the DS/NI server.

The key server network connection uses IPP. The protocol uses a digital certificate to authenticate the key client with the key server, and it has data security for the data keys that are passed between the key client and server. However, use of TLS protocols with the proprietary protocol is recommended to further secure the key server connection between the management server and the key server.

4.7.1 Configuration steps for changing IBM Security Key Lifecycle Manager V2.6 to use TLS 1.2

IBM DS8000 R8.1 does not support Gen-1 certificates and comes with TLS enabled by default. However, the IBM Security Key Lifecycle Manager V2.6 key servers and DS8000 up to Release 8 Code level are not configured to use TLS 1.2 for IBM Security Key Lifecycle Manager to HMC communication by default. The DS GUI does not support making these changes. The IBM Security Key Lifecycle Manager command line and DS CLI must be used to make the changes that are shown in this section. First, export the IBM Security Key Lifecycle Manager SSL certificate, and then investigate the security access level on the DS8000. If necessary, change the security access level on the DS8000, and then redefine the key servers to use TLS 1.2.

Complete the following steps:

1. Log in to the command line of the IBM Security Key Lifecycle Manager host. Modify the configuration file to enable TLS 1.2 communication with the HMC by completing the following steps:
   a. Run the following command:

      ```
      cd /opt/IBM/WebSphere/AppServer/products/sklm/config
      ```

   b. Run the following command:

      ```
      vi SKLMConfig.properties
      ```

   c. Change the following line:

      ```
      TransportListener.ssl.protocols=SSL_TLSv2  ----> add v2
      ```

      This change configures IBM Security Key Lifecycle Manager to support TLS 1.2.

   d. Change the following line:

      ```
      requireSHA2Signatures=true
      ```

      This step configures IBM Security Key Lifecycle Manager to take connections from a client that is in NIST SP 800-131a compliance mode.

      If you are not ready to implement TLS 1.2 communication, do not perform Step d to change `requireSHA2Signatures` to `true`.
2. Log in to the IBM Security Key Lifecycle Manager GUI to create a new SSL certificate. Only one SSL certificate can be active. If an SSL certificate exists, it becomes inactive.
3. With IBM Security Key Lifecycle Manager V2.6, the option was added to export the SSL certificate from the GUI. Using the CLI is still possible, but using the GUI is much easier.

To use the GUI, complete the following steps:

a. Log in to the IBM Security Key Lifecycle Manager GUI to export the new SSL certificate. Click **Advanced Configuration** → **Server Certificates**, as shown in Figure 4-116.

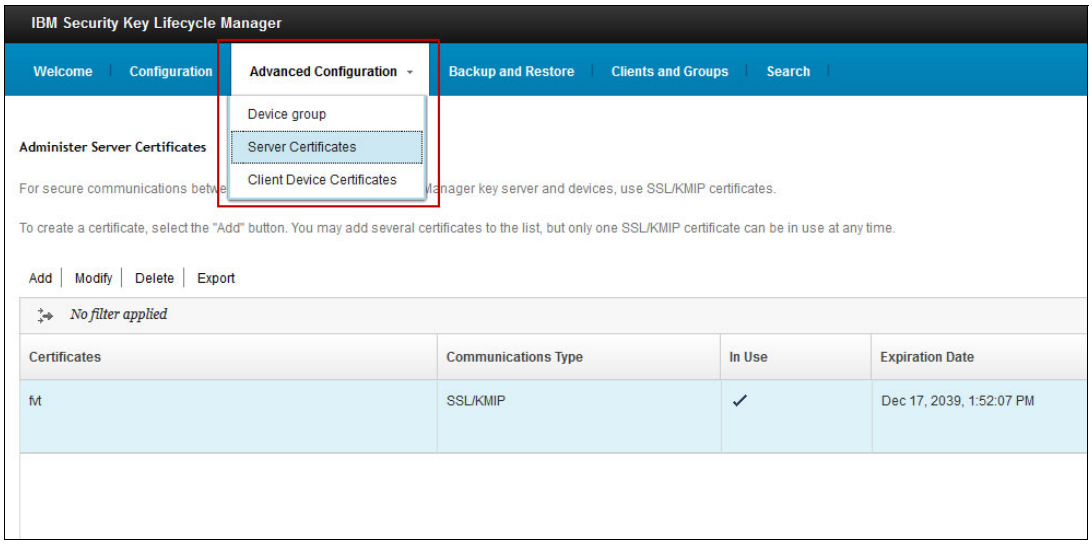


Figure 4-116 Server certificates

b. Right-click the certificate and select **Export**, as shown in Figure 4-117.

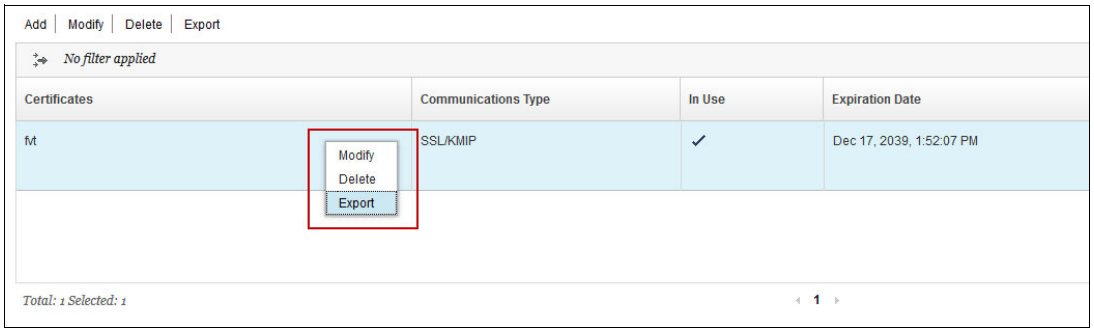


Figure 4-117 Select Export

c. Rename the certificate, if required, and export it to the wanted file location, as shown in Figure 4-118.

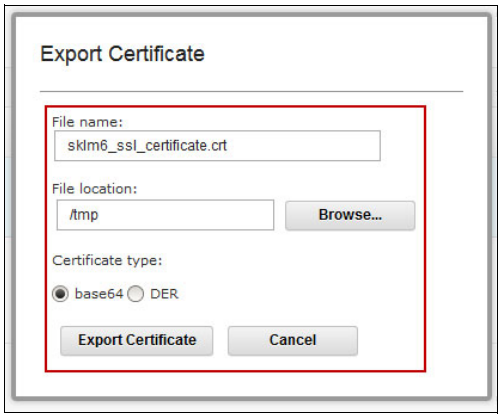


Figure 4-118 Rename and export

To use the CLI, log in to the IBM Security Key Lifecycle Manager command line to identify the active SSL certificate. This certificate must be exported to a file for use when defining key servers on the DS8000. To identify the active SSL certificate, complete the following steps:

a. Run the following command:

```
sklm-rehl64 ~]# cd /opt/IBM/WebSphere/AppServer/bin
```

b. Run the following command:

```
sklm-rehl64 ~]# ./wsadmin.sh -username SKLMAdmin -password passw0rd -lang
jython
```

c. Run the command that is shown in Example 4-44.

Example 4-44 Identify the active SSL certificate on the IBM Security Key Lifecycle Manager

```
wsadmin> print AdminTask.tklmCertList('[-usage SSLSERVER -v y]')
CTGKM0001I Command succeeded.
CTGKM0661I  Found 1 certificates.

uuid                                CERTIFICATE-51e37703-e625-41ac-a050-2c76c68c875a
alias                               ssl_sklm6
information                         null
key store name                      defaultKeyStore
key store uuid                      DUMMY-KEYSTORE-1
owner                               null
key state                           ACTIVE
issuer name                         CN=ssl_sklm6
subject name                        CN=ssl_sklm6
activation date                     10/23/13 7:49:34 PM Central European Summer Time
archive date                        null
compromise date                     null
creation date                       10/23/13 7:49:34 PM Central European Summer Time
expiration date                     10/21/23 7:49:34 PM Central European Summer Time
destroy date                        null
trusted                             1
has private key                     TRUE
serial number                       800958824665661
...
```

Example 4-44 on page 139 shows the default user name and password for the IBM Security Key Lifecycle Manager. Change them to the user name and password that is needed for your configuration.

The UUID shows the certificate that must be exported, and the key state shows the active certificate.

4. Export the certificate to a file by completing the following steps:

   a. Run the following command:

      ```
      sklm-rehl64 ~]# cd /opt/IBM/WebSphere/AppServer/bin
      ```

   b. Run the following command:

      ```
      sklm-rehl64 ~]# ./wsadmin.sh -username SKLMadmin -password passw0rd -lang
      jython
      ```

   c. Run the following command:

      ```
      wsadmin> print AdminTask.tklmCertExport('[-uuid
      CERTIFICATE-a74a853c-4421-4af9-ad29-50062d8dad6d -fileName
      /tmp/sklm6_ssl_certificate.crt]')
      ```

      This command creates a file in `/tmp` with the exported certificate. This file must be added to the file system where DS CLI is running. It is used when the key servers are defined to the DS8000.

5. Log out of the IBM Security Key Lifecycle Manager. Log in to the DS8000 with DSCLI with Storage Administrator authority. If the key servers already are defined to the DS8000, continue with this step. If no key servers is defined, skip to Step 6 on page 141 to install the new certificate that was exported from the IBM Security Key Lifecycle Manager in step 4. Example 4-45 shows how to remove one of the key servers. Perform the tasks in this step 5 and step 6 on page 141 for only one key server.

   Note: Do not delete all working key servers before getting one working with the new certificate and TLS 1.2 communication first. If less than four key servers are defined, then create another one by using same address with the TLS port (The default port is 441).

Example 4-45 Command line to deactivate the key server

```
dscli> lskeymgr -l
-------------------------------------------------
ID  state   status addr         port
==================================
1   active  normal 9.155.000.59 3801
2   active  normal 9.155.000.60 3801
dscli> rmkeymgr 2
dscli> lskeymgr -l
-------------------------------------------------
ID  state   status addr         port
==================================
1   active  normal 9.155.000.59 3801
dscli>
```

6. Now, you are ready to install a new certificate. Example 4-46 shows the **mkkeymgr** command, which you use to define the key servers, install the IBM Security Key Lifecycle Manager certificate that was exported in Step 4 on page 140, and use the IBM Security Key Lifecycle Manager SSL port (441 is the default). The location of the certificate is the location that was chosen in Step 4 on page 140. The location must be in the file system where DS CLI is running. This example creates the key server that was deleted in step 5 on page 140. You must customize these commands to match your environment. This example is from an IBM configuration that was created for the purpose of providing example output.

Example 4-46 Install a NIST SP 800-131a certificate

```
dscli> mkkeymgr -port 441 -addr 9.155.000.60 -cert
/tmp/sklm6_ssl_certificate.crt 2
dscli> lskeymgr -l
-------------------------------------------------
ID  state   status addr         port
=================================
1   active  normal 9.155.000.59 3801
2   active  normal 9.155.000.60 441
dscli>
#Only key server 2 is using SSL port 441, key server one is using TPC port
3801.
```

Notes: You must set the encryption key server SSL port (441 is the default for TLS v1.2 for network communication). Use TPC port 3801 if TLS 1.2 is not being enabled for network communication.

The key server ID is a decimal number 1 - 4. Four is the maximum number of key servers that the DS8000 can support.

Repeat these steps for each encryption key server until all of them are updated. Now, the DS8000 has the new certificate and all key servers can use the new SSL certificate that is created on the key server and SSL port (441 by default) to use TLS 1.2 for HMC to key server communication.

4.8 Migration from a Gen-1 to a Gen-2 certificate for encryption

Note: A Gen2 certificate is set to the DS8000 Release 8.1 in factory. Thus, a migration from Gen-1 to Gen-2 is not possible or required. Only machines that were upgraded from a previous level of code can be migrated.

To use the Gen-2 certificate on the DS8000 for data encryption, complete the following steps:

1. Verify the version of the certificate that is being used for data encryption. Example 4-47 shows how to verify which certificate (Gen-1 or Gen-2) that the DS8000 is using.

Example 4-47 Determine which certificate the DS8000 is using for data encryption

```
dscli> showkeygrp 1
Date/Time: October 23, 2013 7:01:33 PM MST IBM …
ID          1
numranks    1
numpools    1
```

```
state       accessible
reckeystate configured
reckeydate  07/17/2013 23:27:20 MST
datakeydate 04/18/2012 16:27:35 MST
label       ds8k_tuc_02
label2      -
certificate GEN1
dscli>
```

2. Run the **managekeygrp** command, as shown in Example 4-48, to change from the Gen-1 to Gen-2 certificate for data encryption on the DS8000 with Release 7.2 microcode, LMC 7.7.20.xx. No previous models of the DS8000 series support the Gen-2 certificate.

Example 4-48 Update the certificate from Gen-1 to Gen-2 on the DS8000

```
DSCLI> managekeygrp -action updatecert -key data -label ds8k_tuc_02 1
Date/Time: October 23, 2013 7:40:23 PM CET IBM ...
CMUC00472I managekeygrp: The certificate for encryption group 1 has been
updated

If you have secondary certificate label then the -label2 flag must also be
used.
```

3. Run the **showkeygrp 1** command to verify that the Gen-2 certificate is now being used for data encryption (see Example 4-49).

Example 4-49 Verify that the certificate was updated from Gen-1 to Gen-2

```
dscli> showkeygrp 1
Date/Time: October 23, 2013 7:01:33 PM MST IBM …
ID          1
numranks    1
numpools    1
state       accessible
reckeystate configured
reckeydate  07/17/2013 23:27:20 MST
datakeydate 04/18/2012 16:27:35 MST
label       ds8k_tuc_02
label2      -
certificate GEN2
dscli>
```

If the SSL certificate was updated from the key server and the data encryption certificate was updated to Gen-2, the DS8000 encryption configuration is now compliant with NIST SP 800-131a.

4.9 Using A Custom Generated Certificate

With Release 8.2 of DS8880 microcode, a custom certificate can be specified for communication between the encryption key servers (typically SKLM) and the storage system. You can update to a custom defined certificate via DSGUI or DSCLI.

Note: If the current DS8000 encryption certificate is Gen1, before you update to a customer defined certificate, ensure that the certificate authority (CA) signed root certificate is installed on each key server. Encryption certificates must be digitally signed by a CA that is designated as a trusted root CA.

Important: After you update a DS8000 encryption certificate to a customer defined certificate, you can change the certificate back to Gen2 but not Gen1.

4.9.1 Configuring a Custom Certificate via DSGUI

You can update the DS8000 encryption certificate with a custom certificate by using one of the following options:

- The encryption enablement wizard when encryption is not enabled.

  Click on ***Settings*** > **Security > Encryption** then select **Enable Encryption** to start the wizard. See Figure 4-119

Figure 4-119 Enable Encryption

- Update Certificate on the Encryption Settings page when encryption is configured.

  To update the certificate, on the DSGUI home page select ***Settings*** > **Security > Encryption.** The Encryption pannel is shown in Figure 4-120 on page 144.

Figure 4-120 Settings Encryption Window

Click on **Certificate** to view the DS8000 Encryption Certificate. Click **Update Certificate**. The Update DS8000 Encryption Certificate window opens shown in Figure 4-121 on page 145.

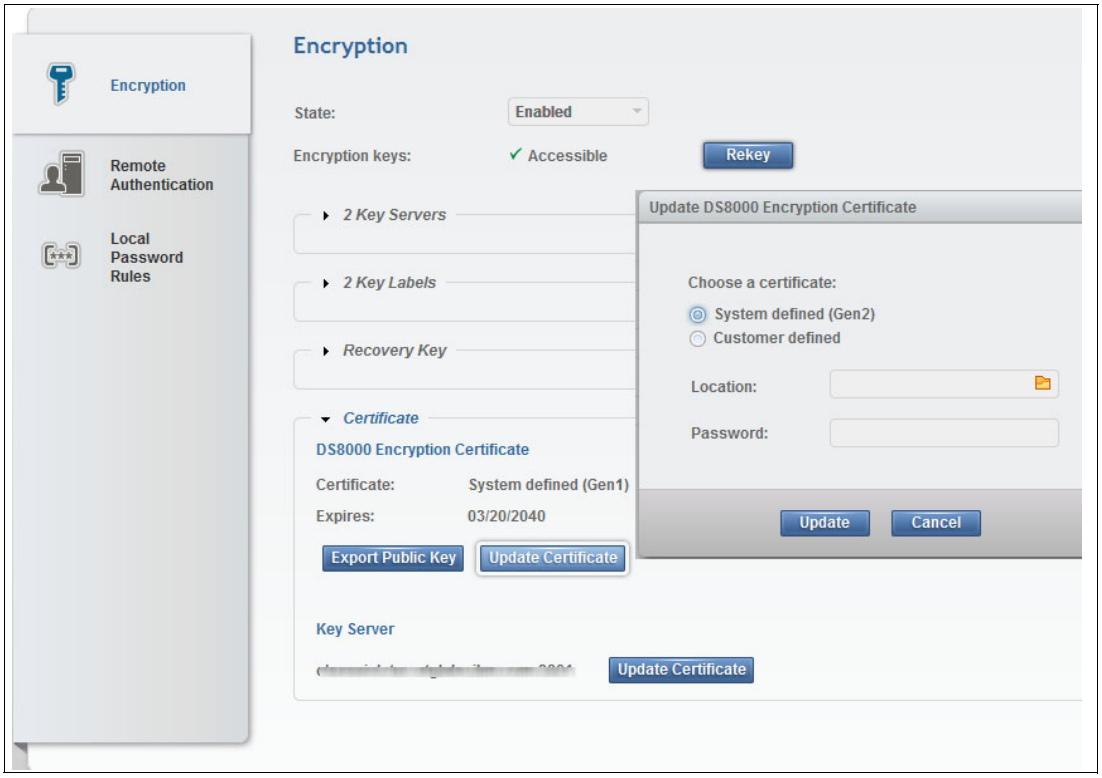


Figure 4-121 Encryption Settings Update Certificate

Click the **Customer defined** radio bottom In the Update DS8000 Encryption Certificate window. Select Customer defined, you must browse for the certificate location and enter a password for the certificate. Click **Update** to update the certificate as shown in Figure 4-121 on page 145.

Figure 4-122 Encryption Update Certificate window

Note: The maximum length of customer-generated machine password is 128 characters. An example of a file name for a customer-generated filename is either 2107-75YZ123.mfg or 2107-75YZ123.p12 and password is XyZABcPQRstu.

4.9.2 Configuring a Custom Certificate via DSCLI

Custom defined certificate can be specified using the DSCLI command, **managekeygrp,** as follows:

- **managekeygrp -action importcert -loc** location **-pw** password encryption_group_ID see Example 4-50.

Example 4-50 importing Custom Certificate

```
dscli> managekeygrp -action importcert -loc /home/hscroot/rashmic/da6.p12 -pw blah
1
IBM DSCLI Version: 0.0.0.0 DS: IBM.2107-75NR641
CMUC00489I managekeygrp: The certificate for encryption group 1 has been imported.
```

- **managekeygrp -action updatecert -loc** location **-certType** customer **-key** data encryption_group_ID see Example 4-51.

Example 4-51 Updating Custom Certificate

```
dscli> managekeygrp -action updatecert -certType CUSTOMER -key data 1
IBM DSCLI Version: 0.0.0.0 DS: IBM.2107-75NR641
CMUC00472I managekeygrp: The certificate for encryption group 1 has been updated.
```

Note: you must specify option **-certType** with **updatecert.** If you do not specify this option, the default is the IBM Gen2 option. Parameter **-key** is also required with ***updatecert*** action.

5

Maintaining the IBM DS8000 encryption environment

This chapter provides information about the maintenance and use of your IBM DS8000 encryption environment.

This chapter contains the following topics:

- Rekeying the data key
- Recovery key use and maintenance

Important: For information about maintaining the IBM Security Key Lifecycle Manager environment, see the IBM Security Key Lifecycle Manager IBM Knowledge Center at:

http://ibm.biz/SKLMv26KC

In particular, pay attention to the backup tasks. Failure to back up your keystore and other critical data properly might result in the unrecoverable loss of all access to your encrypted data. Do not encrypt your backup file, or store a backup file on an encrypting device. Failure to back up data might also result in the inconsistency of the key manager and potential data loss on the storage device.

5.1 Rekeying the data key

The Rekey Data Key option is available on the DS8000. You can use this option with the Storage Administrator role to rekey the data key by changing the data key label. A client might want to use this function to change periodically the data key.

The following procedure describes how to rekey the data key labels. Assume that the former key label is `hmi2` and that a new data key label `hmi3` is defined in the IBM Security Key Lifecycle Manager key servers.

1. Click **Settings** → **Encryption** and expand the **Key Labels** section, as shown in Figure 5-1.

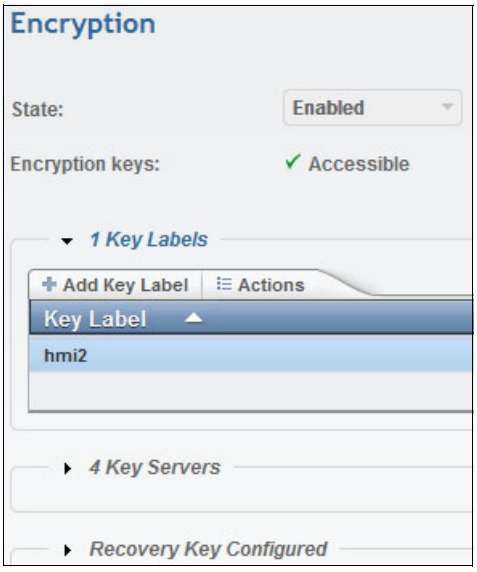


Figure 5-1 Start the Rekey Data Key function

2. Select the key label that you want to change and from the Actions menu, select **Modify** (see Figure 5-2).

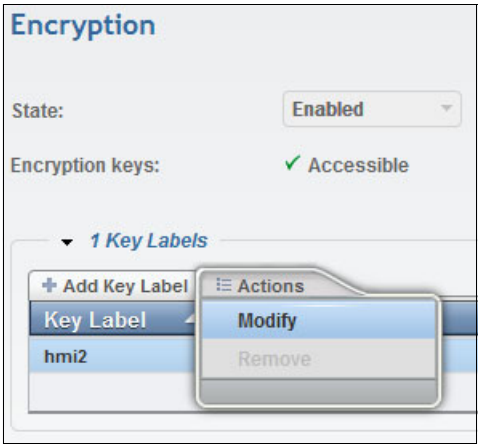


Figure 5-2 Rekey data key: Select Modify

3. The Modify Key Label window opens, as shown in Figure 5-3. Enter the new data key label into the provided field and then click **Modify**.

Figure 5-3 Rekey Data Key window

4. When the rekey task is complete, the confirmation message displays (see Figure 5-4). Click **Close** to return to the main Encryption window.

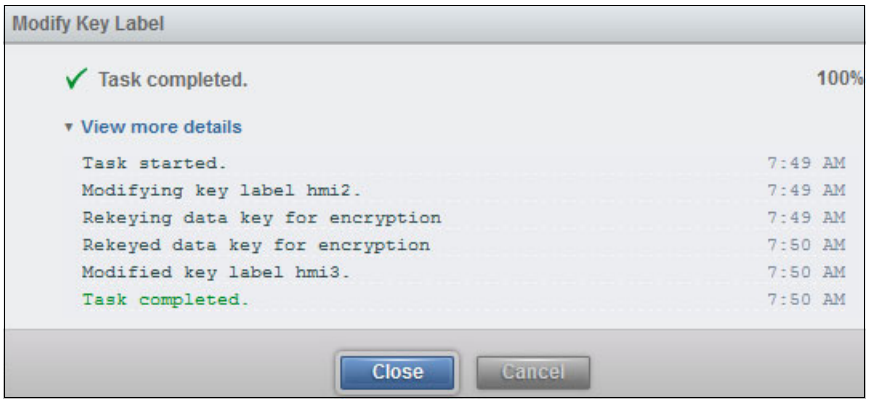


Figure 5-4 Rekey Data Key action complete

In the Encryption window, the new key label is displayed under the Key Labels section.

5.2 Recovery key use and maintenance

Starting with DS8870 Licensed Internal Code level 65.10.xx.xx, two recovery-key-related options are available:

- Recovery key enabling
- Recovery key disabling

When the recovery key is enabled, several functions are available to manage and use the recovery key after its creation:

- Validating or testing a recovery key
- Using the recovery key in an emergency-deadlock situation (recovery action)
- Rekeying the recovery key
- Deleting or deconfiguring a recovery key

When the recovery key is disabled, recovery key enablement is still possible.

The details of each function are described in this section.

5.2.1 Validating or testing a recovery key

Part of the recovery key creation process is the verification of a newly created recovery key. Verification ensures that the recovery key was written correctly. However, after the encryption environment is operational, the recovery key is used only in a deadlock situation. Therefore, verify that the stored recovery key is still valid. The validation process can be performed occasionally and it can be included in your task list for the maintenance of the encryption environment. The Security Administrator can validate the recovery key at any time. Only users with the Security Administrator authority can validate the recovery key. Verifying the recovery key does not change anything in the system, and Storage Administrator approval is not needed.

To validate or test a recovery key, complete the following steps:

1. Log on as a user with the Security Administrator role, click **Settings** → **Encryption**, and click **Test**, as shown in Figure 5-5.

Figure 5-5 Start the Validate /Test Recovery Key function

2. The Test Recovery Key window opens (Figure 5-6). Enter the key into the field and click **Test**.

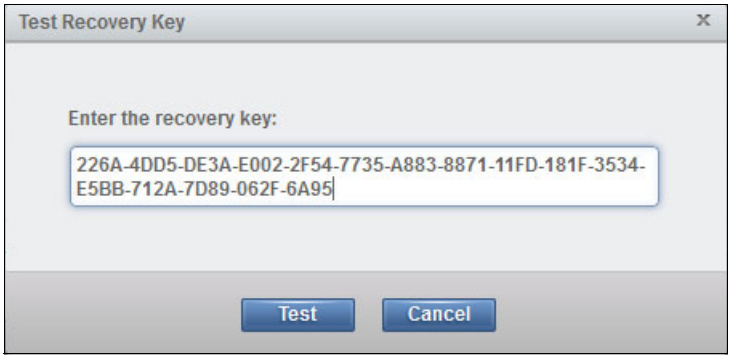


Figure 5-6 Test Recovery Key window

3. When the task is complete, the message in Figure 5-7 displays. Click **Close**.

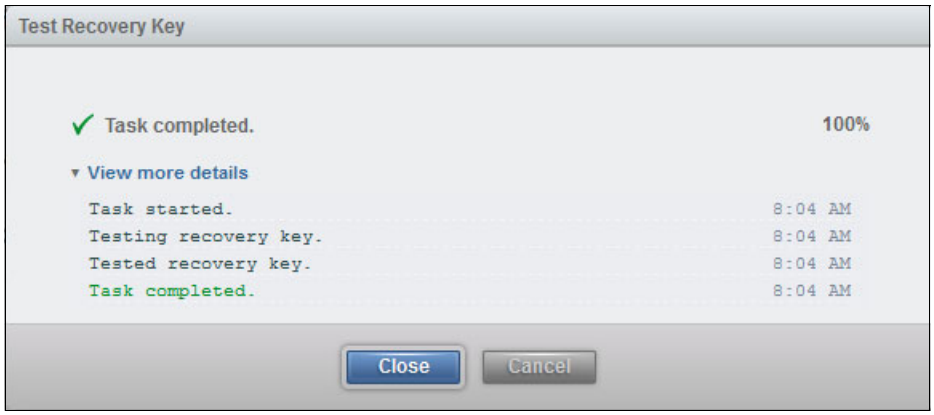


Figure 5-7 Test Recovery Key: Task completed

4. If the key is valid, a successful confirmation message, as shown in Figure 5-8, displays. Click **OK** to return back to the Encryption window.

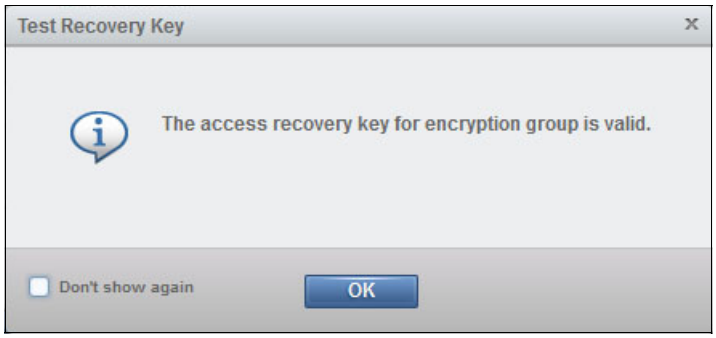


Figure 5-8 Test Recovery Key: Recovery key is valid

The flowchart of this process is shown in Figure 5-9.

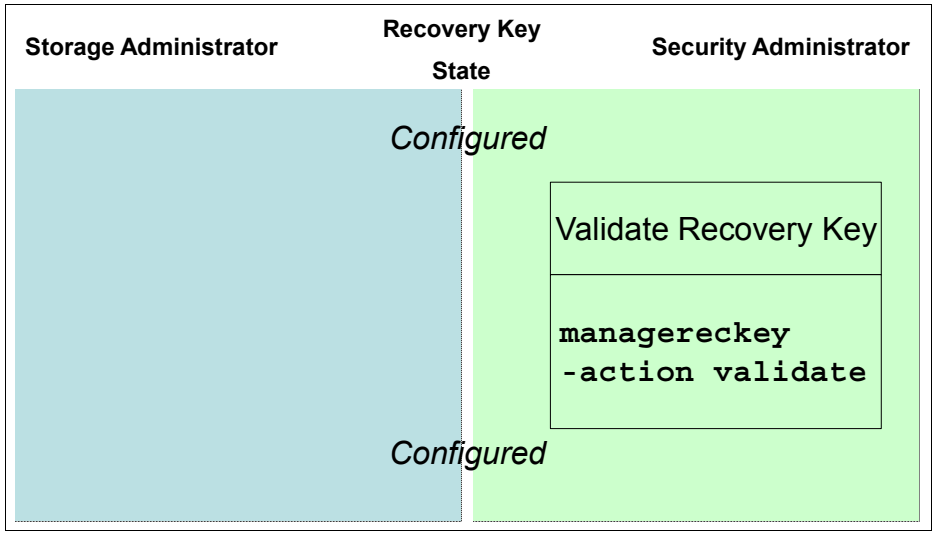


Figure 5-9 Validate recovery key flowchart

5.2.2 Using the recovery key in an emergency-deadlock situation (recovery action)

If the IBM Security Key Lifecycle Manager servers are down or inaccessible for any reason, the DS8000 storage facility image cannot be started because without the keys the storage remains in locked mode. In this situation, two choices are available:

- Repair at least one of the IBM Security Key Lifecycle Manager servers or its network connectivity to serve the necessary key for the DS8000.
- Initiate the recovery process to unlock the DS8000 and to let the volumes be accessible again.

Consider the first option and check the IBM Security Key Lifecycle Manager servers status first. Try to fix the problem with key servers if the problem is not complex, and obviously if you have time to meet your service level agreement (SLA). Otherwise, use the recovery key to initiate the process to unlock DS8000 volumes.

Simulating zn IBM Security Key Lifecycle Manager failure

This section takes you through a real-life example to demonstrate how deadlock recovery works. You can use this scenario soon after implementing encryption in your environment to test and document system recovery in a deadlock situation.

As a first step in this example, shut down all IBM Security Key Lifecycle Manager servers that are connected to the DS8000. You can check the key server status from the DS8000 Storage Manager GUI. Click **Settings** → **Encryption** and expand the key servers section to check the status of your key servers. The state of each server should change to Inaccessible, as shown in Figure 5-10. The Encryption keys status is still Accessible because the keys are stored in the DS8000 cache. However, if you power off/on the DS8000, the encryption keys are gone from the cache and the only way to access the DS8000 data is by obtaining the key from one of the defined IBM Security Key Lifecycle Manager key servers.

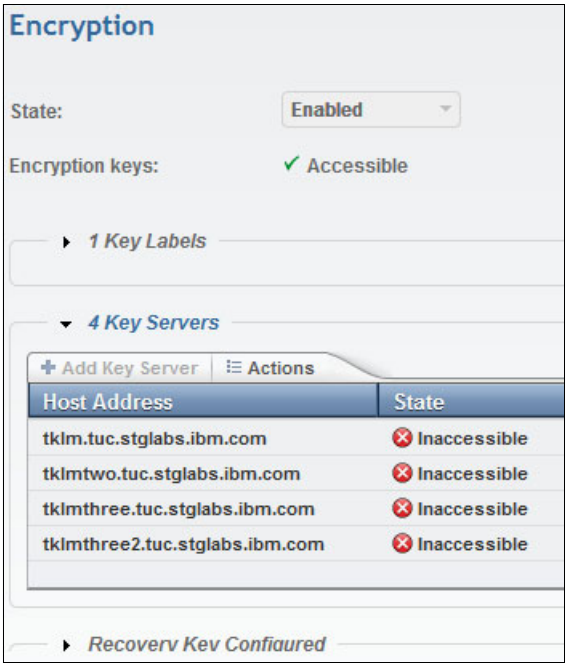


Figure 5-10 IBM Security Key Lifecycle Manager servers status is Inaccessible

The DS8000 Hardware Management Console (HMC) monitors the availability of the IBM Security Key Lifecycle Manager servers. The connection is verified every 5 minutes.

If the HMC detects an outage of a key server, the BE14EAF1 SRC is reported (Figure 5-11), which is only an early notification. When the outage exceeds the 4-hour limit, the BE14EAF2 error is reported as a call-home event, so both the client and IBM are alerted of this incident.

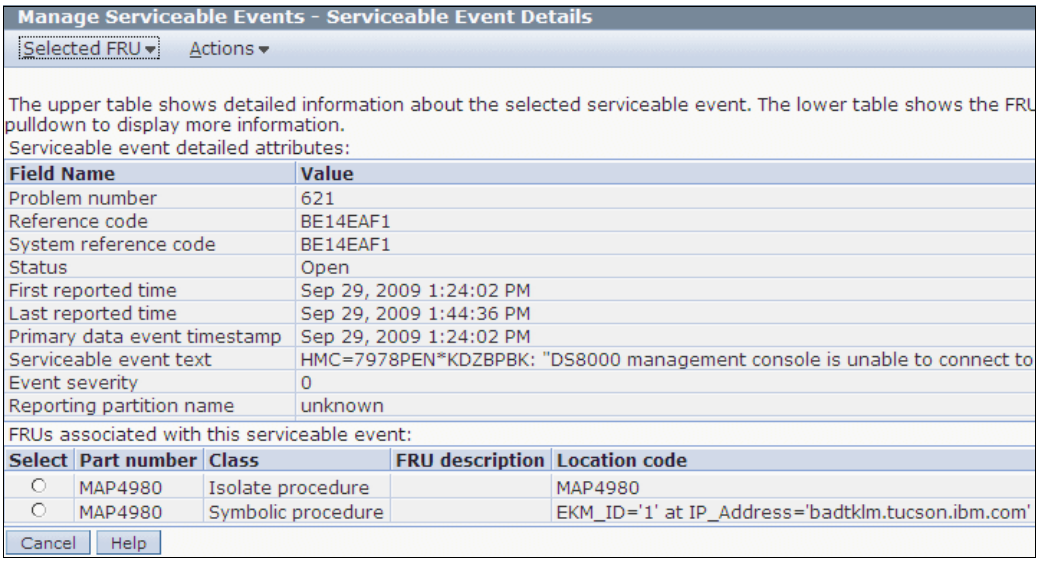


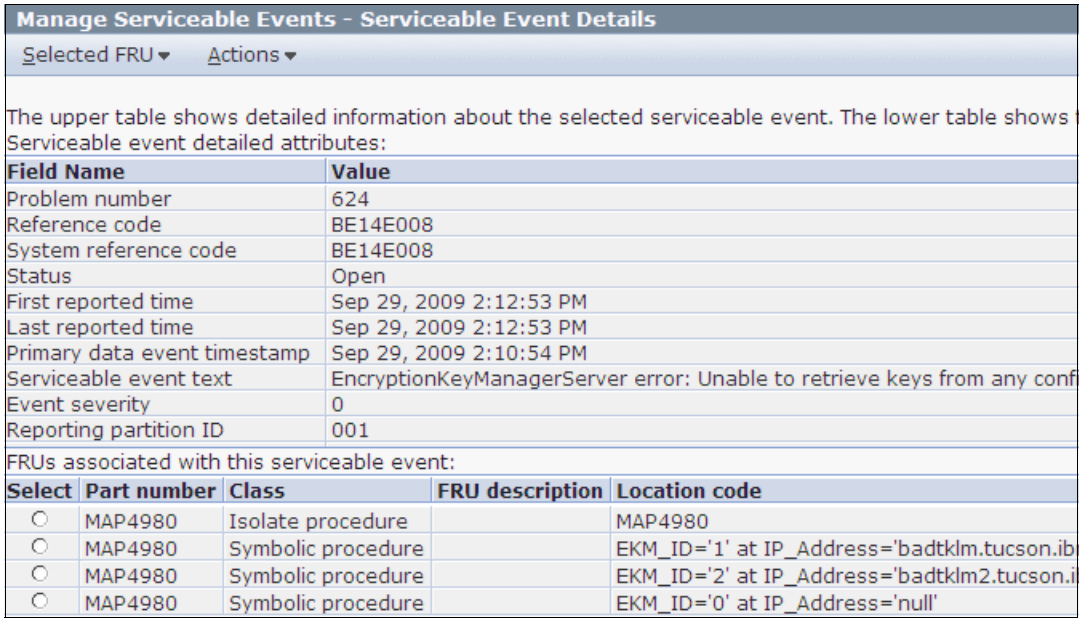
Figure 5-11 BE14EAF1 SRC on DS8000 HMC

The IBM Security Key Lifecycle Manager key servers are accessed only when the key is required for the DS8000 (excluding the heartbeat verification). In most cases, this situation happens when the DS8000 is starting, but several of the reliability, availability, and serviceability (RAS) functions, such as Concurrent Code Load (CCL), also trigger a key retrieval from IBM Security Key Lifecycle Manager key server. The easiest way to reach this point is to turn off the DS8000 by using the DS8000 HMC GUI.

By starting the DS8000 again while the key servers are still down, you notice that the initialization progress is much slower. The reason is that some warm starts are initiated to configure the storage devices. In addition, the DS8000 waits 10 minutes for the IBM Security Key Lifecycle Manager key servers to become available. However, without getting the necessary keys from the IBM Security Key Lifecycle Manager servers, the configuration phase fails and the DS8000 must give up and report the failure.

Possible SRC codes are as follows:

- BE14E008: Unable to retrieve keys from any configured encryption key management servers because of communication errors between the HMC and encryption key management server (or servers). Figure 5-12 shows a sample SRC.



Manage Serviceable Events - Serviceable Event Details

Selected FRU ▾ Actions ▾

The upper table shows detailed information about the selected serviceable event. The lower table shows
Serviceable event detailed attributes:

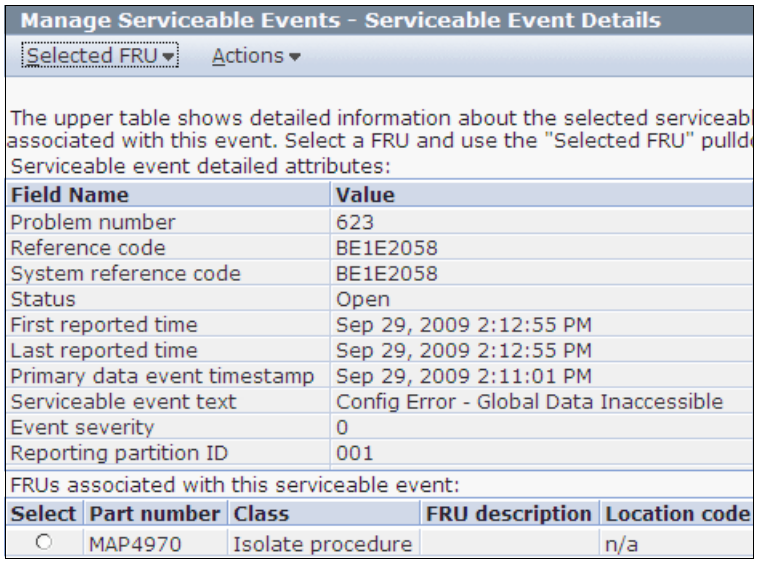
Field Name	Value
Problem number	624
Reference code	BE14E008
System reference code	BE14E008
Status	Open
First reported time	Sep 29, 2009 2:12:53 PM
Last reported time	Sep 29, 2009 2:12:53 PM
Primary data event timestamp	Sep 29, 2009 2:10:54 PM
Serviceable event text	EncryptionKeyManagerServer error: Unable to retrieve keys from any conf
Event severity	0
Reporting partition ID	001

FRUs associated with this serviceable event:

Select	Part number	Class	FRU description	Location code
○	MAP4980	Isolate procedure		MAP4980
○	MAP4980	Symbolic procedure		EKM_ID='1' at IP_Address='badtklm.tucson.ib
○	MAP4980	Symbolic procedure		EKM_ID='2' at IP_Address='badtklm2.tucson.i
○	MAP4980	Symbolic procedure		EKM_ID='0' at IP_Address='null'

Figure 5-12 BE14E008 SRC on DS8000 HMC

- BE1E2058: Config Error - Global Data Inaccessible, as shown in Figure 5-13.



Manage Serviceable Events - Serviceable Event Details

Selected FRU ▾ Actions ▾

The upper table shows detailed information about the selected serviceab
associated with this event. Select a FRU and use the "Selected FRU" pulld
Serviceable event detailed attributes:

Field Name	Value
Problem number	623
Reference code	BE1E2058
System reference code	BE1E2058
Status	Open
First reported time	Sep 29, 2009 2:12:55 PM
Last reported time	Sep 29, 2009 2:12:55 PM
Primary data event timestamp	Sep 29, 2009 2:11:01 PM
Serviceable event text	Config Error - Global Data Inaccessible
Event severity	0
Reporting partition ID	001

FRUs associated with this serviceable event:

Select	Part number	Class	FRU description	Location code
○	MAP4970	Isolate procedure		n/a

Figure 5-13 BE1E2058 SRC on DS8000 HMC

Initiating the recovery process

When all defined IBM Security Key Lifecycle Manager key servers are not accessible, the recovery process should be initiated to get the access to the DS8000 data.

To start the recovery process, complete the following steps:

1. Log on to DS8000 GUI as a user with Security Administrator authority and click **Settings** → **Encryption**. As shown in Figure 5-14, the Encryption **keys** status is Inaccessible due to the broken communication between DS8000 and IBM Security Key Lifecycle Manager key servers. Click **Recover** to initiate the recovery process by using the recovery key.

Figure 5-14 Start the recovery process

2. The Recover Storage System window opens (Figure 5-15). The recovery process requires the valid recovery key. Enter the key into the input field (uppercase characters with dash separation) and click **Recover**.

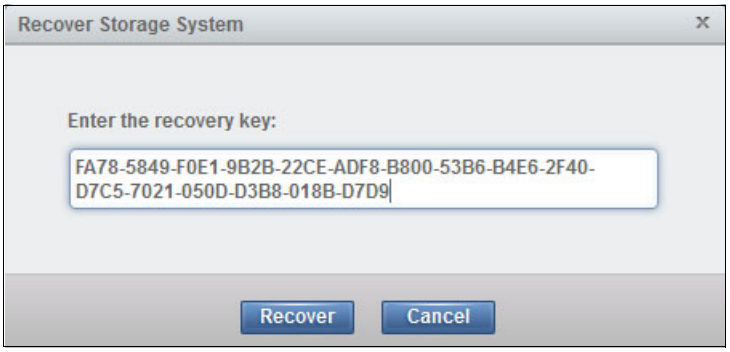


Figure 5-15 Enter the recovery key

3. A task completion message is displayed when the recovery task completes. Click **Close** to continue (see Figure 5-16).

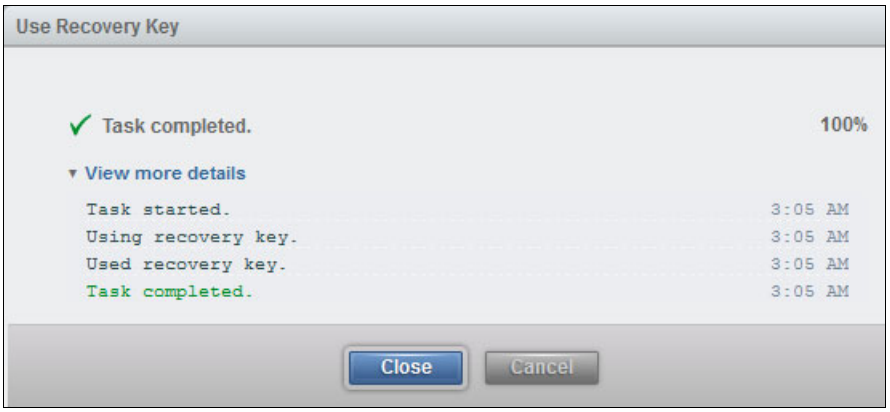


Figure 5-16 The recovery process was initiated successfully

4. Figure 5-17 shows that the recovery key has a pending authorization request that is addressed to Storage Administrator user. The Encryption keys are still in the Inaccessible state.

Figure 5-17 Request Recovery Authorization Pending state

5. At this stage, the user with Storage Administrator authority must approve this recovery request. Therefore, log on as a Storage Administrator and click **Settings** → **Encryption**. Expand the **Recovery Key Authorization pending (initiate recovery)** section. There are two options available: Authorize and Decline (see Figure 5-18).

Figure 5-18 Start Authorize Recovery Key Update

6. After the authorization task completes, click **Close** to complete the recovery action (see Figure 5-19).

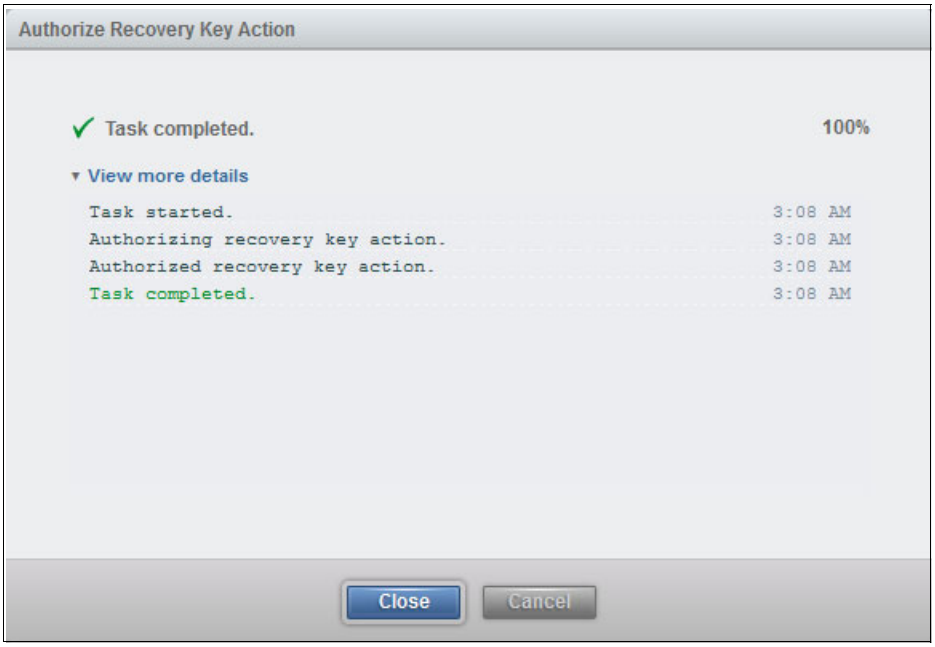


Figure 5-19 Authorize Recovery Key task completed

The recovery process is now completed and you are logged off due to the DS8000 restart process.

You can follow the restart process by viewing the DS8000 status messages at the HMC (Figure 5-20).

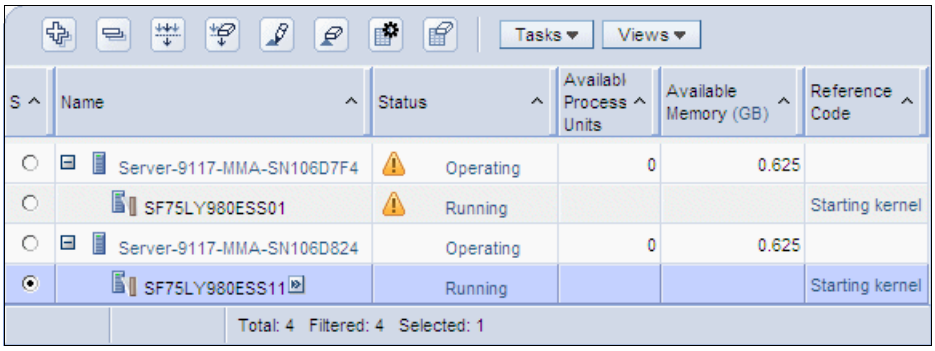


Figure 5-20 Storage facility image restart is in progress

7. Depending on the DS8000 configuration, you must wait several minutes to finish the initialization. When it is running, log on with as a Storage Administrator user and click **Settings** → **Encryption**. The Encryption keys status changed to the Accessible state, which means that the DS8000 volumes are online and accessible from hosts (see Figure 5-21).

Figure 5-21 Encryption keys are accessible

Important: This operation is not permanent. The recovered DS8000 is unlocked only until the next power cycle. However, while the system is running, you have time to repair the IBM Security Key Lifecycle Manager key servers and the communication links between key servers and DS8000. If all the IBM Security Key Lifecycle Manager key servers are lost forever (and there is no backup available), the encryption must be reenabled in the future, which is a destructive process, and all the client data must be offloaded first.

Figure 5-22 shows the flowchart and corresponding DS CLI commands of the recovery process.

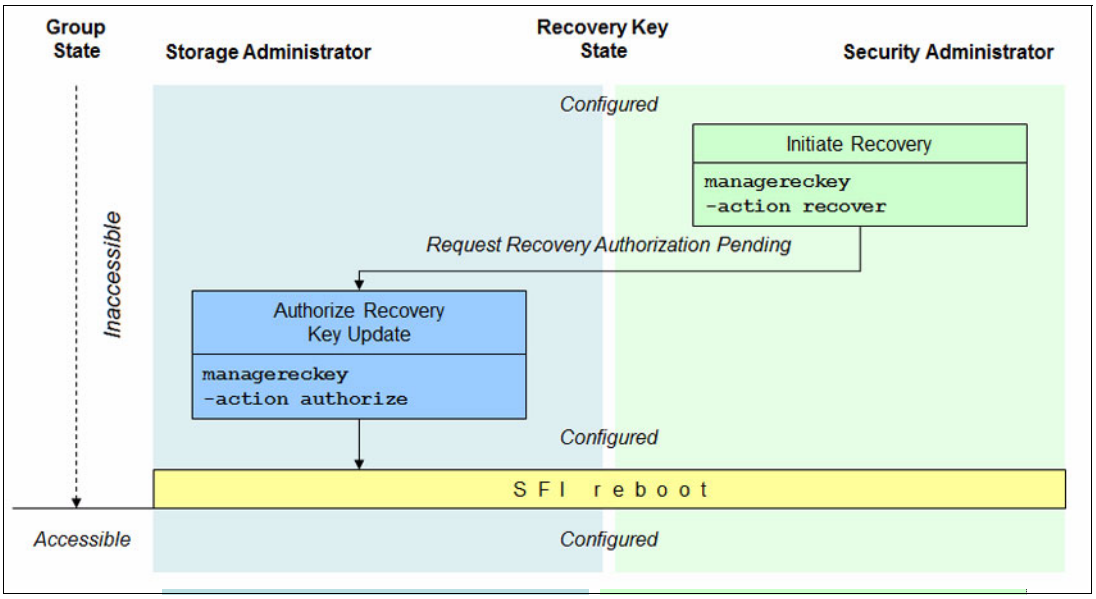


Figure 5-22 Initiate Recovery flowchart

5.2.3 Rekeying the recovery key

In the case of a lost recovery key or an unauthorized person is suspected to have access to the key, the Security Administrator can decide to generate a new recovery key. The old key is revoked and cannot be used anymore. The name of this process is *rekey recovery key*.

To rekey the recovery key, complete the following steps:

1. Log on to the DS8000 GUI as a user with Security Administrator authority and click **Settings** → **Encryption**. Expand the **Recovery Key** section and select **Rekey**, as shown in Figure 5-23.

Figure 5-23 Rekey recovery key

2. The rekey task starts. After it completes, the task completion message displays (see Figure 5-24). Click **Close** to continue.

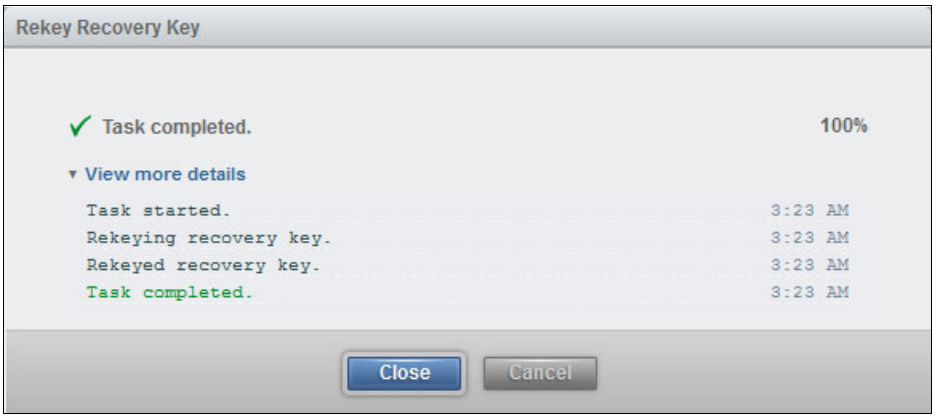


Figure 5-24 Rekey recovery key: Task completion

3. The next window (Figure 5-25) displays the newly generated recovery key. You must record the key (select and copy). It is required for validation in step 4. Click **Rekey**.

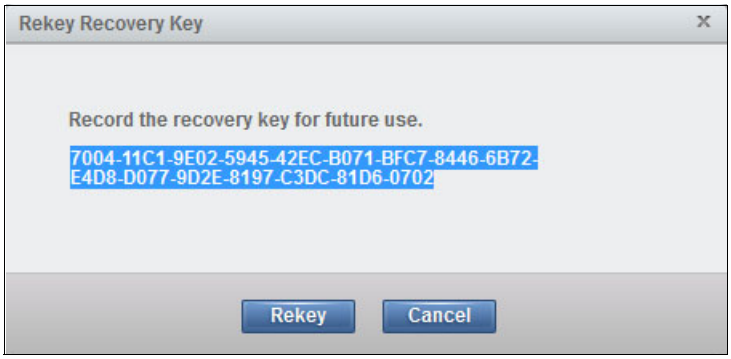


Figure 5-25 Rekey recovery key: Generated

4. Although the new key replaces the old one, do not shred the old key yet because the old key is still active until the Storage Administrator approves the new recovery key. The process is similar to the process of creating a key (see 4.4.3, “Creating the recovery key” on page 99).

   Verify that the new key is written correctly by entering the key text into the input field and clicking **Verify** (Figure 5-26).

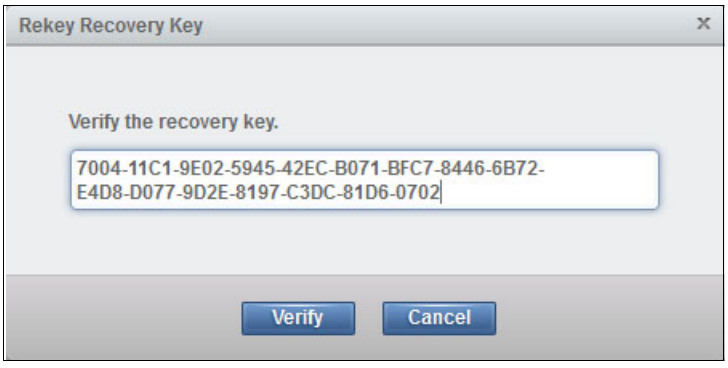


Figure 5-26 Rekey recovery key: Verification

5. When the recovery key verification task completes, the confirmation message displays (Figure 5-27). Click **Close**.

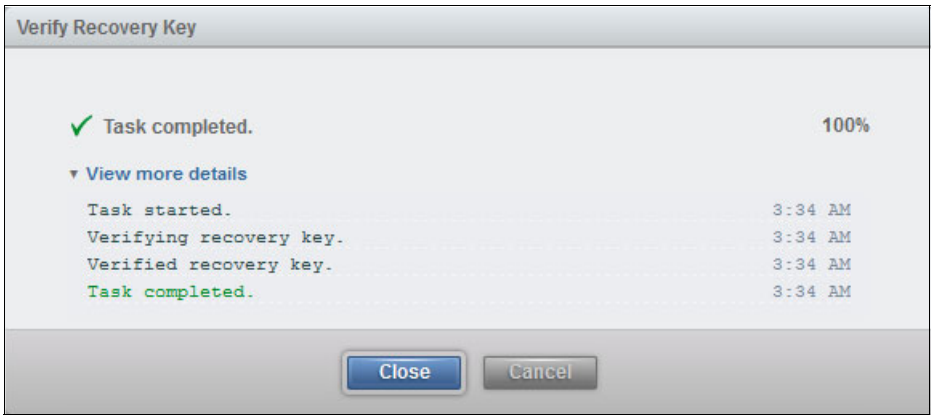


Figure 5-27 Rekey recovery key: Verification

6. In the Encryption window (Figure 5-28), the Recovery Key is now in the Authorization pending (rekey) state, which indicates that any user with Storage Administrator authority must approve this rekey request. Contact the Storage Administrator user to authorize the new recovery key.

Figure 5-28 Rekey recovery key: Authorization pending

7. A user with Storage Administrator authority logs on and clicks **Settings** → **Encryption**. Expand the Recovery Key section and click **Authorize**, as shown in Figure 5-29.

Figure 5-29 Authorize the recovery key

8. The recovery key authorization task starts. After it completes, the confirmation message displays, as shown in Figure 5-30. Click **Close**.

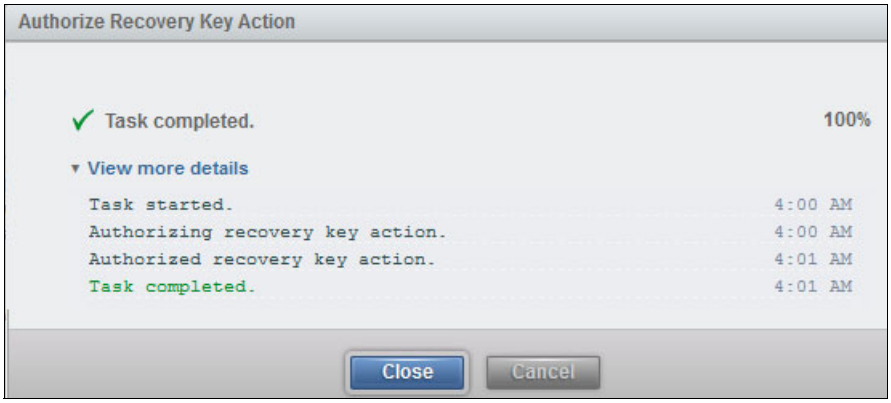


Figure 5-30 Authorize recovery key: Task completion

Now, only the new recovery key is valid. The old key is revoked. The state of the key changes back to a Configured state, as shown in Figure 5-31.

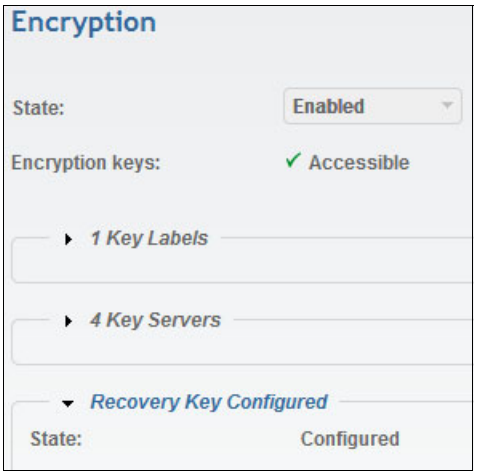


Figure 5-31 Recovery key configured

The flowchart of the rekey recovery key process is shown in Figure 5-32. The corresponding DS CLI commands are also provided.

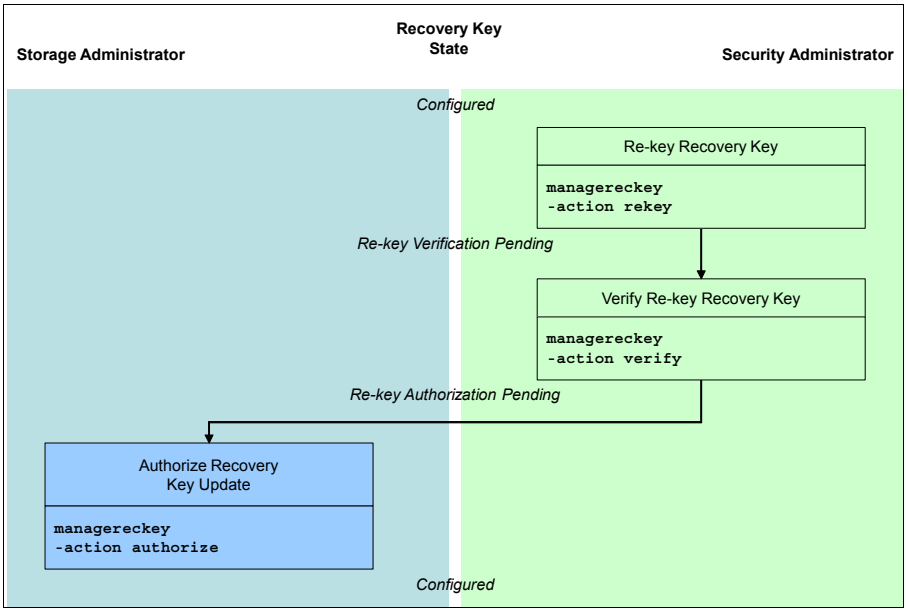


Figure 5-32 Rekey recovery key flowchart

5.2.4 Deleting or deconfiguring a recovery key

Deleting the actual recovery key might be needed only if the client wants to convert an encryption-enabled DS8000 to encryption-disabled mode. As a prerequisite, the encryption should be disabled, that is, delete all DS8000 volumes, ranks, and extent pools.

To delete or deconfigure a recovery key, complete the following steps:

1. Log on to the DS8000 GUI as a user with Security Administrator authority and click **Settings** → **Encryption**. Click **Deconfigure**, as shown in Figure 5-33.

Figure 5-33 Delete/deconfigure recovery key

2. The Deconfigure Recovery Key task starts. After it completes, the confirmation message displays, as shown in Figure 5-34. Click **Close**.

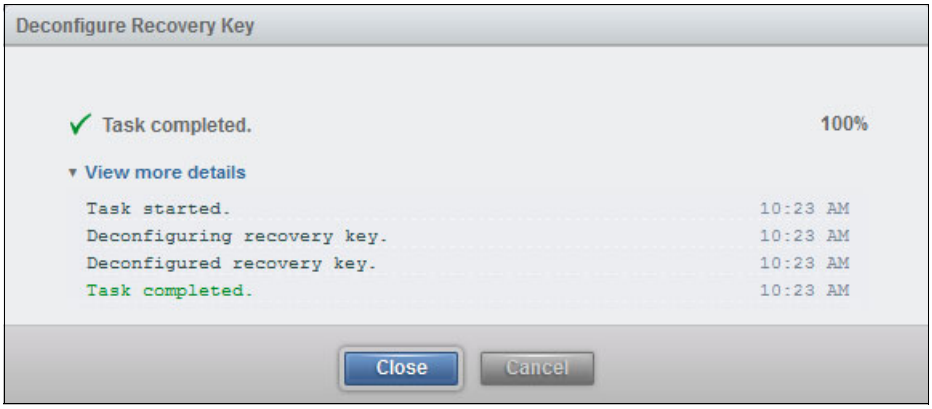


Figure 5-34 Delete Recovery Key window

Figure 5-35 shows that the Recovery Key State changed to a Deconfigure Key Authorization Pending state.

Figure 5-35 Deconfigure key authorization pending

3. The system is waiting for a Storage Administrator to authorize this request. A user with the Storage Administrator authority logs on and clicks **Settings** → **Encryption**. Expand the **Recovery Key** section and click **Authorize**. This action completes the process of deleting Recovery Key. The encryption is disabled and starting from this state, non-encrypted arrays, and ranks can be created on this storage system.

Figure 5-36 shows the flowchart and the corresponding DS CLI commands of the delete recovery key process.

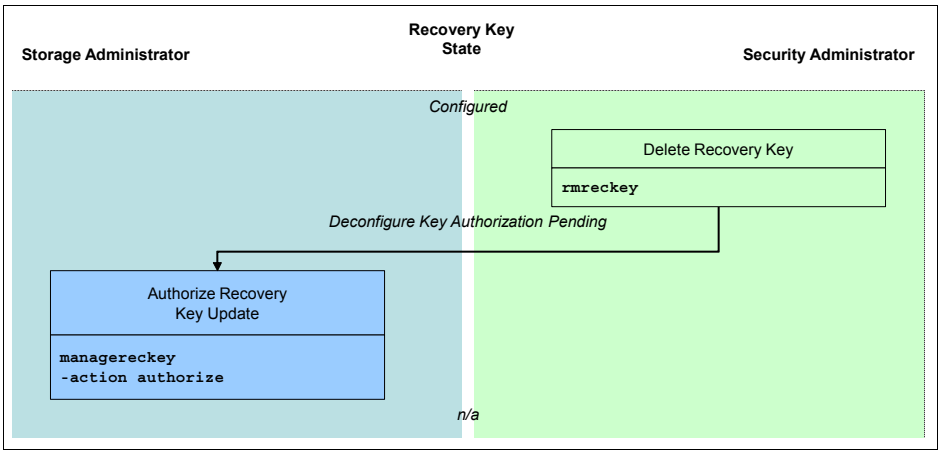


Figure 5-36 Delete recovery key flowchart

5.3 Recovery key state summary

This chapter introduced recovery key functions. In most cases, the recovery key has multiple temporary states. Figure 5-37 summarizes all possible recovery key states and their relationships.

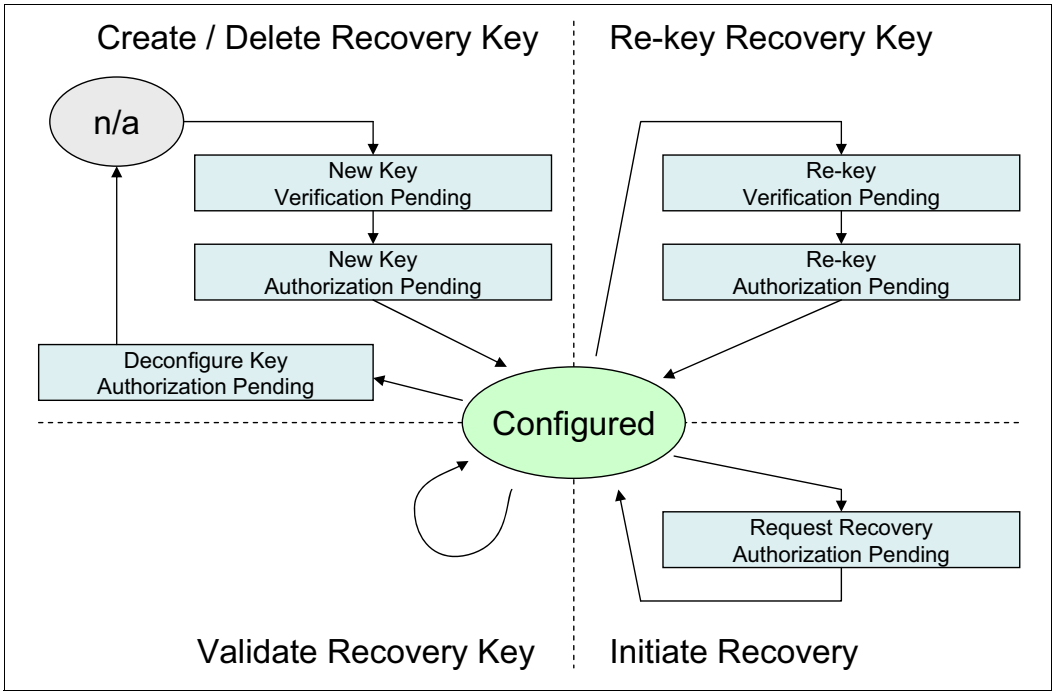


Figure 5-37 Overview of the recovery key states

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publication provides additional information about the topic in this document. It might be available in softcopy only.

- *IBM DS8880 Architecture and Implementation (Release 8.2.1)*, SG24-8323

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *IBM DS8880 Release 8.2 Introduction and Planning Guide*, GC27-8525
- *IBM Security Key Lifecycle Manager for z/OS Version 1.1 Planning, and User's Guide*, SC14-7628

Online resources

These websites are also relevant as further information sources:

- IBM DS8000 Series V8.2 documentation:

  https://www.ibm.com/support/knowledgecenter/ST5GLJ_8.2.1/com.ibm.storage.ssic.help.doc/f2c_ichomepage_v8.21.html

- IBM Security Key LifeCycle Manager V2.6 documentation:

  http://www.ibm.com/support/knowledgecenter/SSWPVP_2.6.0/com.ibm.sklm.doc/welcome.htm

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-4500-06

ISBN 0738455407

Printed in U.S.A.