Redbooks
ibm.com/redbooks

IBM DS8000 Encryption for Data at Rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 Release 9.1)

Bert Dufrasne

Rinkesh Bansal

Tony Eriksson

Leandro Cesar Fida

Lisa Martinez

Security

Storage

IBM

Redpaper

IBM Redbooks

IBM DS8000 Encryption for Data at Rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 R9.1)

April 2021

REDP-4500-09

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

Tenth Edition (April 2021)

This edition applies to IBM DS8900F storage systems that are available with IBM DS8000 Licensed Machine Code (LMC) 7.9.10 (bundle version 89.10.xx.x), referred to as Release 9.1.

© Copyright International Business Machines Corporation 2009, 2021. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices . . . vii
Trademarks . . . viii

Preface . . . ix
Authors . . . ix
Now you can become a published author, too . . . x
Comments welcome . . . xi
Stay connected to IBM Redbooks . . . xi

Summary of changes . . . xiii
April 2021, Tenth Edition . . . xiii

Chapter 1. Encryption overview . . . 1
1.1 Business context . . . 2
1.1.1 Threats and security challenges . . . 2
1.1.2 Need for data at rest encryption . . . 3
1.1.3 Need for Transparent Cloud Tiering encryption . . . 4
1.1.4 Need for IBM Fibre Channel Endpoint Security . . . 5
1.2 Encryption concepts and terminology . . . 7
1.2.1 Symmetric key encryption . . . 7
1.2.2 Asymmetric key encryption . . . 8
1.2.3 Hybrid encryption . . . 10
1.2.4 Communication protocols: IBM Proprietary Protocol, SSL/TLS 1.2, and Key Management Interoperability Protocol . . . 11
1.3 Encryption challenges . . . 12

Chapter 2. External key managers . . . 13
2.1 External key managers for IBM DS8000 systems . . . 14
2.2 IBM Security Key Lifecycle Manager . . . 14
2.2.1 IBM Security Key Lifecycle Manager V4.0 . . . 15
2.2.2 Key serving . . . 17
2.2.3 How to protect IBM Security Key Lifecycle Manager data . . . 18
2.3 IBM Security Key Lifecycle Manager for z/OS . . . 18
2.3.1 IBM Security Key Lifecycle Manager for z/OS components . . . 19
2.3.2 Functions that are performed by IBM Security Key Lifecycle Manager for z/OS . 20
2.3.3 Preventing a deadlock situation . . . 20
2.3.4 Installing the IBM Security Key Lifecycle Manager for z/OS and keystores . . . 21
2.4 Gemalto SafeNet KeySecure . . . 24
2.5 Thales Vormetric Data Security Manager . . . 25
2.6 Thales CipherTrust Manager . . . 25

Chapter 3. IBM DS8000 encryption mechanisms . . . 27
3.1 DS8000 data at rest disk encryption . . . 28
3.1.1 Key management for IBM Proprietary Protocol with IBM Security Key Lifecycle Manager . . . 31
3.1.2 Key management by using KMIP . . . 41
3.2 Encryption deadlock . . . 43
3.3 Working with a recovery key . . . 44
3.3.1 Recovery key management . . . 45

© Copyright IBM Corp. 2009, 2021. All rights reserved.

3.3.2 Disabling or enabling a recovery key 49
3.4 Dual key server support (IBM Proprietary Protocol only) 50
3.5 DS8000 TCT encryption Key Management using KMIP 52
3.6 DS8000 endpoint encryption key management using KMIP 56
3.6.1 IBM Fibre Channel Endpoint Security settings and policies 61

Chapter 4. Planning and guidelines for IBM DS8000 encryption 63
4.1 About certificates 64
4.2 Planning and implementation process flow 65
4.3 Encryption-capable DS8000 ordering and configuration 66
4.4 Licensing 67
4.5 Advice for encryption in storage environments 67
4.5.1 Using LDAP authentication 67
4.5.2 Availability 68
4.5.3 Encryption deadlock prevention 68
4.6 Multiple key managers for redundancy 71

Chapter 5. IBM DS8000 encryption implementation 73
5.1 Installing IBM Security Key Lifecycle Manager V4.0 74
5.1.1 Before you start the installation 75
5.1.2 Silent mode installation on Linux 75
5.1.3 Graphical user interface mode installation on Linux 77
5.1.4 Verifying the IBM Security Key Lifecycle Manager installation 81
5.1.5 Installing Fix Pack 2 (or later) for IBM Security Guardium Key Lifecycle Manager Version 4.0 82
5.2 WebSphere, Java, and IBM Security Key Lifecycle Manager hardening 83
5.2.1 WebSphere Application Server hardening 83
5.2.2 Java hardening 83
5.2.3 IBM Security Key Lifecycle Manager hardening 84
5.3 Migrating from an earlier version of IBM Security Key Lifecycle Manager to IBM Security Key Lifecycle Manager V4.0 85
5.3.1 Migration by using inline migration 85
5.3.2 Migration by using cross migration 92
5.4 Key manager setup 93
5.4.1 Configuring IBM Security Key Lifecycle Manager V4.0 93
5.4.2 Gemalto SafeNet KeySecure configuration 117
5.4.3 Configuring Thales Vormetric Data Security Manager 133
5.4.4 Configuring Thales CipherTrust Manager 140
5.5 Configuring data at rest 151
5.5.1 Setting up IBM Security Key Lifecycle Manager Key management by using IBM Proprietary Protocol 151
5.5.2 IBM Security Key Lifecycle Manager Key management setup by using KMIP . . 158
5.5.3 DS8000 configuration for data at rest encryption in DS GUI 158
5.5.4 DS8000 CLI configuration for data at rest encryption 172
5.5.5 Various authentication mechanisms in IBM Security Key Lifecycle Manager . . . 177
5.6 Configuration for TCT encryption 179
5.6.1 Setting up TCT encryption 180
5.7 Configuration for IBM Fibre Channel Endpoint Security 188
5.7.1 DS8000 GUI configuration for IBM Fibre Channel Endpoint Security 188
5.7.2 DS8000 CLI configuration for IBM Fibre Channel Endpoint Security 192
5.8 Data at rest encryption and Copy Services functions 194
5.9 NIST SP 800-131a requirements for key servers 194
5.9.1 Configuring IBM Security Key Lifecycle Manager to use TLS 1.2 194

5.10 Migrating certificates 199
5.10.1 Migrating from a Gen 1 to a Gen 2 certificate for encryption 199
5.10.2 Migrating from a Gen 2 to a Gen 3 certificate for encryption 200
5.11 Using a custom-generated Gen 1 or Gen 2 certificate 208
5.11.1 Configuring a custom certificate by using the DS GUI 208
5.11.2 Configuring a custom certificate by using DS CLI. 210

Chapter 6. Maintaining the IBM DS8000 encryption environment 213
6.1 Rekeying the data key for data at rest encryption. 214
6.1.1 Rekeying the data key when using the IBM Proprietary Protocol 214
6.1.2 Rekey the data key when using the KMIP protocol. 219
6.2 Recovery key use and maintenance 221
6.2.1 Validating or testing a recovery key 221
6.2.2 Using the recovery key in an emergency-deadlock situation (recovery action) . 223
6.2.3 Rekeying the recovery key 231
6.2.4 Deleting or deconfigure a recovery key. 236
6.3 Recovery key state summary 239

Related publications 241
IBM Redbooks 241
Other publications 241
Online resources 241
Help from IBM 241

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.

© Copyright IBM Corp. 2009, 2021.

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®
Db2®
DS8000®
FICON®
FlashCopy®
Guardium®
HyperSwap®
IBM®
IBM Security™
IBM Spectrum®
IBM Z®
Passport Advantage®
POWER®
POWER9™
RACF®
Redbooks®
Redbooks (logo) ®
System z®
Tivoli®
WebSphere®
XIV®
z/OS®
z15™

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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.

Gemalto Safenet KeySecure (KS), Thales Vormetric Data Security Manager (DSM), Thales CipherTrust Manager documentation and other artifacts Reprinted by Permission of Thales DIS CPL USA, Inc.

Preface

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

The IBM DS8000® supports encryption-capable hard disk drives (HDDs) and flash drives. These Full Disk Encryption (FDE) drive sets are used with key management services to allow encryption for data at rest (DAR). Use of encryption technology involves several considerations that are critical for you to understand to maintain the security and accessibility of encrypted data.

Important: Failure to follow the requirements that are described in the IBM Redpaper can result in an *encryption deadlock*.

Starting with Release 8.5 code, the DS8000 also supports Transparent Cloud Tiering (TCT) data object encryption. With TCT encryption, data is encrypted before it is transmitted to the cloud. The data remains encrypted in cloud storage and is decrypted after it is transmitted back to the DS8000.

Starting with DS8000 Release 9.0, the DS8900F provides IBM Fibre Channel Endpoint Security when communicating with an IBM z15™, which supports link authentication and the encryption of data that is in-flight. For more information, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

This edition focuses on IBM Security™ Key Lifecycle Manager V4.0 or later with the DS8000 Release 9.1 code or later and an updated DS GUI for encryption functions. Other external key managers, such as Thales CipherTrust Manager, Thales Vormetric Data Security Manager (DSM), and Gemalto SafeNet KeySecure (KS) are referenced as supported for DAR encryption and TCT encryption.

Authors

This paper was produced by a team of specialists from around the world.

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 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.

Rinkesh Bansal is Senior Development and Release Manager for the Encryption and Key Management product at IBM. He is an expert in the Key Management domain and has 17 years of experience, with 11 years in the security domain. He joined IBM in 2009 and worked with the IBM Security Key Lifecycle Manager team since 2012. His experience includes roles as an installation package developer, test engineer, test lead, automation lead, project manager, and release manager. He manages the development team of IBM Security Key Lifecycle Manager. He is a passionate innovator with nine patents.

© Copyright IBM Corp. 2009, 2021. All rights reserved.

Tony Eriksson works at IBM Systems Lab Services Nordic, based in Stockholm, Sweden. He joined IBM in 1988 as a mainframe IBM service representative. From 1999, he has worked as a product specialist on enterprise storage solutions, including IBM DS8000, IBM SAN Volume Controller, and storage area network (SAN) products. In 2014, he joined an IBM Business Partner as a pre-sales and an implementation consultant mainly migrating to IBM storage products. In 2019, he joined IBM Systems Lab Services Nordic working close with the Swedish technical sales team for storage solutions. He holds a Technical College Graduation in Electrical Engineering.

Leandro Cesar Fida is an IT Specialist Certified for IBM Global Technology and Services in Brazil. He has 20 years of experience in IBM Z® and its predecessors. He is experienced in working with large system installations and IBM Z server migrations. He holds a degree in information systems from Faculdade Metrocamp and a post-graduation degree in architecture and IT solutions. His areas of expertise include IBM z/OS®, architecture, security, and cryptography.

Lisa Martinez has been working in the Washington Systems Center-Storage Team as a storage specialist since January 2012. Her focus has been with pre-sales support for IBM DS8000 and the IBM Spectrum® Accelerate family of storage products (FlashSystem A9000 and A9000R, IBM XIV®, and IBM Spectrum Accelerate). She is also the lead instructor for FlashSystem A9000 and A9000R, XIV, and Spectrum Accelerate customer-based workshops. Her experience includes roles as a storage architect in the Specialty Services Area in GTS and test architect in disk storage, focusing on system-level test for XIV for three years, and copy services for DS8000. Lisa holds degrees in computer science from New Mexico Highlands University and electrical engineering from the University of New Mexico.

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

Justin Cripps, Roger Hathorn, Matthew Houzenga, Jacob Sheppard, Robert Tondini, Alexander Warmuth
IBM

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

- 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 DS8000 Encryption for Data at Rest, Transparent Cloud Tiering, and Endpoint Security (DS8000 R9.1)

April 2021, Tenth Edition

This revision includes the following new and changed information.

New information

- IBM Security Key Lifecycle Manager migration procedure
- Support for more external key servers: Thales CipherTrust Manager and Thales Vormetric Data Security Manager (DSM)

Changed information

- Updated DS GUI configuration procedure
- Revised instructions for IBM Security Key Lifecycle Manager for z/OS
- Removed instructions that related to older versions of the IBM Security Key Lifecycle Manager

© Copyright IBM Corp. 2009, 2021. All rights reserved.

1

Encryption overview

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

When Transparent Cloud Tiering (TCT) is used, DS8000 Release 8.5 and above allows encryption before data is transmitted to the cloud.

DS8000 Release 9.0 supports encryption for host data communication with IBM Z. For more information about this feature, see *IBM DS8900F Architecture and Implementation Release 9.1*, SG24-8456.

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 encrypted data that a key server manages, as described in 3.2, “Encryption deadlock” on page 43.

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

This chapter includes the following topics:

- 1.1, “Business context” on page 2
- 1.2, “Encryption concepts and terminology” on page 7
- 1.3, “Encryption challenges” on page 12

© Copyright IBM Corp. 2009, 2021. All rights reserved.

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 shows how threats and challenges add to the complexity and the cost of running your business.

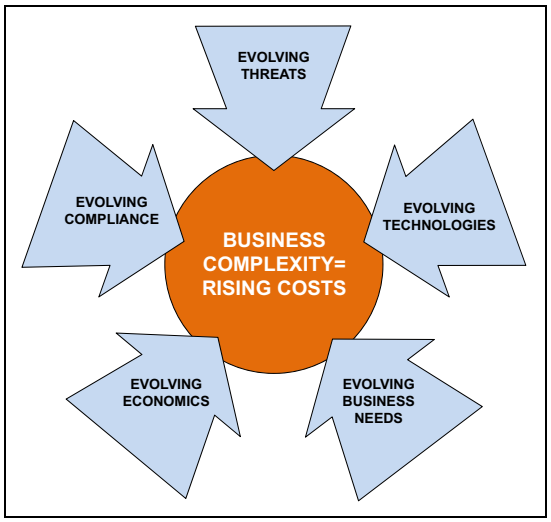


Figure 1-1 Business complexity

Companies face the following 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 data at rest 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 IBM Security Key Lifecycle Manager or other external key managers, which all provide similar capabilities (see Chapter 2, “External key managers” on page 13). These security lifecycle management software products help organizations efficiently deploy, back up, restore, and delete keys and certificates securely and consistently.

Important (more encryption): The DS8000 provides disk-based encryption for data that is at rest on disk. It also allows encryption of data that is transmitted to the cloud when the TCT function is used, and encryption when connecting to an IBM Z z15 host. If encryption over the network is required, more encryption services must be investigated and deployed for other hosts connectivity or Copy Services traffic.

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

For more information, see IBM Security.

1.1.3 Need for Transparent Cloud Tiering encryption

TCT for IBM DS8000 was introduced to help customers use IBM Z resources more efficiently. Integration with IBM z/OS through DFSMShsm allows clients to reduce CPU usage by eliminating constraints that are tied to original data migration methodologies.

TCT enables direct data movement from IBM DS8000 to cloud object storage, without the need for data to go through the host. DFSMS communicates with DS8000 through a REST API interface. It issues commands for the DS8000 to move the data directly to and from a public, private, or hybrid cloud.

For more information about TCT, see *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381.

Having a storage cloud that uses object storage has several benefits. A storage cloud significantly reduces the complexity of storage systems by simplifying data scaling within a single namespace. The use of high-density, low-cost commodity hardware turns storage clouds into a scalable, cost-efficient storage option.

When data is offloaded to a storage cloud in its original, unencrypted condition, unauthorized access is *not* prevented.

TCT encryption now changes this situation and ensures that critical mainframe data is encrypted while it is transferred over the network. It uses the DS8000 internal IBM POWER® servers hardware acceleration with 256-bit Advanced Encryption Standard (AES) encryption at full speed, and I/O performance is not affected. The data remains encrypted in the cloud storage and is decrypted when it is transferred back to the DS8000.

If the data set is already encrypted by data set-level encryption, DFSMS informs the DS8000, and TCT encryption avoids double encrypting the data.

You can use TCT encryption to offload and decrypt data with any DS8000 storage system and use the same key servers as the system that first encrypted the data. The data remains encrypted, even when it lands on the object storage. Without a decryption key, the data object is unreadable.

In IBM HyperSwap® High Availability and Disaster Recovery scenarios with Metro Mirror and Global Mirror, all DS8000s must be connected to the same cloud and all must be configured for TCT encryption. The certificates of every DS8000 must be added to the encryption group in IBM Security Key Lifecycle Manager during setup. This setup is required so that any DS8000 in the encryption group can decrypt data that is migrated from any other DS8000.

TCT encryption does not require a specific license. It can be used along with or independently from DAR encryption.

Note: Starting with Release 8.5, DS8880 supports TCT encryption. Previous models, such as DS8100 to DS8870, do not support TCT encryption.

1.1.4 Need for IBM Fibre Channel Endpoint Security

IBM Fibre Channel Endpoint Security is designed to protect data that is transferred over Fibre Channel (FC) storage area networks (SANs). It consists of the following components:

- Link authentication
- Encryption of data in flight (EDIF)

Today, data that is stored and processed in IT systems is one of the core assets of most enterprises or organizations. Losing or exposing data often results in high costs or even irreparable damage. In addition, more regulatory requirements are introduced, which forces organizations to protect the data they store and process and induces severe penalties if requirements are not met or sensitive data is lost or exposed. Thus, organizations are experiencing increased pressure from internal and external sources to protect and govern data.

Data protection features the following main aspects:

- Protection against loss

  Although losing access to data can severely affect an organization's ability to function, such a loss generally has a limited effect on third parties. In the past, most of the efforts of data protection focused on this aspect. Hardware redundancy, back up and restore processes, or disaster recovery solutions are examples of methods that are used here.

- Protection against unauthorized access and abuse

  Losing control of data affects the storing and processing capabilities of an organization and other organizations or persons with which it is interacting. This aspect is gaining significance in recent years because data breaches and abuse are reported frequently. Here, the most effective methods of protection are access control and encryption.

With the IBM Z z14, IBM introduced the concept of Pervasive Encryption. IBM Z clients should no longer be required to put excessive effort into planning, implementing, and maintaining effective access control and encryption of their data. Pervasive encryption provides the means to encrypt all data at all levels and in all components of the IT infrastructure, without affecting the operation or requiring changes to applications.

Figure 1-2 shows a graphical representation of the layers where encryption can occur.

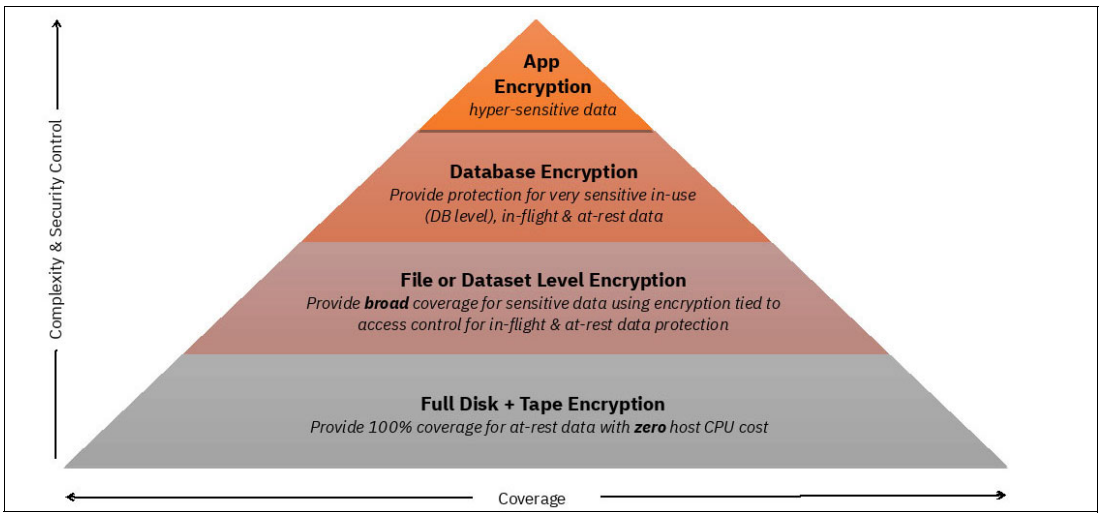


Figure 1-2 The IBM Pervasive Encryption Pyramid

The width of each layer represents the coverage that can be achieved that is related to overall protection. The vertical positions of the layers represent the granularity of control, but also the complexity of implementation and management. Starting with the top layer, we briefly explain each layer:

- Application encryption provides encryption and data protection by each individual application. It is highly granular and specific, but also requires the highest efforts because each application needs the necessary encryption capabilities and must be managed individually. It can provide protection of highly sensitive application data, which is not covered by any of the lower levels or if their protection is not sufficient.
- Database encryption provides the capability to protect key database files and database backup images from inappropriate access. Therefore, it is less granular and easier to manage, but covers only a certain subset of data.
- File or data set encryption provides broad coverage for sensitive data by using encryption that is tightly integrated with the operating system and managed by policies. It is not apparent to applications and allows for separation of duties within an organization. Security administration can be performed independently of application, database, or storage administration.
- Disk and tape encryption provide coverage for data-at-rest at the storage infrastructure level. It is an “all or nothing” solution and encrypts DAR within a storage controller without differentiating type, sensitivity, or importance of data. Therefore, it requires the least organizational effort of all layers with the broadest coverage. It protects against intrusion, tampering, or removal of physical infrastructure.

For more information about Pervasive Encryption, see *Getting Started with z/OS Data Set Encryption*, SG24-8410 and *Getting Started with z/OS Data Set Encryption*, SG24-8410.

For the upper three levels that are shown in the pyramid in Figure 1-2 on page 5, data is encrypted on the host side. Therefore, it is protected at-rest on external storage media in-flight while being read or written. With conventional disk encryption, data is unprotected if it is outside of the respective storage system.

IBM Fibre Channel Endpoint Security adds the protection of data in-flight between the IBM Z and the IBM DS8900 storage system, controlling access and encrypting data that is transferred over a SAN.

Note: Only data that is transferred between an IBM Z (z15) and IBM DS8900F storage systems can be protected with IBM Fibre Channel Endpoint Security.

At the time of this writing, data that is in-flight is not protected in following instances:

- On PPRC replication links between DS8000 storage systems
- Between an IBM Z system and virtual or physical tape devices

For more information about the IBM Fibre Channel Endpoint Security feature, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

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 US 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 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 1-3 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 make symmetric encryption an ideal candidate for encrypting large amounts of data.

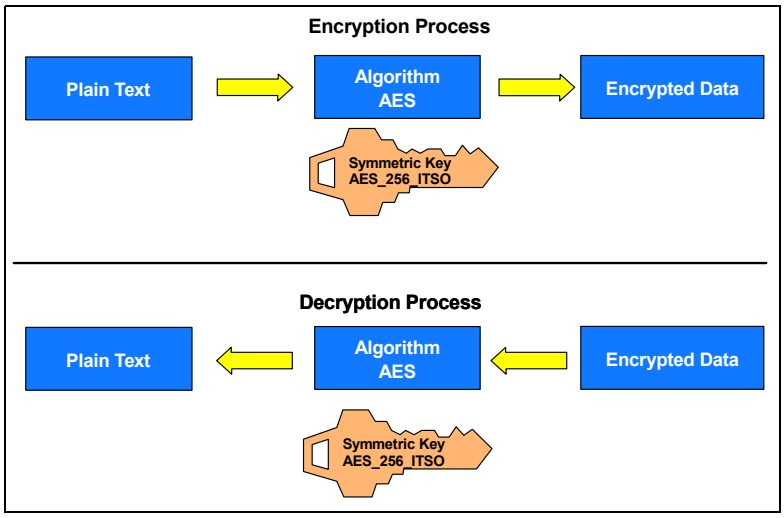


Figure 1-3 Symmetric key encryption

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) and Transport Layer Security (TLS).

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 10. 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 and the TCT encryption solution uses only the asymmetric RSA algorithm to encrypt symmetric AES keys that are used for data encryption.

Figure 1-4 shows an encryption and decryption data path when using public key encryption.

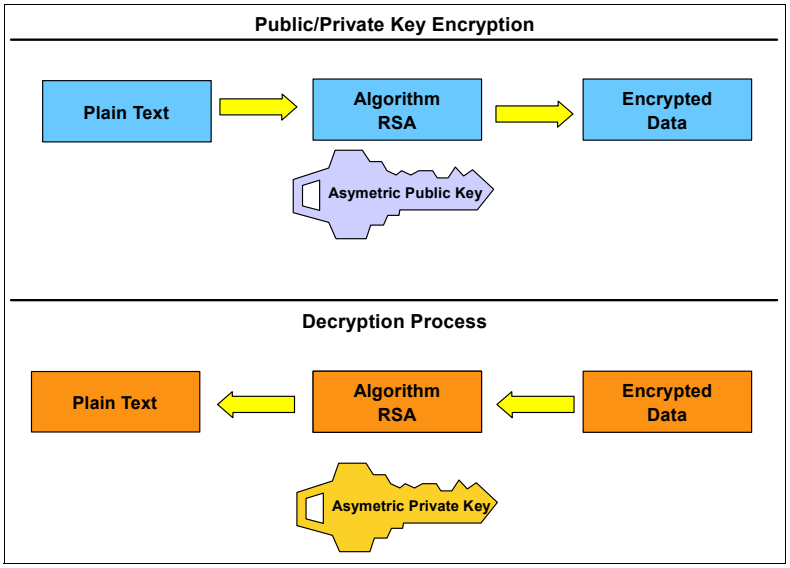


Figure 1-4 Public/private key encryption

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

Digital signature

You can use public/private key pairs to protect the content of a message and to digitally sign 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 1-5 shows how the digital signature is used in the communication between the DS8000 and a key server, such as IBM Security Key Lifecycle Manager or other external key managers, 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.

Note: IBM Security Key Lifecycle Manager V4.0 and later or other external key managers is a requirement if you want to comply with the NIST Special Publication (SP) 800-131a. For more information, see Chapter 2, "External key managers" on page 13.

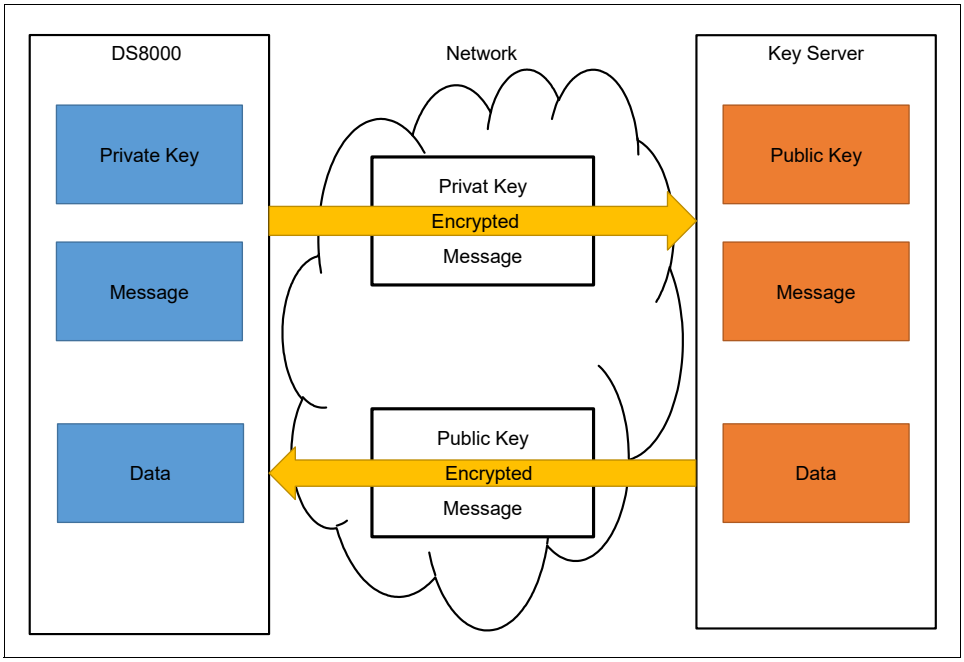


Figure 1-5 Identity verification by using public/private key encryption

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 it the Gen 1 certificate.

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

DS8900F Release 9.0 and later now offers 128 to 192-bit security strength, which is the Gen 3 certificate. The DS8900F includes an active Gen 2 certificate and a Gen 3 certificate that is dormant.

Both asymmetric and symmetric key encryption schemes are powerful ways to protect and secure data. Sections 3.1.1, "Key management for IBM Proprietary Protocol with IBM Security Key Lifecycle Manager" on page 31 or see 3.6, "DS8000 endpoint encryption key management using KMIP" on page 56 give details about their use with the IBM DS8000 series. Both provide an 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 (DK) to encrypt and decrypt data. They do not transfer this symmetric DK in the clear, but use public/private key encryption to encrypt the DK. The recipient can decrypt the encrypted DK and use the DK 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 and the TCT encryption solution uses a symmetric AES DK to encrypt and decrypt data. This DK is protected by the asymmetric RSA algorithm and is not available in plain text when the storage device communicates with the IBM Security Key Lifecycle Manager or any other third-party key server, such as Gemalto SafeNet KeySecure (KS). For more information, see Chapter 3, "IBM DS8000 encryption mechanisms" on page 27.

1.2.4 Communication protocols: IBM Proprietary Protocol, SSL/TLS 1.2, and Key Management Interoperability Protocol

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

IBM Proprietary Protocol

The DS8000 can support IBM Proprietary Protocol while it communicates with the IBM Security Key Lifecycle Manager. The IBM Proprietary Protocol protocol was first developed for tape drives. The DS8000 incorporated IBM Proprietary Protocol into its earliest encryption versions and continues to support it. The IBM Proprietary Protocol is wrapped with TLS when it communicates with the IBM Security Key Lifecycle Manager, and uses the default port 441. IBM Proprietary Protocol is available for DAR encryption only.

Important: TCT encryption and IBM Fibre Channel Endpoint Security do *not* support IBM Proprietary Protocol.

Secure Sockets Layer/Transport Layer Security 1.2

For the United States federal government, a minimum security strength of 80 bits was the recommendation until 2010. This requirement was achieved by IBM Proprietary Protocol. 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. IBM Proprietary Protocol is wrapped into SSL secured by TLS when communicating with the IBM Security Key Lifecycle Manager by using default port 441 and enabling NIST mode.

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

Key Management Interoperability Protocol

Key Management Interoperability Protocol (KMIP) is an industry standard that aims 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, DS8000 from Release 8.5 provides customers with more flexibility and choice in key management. DS8000 customers now can take advantage of encryption if KMIP is a requirement in their infrastructure. KMIP is supported by IBM Security Key Lifecycle Manager and other external key managers, as described in Chapter 2, “External key managers” on page 13.

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 DKs are encrypted with a wrapping key (another key to encrypt/decrypt the DKs). This wrapped key method is used with the DS8000 by separating the storage of a wrapped DK 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 DAR encryption RK to provide access to data if none of the key servers are available. To prevent one person from gaining access to the data, the handling of an RK requires two people with separate roles: The Security Administrator and the Storage Administrator. It is also possible to disable the RK, but it is done 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. Generally speaking, even when you are using a key server, at least one encryption key or RK must be maintained manually. The encryption key might be an overall key that manages access to all other encryption keys or a key that encrypts the data that is used by the key server.

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 3, "IBM DS8000 encryption mechanisms" on page 27.

2

External key managers

In an enterprise, many symmetric keys, asymmetric keys, and certificates can exist. All these keys and certificates must be managed. Key management is handled by an external key manager.

This chapter includes the following topics:

- 2.1, “External key managers for IBM DS8000 systems” on page 14
- 2.2, “IBM Security Key Lifecycle Manager” on page 14
- 2.3, “IBM Security Key Lifecycle Manager for z/OS” on page 18
- 2.4, “Gemalto SafeNet KeySecure” on page 24
- 2.5, “Thales Vormetric Data Security Manager” on page 25
- 2.6, “Thales CipherTrust Manager” on page 25

© Copyright IBM Corp. 2009, 2021. All rights reserved.

2.1 External key managers for IBM DS8000 systems

External key managers provide a centralized encryption and key management solution to minimize the risk of exposure and reduce operational costs.

> **Important:** For IBM Fibre Channel Endpoint Security, IBM Security Key Lifecycle Manager V4.0.0.2 is recommended.

The DS8000 ranks must all be either encrypted or non-encrypted for data at rest (DAR) encryption. An environment verification process or solution assurance must be completed to ensure that best practices regarding the configuration of the encryption solution are taken. This verification can be requested from IBM Lab Services or completed by the client's staff, but it is a prerequisite to activate the encryption solution.

Table 2-1 show a comparison of various external key managers and their support for DS8000 features.

Table 2-1 External key manager comparison

Key Managers	Data at rest over IBM Proprietary Protocol	Data at rest over Key Management Interoperability Protocol	Transparent Cloud Tiering	IBM Fibre Channel Endpoint Security
IBM Security Key Lifecycle Manager	x	x	x	x
IBM Security Key Lifecycle Manager for z/OS	x			
Safenet Gemalto KS		x	x	
Thales CipherTrust Manager		x	x	
Thales Vormetric DSM		x	x	

2.2 IBM Security Key Lifecycle Manager

The IBM Security Key Lifecycle Manager provides key storage, key serving, and key lifecycle management for storage devices from IBM and other vendors. For more information about supported products, see IBM Security Key Lifecycle Manager V4.0.0.

The focus of this publication is IBM key server interoperability with the DS8000. The DS8000 supports data encryption with the *Full Disk Encryption (FDE)* feature. With Version 8.5, it supports Transparent Cloud Tiering (TCT) encryption. IBM Fibre Channel Endpoint Security is supported by Version 9.0 and later.

The FDE drives can encrypt and decrypt at interface speeds, so there is no impact on performance. TCT encryption uses IBM POWER9™ hardware acceleration with 256-bit Advanced Encryption Standard (AES) encryption at full speed with no impact on performance for encrypting the data before it is transferred over the network.

All drives in the DS8880 and DS8900F are encryption-capable by default. To use the encryption feature, the DS8000 must be configured to communicate with the IBM Security Key Lifecycle Manager.

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

Note: At the time of writing, IBM Security Key Lifecycle Manager is being rebranded as IBM Security Guardium® Key Lifecycle Manager (Version 4.1). For more information, see IBM Security Guardium Key Lifecycle Manager.

2.2.1 IBM Security Key Lifecycle Manager V4.0

The following functions became available with IBM Security Key Lifecycle Manager V4.0:

- Installation, upgrade, and migration enhancements

  IBM Security Key Lifecycle Manager processes now run under a non-administrator or non-root user account even when you install the product under an administrator or root user account.

- New REST-based key management and serving

  Cloud applications or clients that need to use keys and other cryptographic objects from IBM Security Key Lifecycle Manager can now use REST APIs to get the key.

- Improved replication performance

  IBM Security Key Lifecycle Manager supports incremental replication. When the frequency of cryptographic object generation is high, you can use incremental replication so that the clone servers contain almost up-to-date data.

- Enhanced Multi-Master:
  - You can start, stop, and restart a Multi-Master cluster by using the GUI, REST interface, or scripts.
  - Data availability is enhanced in a Multi-Master cluster with support for new high availability disaster recovery (HADR) takeover scenarios.
  - The capability of the **Test connection** button on the "Add master" page is enhanced. When you click **Test connection**, IBM Security Key Lifecycle Manager checks the prerequisites to add a master server, and if there is an error, it displays the relevant message. Also, the button is renamed **Check prerequisites**.

  Note: A Multi-Master configuration is mandatory if you use IBM Security Key Lifecycle Manager for TCT encryption or IBM Fibre Channel Endpoint Security. You can also use Multi-Master for DAR encryption.

- GUI enhancements:
  - You can use the following options in the IBM Security Key Lifecycle Manager user (for example, `sklmadmin`) menu to change the corresponding user passwords: **Change WebSphere Application Server Password**, and **Change Database Password**.
  - You can now upload and download files (for example, certificates, keys, and backup files) in the IBM Security Key Lifecycle Manager server from the GUI.
  - The Replication section on the Welcome page is enhanced to display the status of the last run of the replication process. A relevant message is displayed if an error occurs. You can review the error message to identify the problem and take a corrective action.
  - The **Clients and Groups** option on the Welcome page is now renamed **Clients**.

- Simplified process to update the IBM Db2® password in IBM Security Key Lifecycle Manager.

  The procedures to update the Db2 password for a stand-alone IBM Security Key Lifecycle Manager server and a Multi-Master cluster are now simplified.

- Interactive and easy-to-use REST API console.

  The Swagger UI is now integrated with IBM Security Key Lifecycle Manager, and you can use it to call any REST API.

- 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.
- There is a GUI, RESTful interface, 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 database information.
- Migration of the IBM Security Key Lifecycle Manager Version 1.0, 2.x, 3.x, and IBM Encryption Key Manager V2.1 component during installation by using inline migration or cross migration.
- 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 active files and data across operating systems. This replication enables cloning of IBM Security Key Lifecycle Manager environments on multiple servers independently of operating systems and the directory structure of the server.
- Multi-Master configuration for IBM Security Key Lifecycle Manager: You can set up IBM Security Key Lifecycle Manager instances with a Multi-Master configuration to achieve continuous availability of data across multiple IBM Security Key Lifecycle Manager deployment environments. The following IBM Security Key Lifecycle Manager features are related to the DS8000:
  - Key Management Interoperability Protocol (KMIP) Support for DS8000 DAR encryption, TCT encryption, and IBM Fibre Channel Endpoint Security.
  - Role-based access control (RBAC) that provides permissions to do tasks such as create, modify, and delete for specific device groups. Most permissions are associated with specific device groups.

Note: You must use the predefined device group “DS8000” for DAR encryption, “DS8000_TCT” for TCT encryption, “Peer_to_Peer” for IBM Fibre Channel Endpoint Security.

Important: DS8000 does not support communication with IBM Security Key Lifecycle Manager over IBM Proprietary Protocol by using custom device groups. KMIP is recommended for DS8000 systems communicating with IBM Security Key Lifecycle Manager Key Servers in a Multi-Master configuration. When using IBM Proprietary Protocol to communicate with IBM Security Key Lifecycle Manager Key Servers in a Multi-Master configuration, it is not possible for the DS8000 systems to detect automatically problems that are related to key redundancy, so you are responsible for determining when HADR synchronization is not functioning properly. Loss of data in the IBM Security Key Lifecycle Manager keystore can result in loss of DS8000 data.

For more information about the features of the IBM Security Key Lifecycle Manager and its predecessors, see IBM Security Key Lifecycle Manager V4.0.0.

2.2.2 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 keys, its alias, and metadata are stored in an 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 an entry that is in a pending state. It waits for the return of a certificate that was approved and certified by a certificate authority (CA).

Changing attributes: The compromised and information attributes of a key, regardless of its state, can be changed by using REST or 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 software provider. It can be used for all distributed operating systems.

2.2.3 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 or Syslog format by IBM Security Key Lifecycle Manager.
- Automated clone replication: A master IBM Security Key Lifecycle Manager server can be configured to replicate data to up to 20 clones. The replicated data includes keys and certificates that are stored in tables in an IBM Security Key Lifecycle Manager database, certificates in a truststore keystore, and a master key that is stored in Java keystore and IBM Security Key Lifecycle Manager configuration files.

Note: The replicated data is identical to the IBM Security Key Lifecycle Manager backup, except for the replication configuration file. During replication, these items are not backed up or passed to the clones.

- Master key in HSMs: IBM Security Key Lifecycle Manager can use an HSM to store the IBM Security Key Lifecycle Manager master key to protect all keys and certificates that are stored in the database. This option adds extra protection to the storage and use of the master key.

2.3 IBM Security Key Lifecycle Manager for z/OS

As an alternative to the IBM Security Key Lifecycle Manager V3.0 or later for open systems, you can use the IBM Security Key Lifecycle Manager for z/OS product. Previous names for this product were IBM Encryption Key Manager and IBM Tivoli® Key Lifecycle Manager for the z/OS.

IBM Security Key Lifecycle Manager for z/OS also helps when you generate, protect, store, and maintain encryption keys that are used as follows:

- To encrypt information that is being written to devices
- To 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.

Attention: Do not confuse IBM Security Key Lifecycle Manager for z/OS with IBM Security Key Lifecycle Manager V4.0 for Open Systems. IBM Security Key Lifecycle Manager for z/OS supports DAR encryption key management *only*. It does not support TCT encryption and IBM Fibre Channel Endpoint Security.

However, you can install IBM Security Key Lifecycle Manager V4.0 in IBM Z Linux.

2.3.1 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.

2.3.2 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 Rivest-Shamir-Adleman (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.

2.3.3 Preventing a deadlock situation

If you use IBM Security Key Lifecycle Manager for z/OS as a key manager for DS8000 DAR encryption, you can run into a deadlock situation if you use key servers that are run only on encrypted DS8000 disk storage systems.

When all z/OS logical partitions (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 is 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 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.

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

Install IBM Security Key Lifecycle Manager for z/OS as described in *Program Directory for IBM Security Key Lifecycle Manager for z/OS V1.1.0*.

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 this website.

Here are the highlights of the installation steps:

1. Install the Java SDK.

   A recent version, IBM 64-bit SDK for z/OS Java Technology Edition Version 8 is preferred. It already should be installed in your z/OS environment.

   Verify that the correct version of Java is installed. First, make sure that the Java `bin` directory is in your PATH by running the **`USSexport`** command:

   ```
   export JAVA_HOME=/usr/lpp/java84
   export PATH="$PATH":"${JAVA_HOME}"/bin
   ```

   Replace the path with the path where your Java SDK was installed. Then, run the **`java -version`** command from the UNIX console under z/OS.

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 IBM Java Cryptography Extension CCA provider is installed in `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 step 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.
4. Set up a user to run IBM Security Key Lifecycle Manager for z/OS.

   In our case, we used the ISKLMSRV user ID (UID). This UID must have an OMVS segment with a UID and group ID (GID) that is defined. The UID does not need to be zero; it can be any value. The home directory of this UID OMVS segment is where IBM Security Key Lifecycle Manager for z/OS is started. The UID must also run the standard shell at login (`/bin/sh`), and be connected to a default group that has a GID. You can allow IBM RACF® to automatically assign the UID or explicitly define the UID. The ISKLMSRV UID is a protected user. It cannot be used as a TSO UID.

Note: In OMVS, the configuration file permission is set so that only the owner can read or write the configuration file. If you log on and you are not the owner of the configuration file, you do not have permission to write to the configuration file. You might encounter an error similar to the following one:

```
- java.io.FileNotFoundException:
/u/isklmsrv/JAO/ISKLMConfig.properties.zos.JCECCARACFKS (EDC5111I Permission
denied.)
```

You might encounter the same error when stopping the server, running the refresh operation, or changing passwords. As a best practice, log on by using the UID with owner permissions.

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 (IBM Security Key Lifecycle Manager 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 21.
   e. `TransportListener.ssl.truststore.name`: Specify the path and file name of the keystore that is created in step 1 on page 21.
   f. `Admin.ssl.keystore.name`: Specify the path and file name of the keystore that is created in step 3 on page 22.
   g. `Admin.ssl.truststore.name`: Specify the path and file name of the keystore that is created in step 3 on page 22.
   h. `config.keystore.file`: Specify the path and file name of the keystore that is created in step 3 on page 22.
   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 extra 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 22.

– `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. Use option 6 from the ISPF primary screen to enter the following TSO commands:

```
SETROPTS GENERIC(STARTED)
RDEFINE STARTED ISKLM*.* STDATA(USER(ISKLMSRV) GROUP(STCGROUP) TRACE(YES))
SETROPTS CLASSACT(STARTED) SETROPTS RACLIST(STARTED)
SETROPTS RACLIST(STARTED) GENERIC(STARTED) REFRESH
```

To start IBM Security Key Lifecycle Manager, run the **`S ISKLM`** command from SDSF or any z/OS console (the output of the command is shown after the command):

```
S ISKLM
$HASP100 ISKLM ON STCINRDR
IEF695I START ISKLM WITH JOBNAME ISKLM IS ASSIGNED TO USER ISKLMSRV, GROUP SYS1
$HASP373 ISKLM      STARTED
ISKLM console interaction is now available. 546
```

To submit commands to the IBM Security Key Lifecycle Manager from the console, run the following command:

```
F ISKLM,APPL='ISKLM command'
```

To stop the IBM Security Key Lifecycle Manager, run the following command:

```
P ISKLM
Loaded drive keystore successfully
Starting the Encryption Key Manager 2.0-20070
Processing Arguments
Processing
Server is started
Server is running. TCP port: 3801, SSL port: 4
```

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
- IBM Security Lifecycle Manager for z/OS 1.1.0

2.4 Gemalto SafeNet KeySecure

Gemalto SafeNet KS is a third-party, centralized key management platform. It offers an alternative to IBM Security Key Lifecycle Manager for clients and is fully supported by DS8880 Release 8.5 and later for DAR encryption and TCT encryption.

Gemalto SafeNet KS is provided 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 KS supports 128-bit encryption and provides a GUI named "Gemalto SafeNet KS Management Console" and an SSH CLI.

Gemalto SafeNet KS can manage up to 1,000,000 keys and 1,000 devices, and it supports HSM for storing the master key.

For more information about Gemalto SafeNet KS, see Cloud Protection and Licensing Solutions.

2.5 Thales Vormetric Data Security Manager

The Thales Vormetric DSM is another external key manager that is supported for DS8000 DAR encryption and TCT encryption to cloud storage.

For organizations that opt for the Thales Vormetric DSM platform products, Vormetric DSM is the central management point of the platform. DSM creates, stores, and manages the encryption keys that protect data, and it also enables organizations to manage every aspect of their data security platform implementation. Thales Vormetric DSM allows administrators to specify data access policies, administer DSM users and logical domains, generate usage reports, register hosts, access security logs, and manage third-party keys and digital certificates.

For more information, see Vormetric Data Security Manager.

2.6 Thales CipherTrust Manager

Thales CipherTrust Manager (formerly known as Next Generation KS) is another external key manager that is supported for DS8000 DAR encryption and TCT encryption to cloud storage.

Thales CipherTrust Manager offers an industry-leading enterprise key management solution that organizations can use to centrally manage encryption keys, provide granular access control, and configure security policies. Thales CipherTrust Manager is the central management point for the Thales CipherTrust Manager data security platform. It manages key lifecycle tasks that include generation, rotation, destruction, import, and export, and it provides RBAC to keys and policies, supports robust auditing and reporting, and offers a developer-friendly REST API.

For more information, see CipherTrust Manager.

3

IBM DS8000 encryption mechanisms

This chapter provides information about the DS8000 disk encryption mechanisms.

This chapter includes the following topics:

- 3.1, “DS8000 data at rest disk encryption” on page 28
- 3.2, “Encryption deadlock” on page 43
- 3.3, “Working with a recovery key” on page 44
- 3.4, “Dual key server support (IBM Proprietary Protocol only)” on page 50
- 3.5, “DS8000 TCT encryption Key Management using KMIP” on page 52
- 3.6, “DS8000 endpoint encryption key management using KMIP” on page 56

© Copyright IBM Corp. 2009, 2021. All rights reserved.

3.1 DS8000 data at rest disk encryption

The DS8000 supports data encryption in systems that are equipped with Full Disk Encryption (FDE) capable hard disk drives (HDDs) and flash drives, such as in the High Performance Flash Enclosure (HPFE).

All drives (disk or flash) in the DS8880 or DS8900F systems are encryption-capable (DS8900F supports only flash drives). Those drives include encryption hardware and can perform symmetric encryption and decryption of data at full disk speed with no effect on performance.

The disk encryption hardware is used with various key managers.

All key managers support running Key Management Interoperability Protocol (KMIP) by using the direct key method to deliver keys to encrypting storage devices.

IBM Security Key Lifecycle Manager also supports running IBM Proprietary Protocol by using a wrapped key method. 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, the data on disk *cannot* be decrypted.

Cryptographically erased: If all copies of the decryption key are lost (whether intentionally or accidentally), 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. This issue is relevant for data at rest (DAR) and Transparent Cloud Tiering (TCT) encryption, but does not apply to IBM Fibre Channel Endpoint Security where the keys are used only to encrypt data while over the wire (data in flight).

For more information about encryption key management, see 3.1.1, “Key management for IBM Proprietary Protocol with IBM Security Key Lifecycle Manager” on page 31, and 3.6, “DS8000 endpoint encryption key management using KMIP” on page 56.

An encryption-capable DS8000 can be configured to enable or disable encryption for DAR for all data that is stored on client disks.

Attention: Enabling DAR encryption cryptographically erases all data on the disks. Therefore, encryption must be enabled directly at the beginning, not when data is 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. 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 Internet Protocol network, as shown in Figure 3-1.

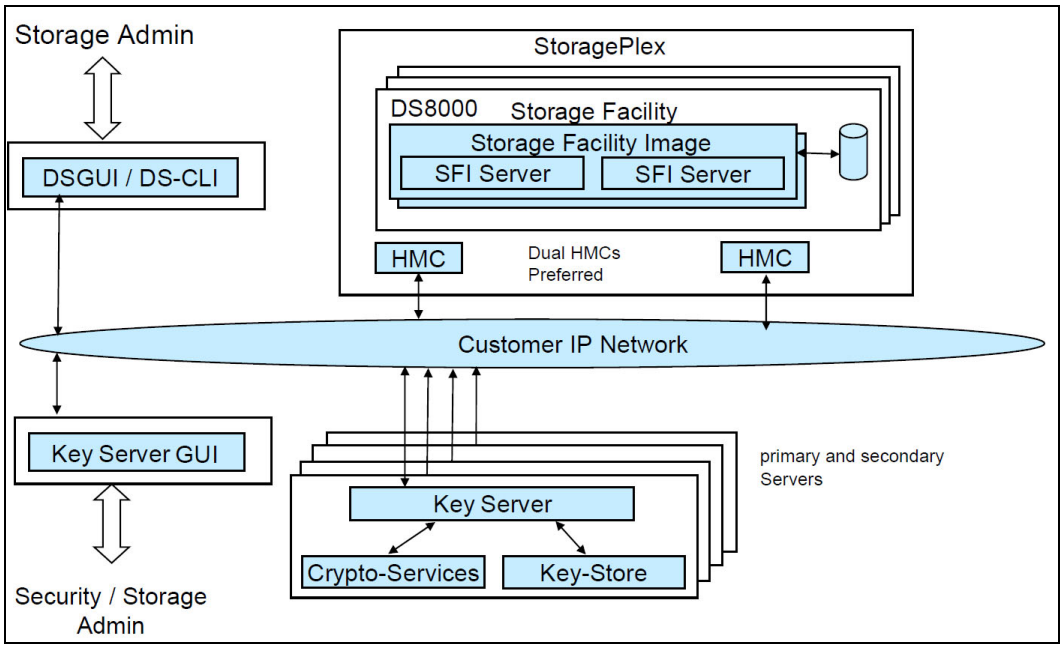


Figure 3-1 Connection between the DS8000 HMC key servers

Before explaining the various keys that are used by DS8000 and key servers 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 3-2 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 receiver reproduces the message hash in the following 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.

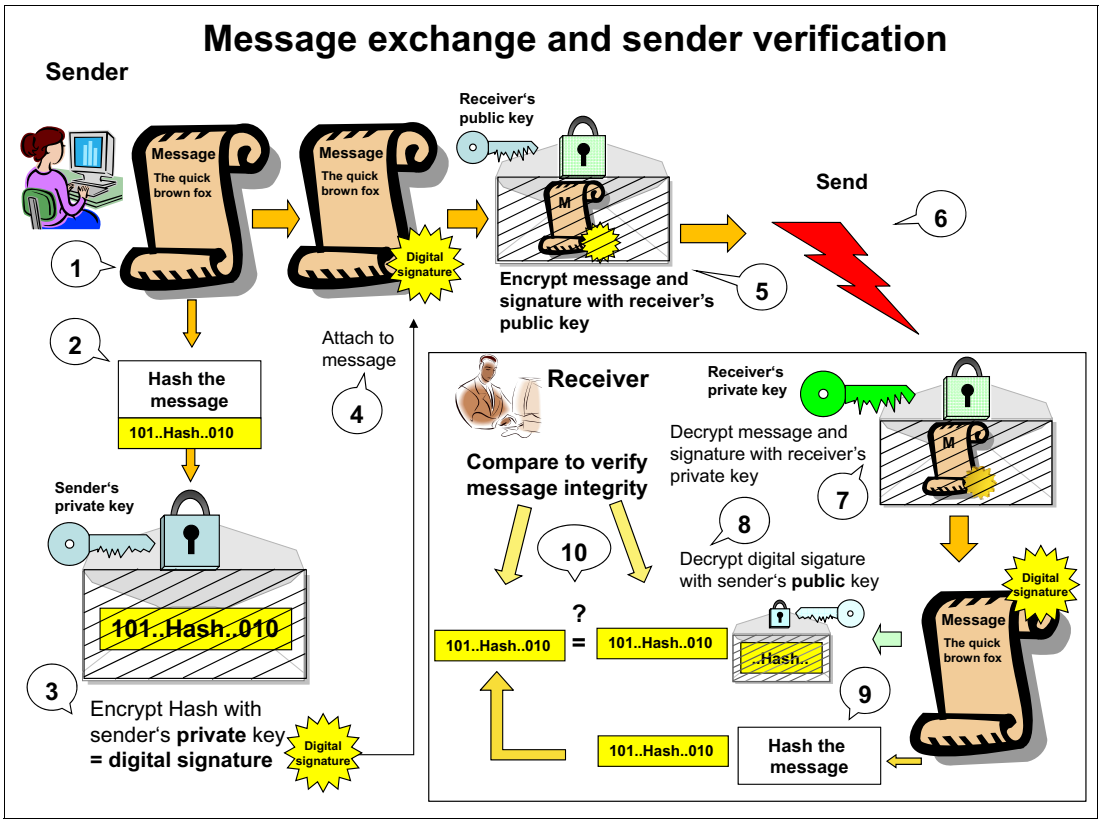


Figure 3-2 Authentication with digital signatures

3.1.1 Key management for IBM Proprietary Protocol with IBM Security Key Lifecycle Manager

In this section, we describe 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. This section describes the IBM Proprietary Protocol.

Important: Key negotiation and authentication between the IBM Security Key Lifecycle Manager and DS8000 take place at DS8000 power-on time only. Traffic does not increase 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 5.2, "WebSphere, Java, and IBM Security Key Lifecycle Manager hardening" on page 83.

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 3-3). 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 KEK-pair key is kept secret by IBM Security Key Lifecycle Manager.

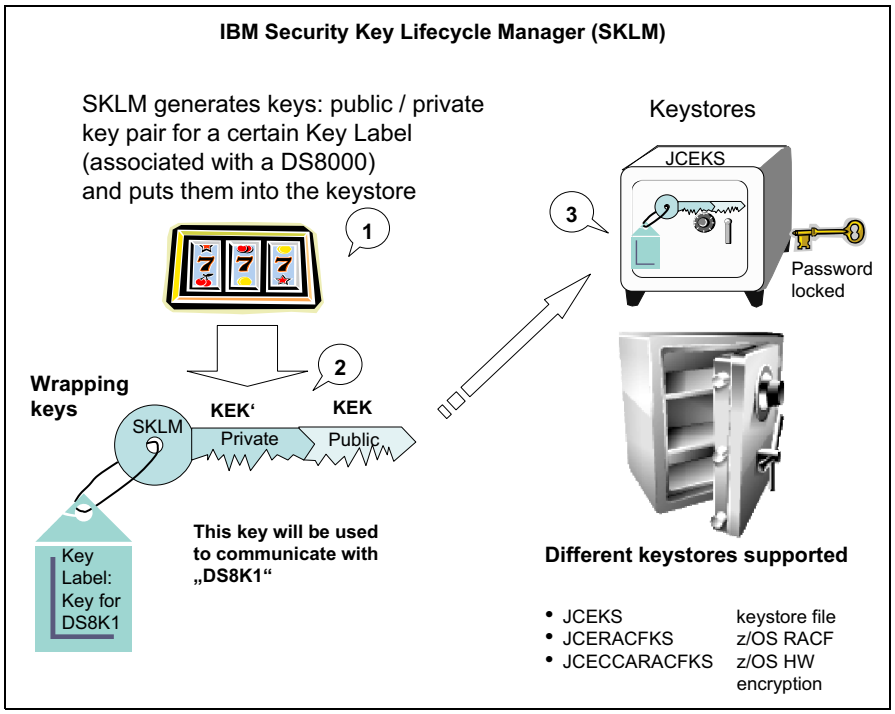


Figure 3-3 Configure IBM Security Key Lifecycle Manager key label

Rekey Data Key feature: With this feature, a user can change the DK labels (see 6.1, "Rekeying the data key for data at rest encryption" on page 214).

Now, the user (storage administrator) can use the DS8000 GUI or DS Command-Line Interface (DS CLI) to register the key server on the DS8000. Next, still using the DS8000 GUI or DS CLI, an encryption group is created. For more information, see “DS8000 enabling data at rest encryption” on page 163).

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

Note: The DS8000 must have separate encryption groups: one encryption group for DAR encryption, one for TCT, and one for IBM Fibre Channel Endpoint Security encryption. Only DAR supports IBM Proprietary Protocol.

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 3-4).

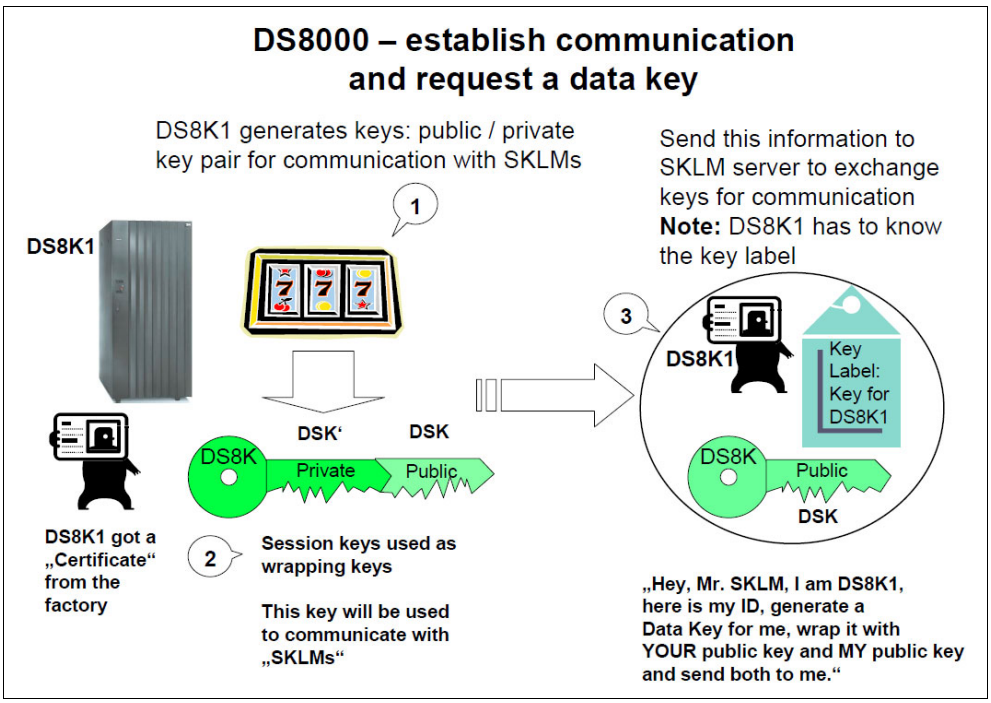


Figure 3-4 DS8000 creates session keys and requests a data key

Upon reception of these elements, IBM Security Key Lifecycle Manager carries out the following steps (see Figure 3-5 on page 33):

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 that is referred to as the externally encrypted data key (EEDK).

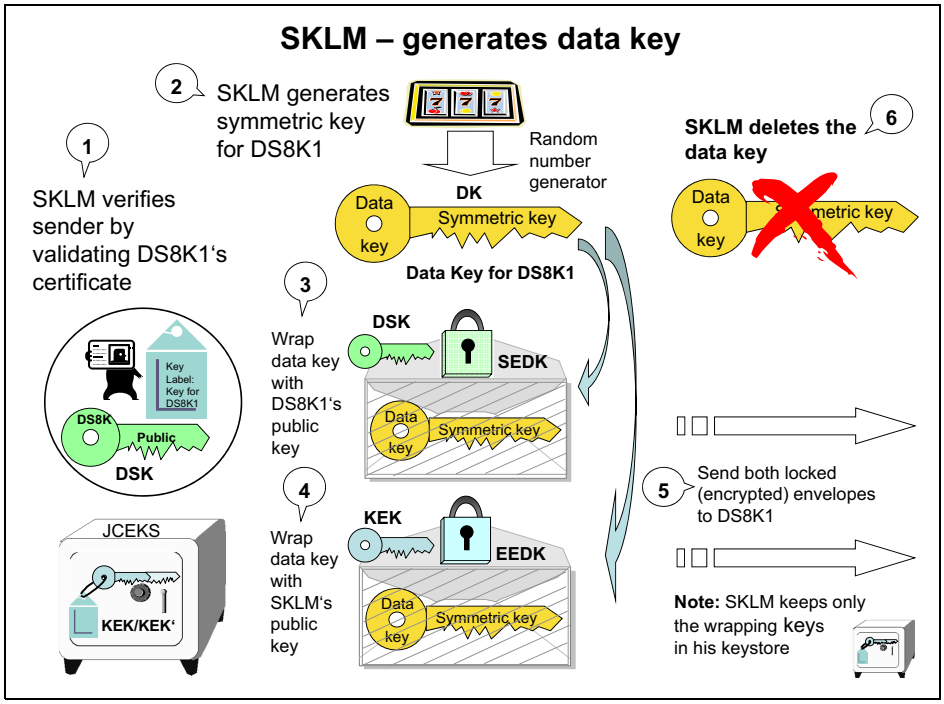


Figure 3-5 IBM Security Key Lifecycle Manager generates data key

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 3-6.

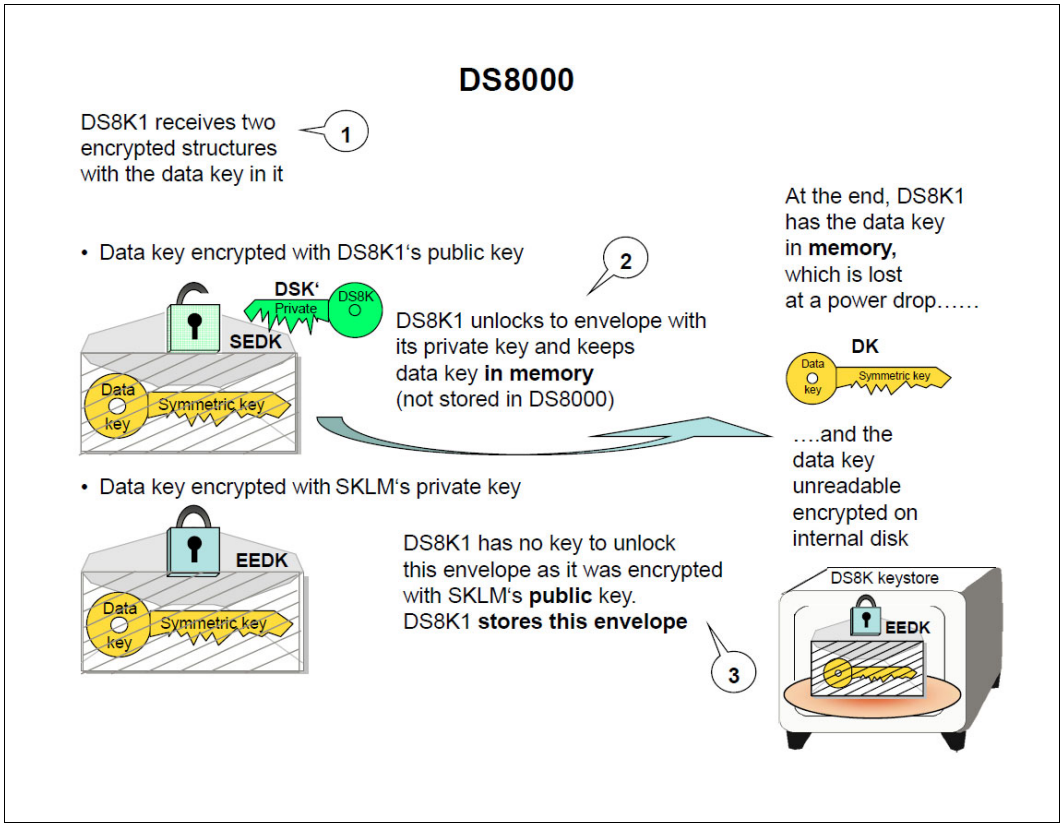


Figure 3-6 DS8000 unwraps data key and stores encrypted data key

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 3-7 on page 35.

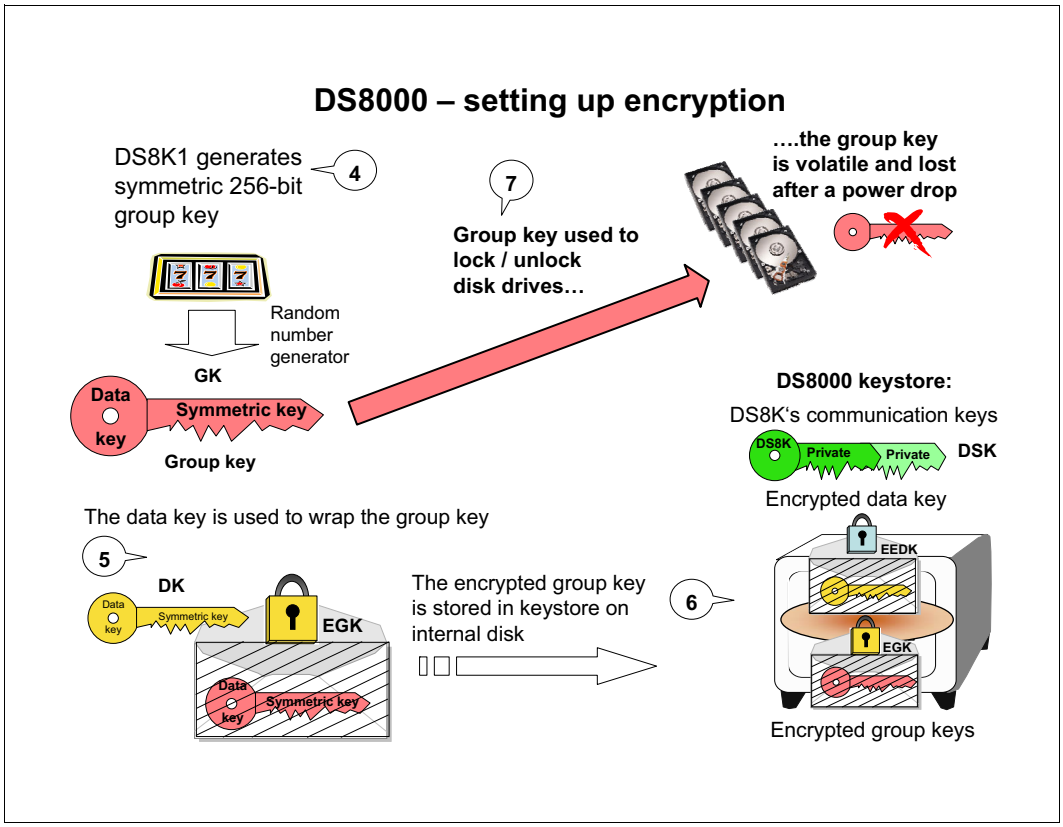


Figure 3-7 Set up encryption

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 3-8. 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, its encryption key is wrapped with the access credential. A hash of the access credential is also stored on the drive.

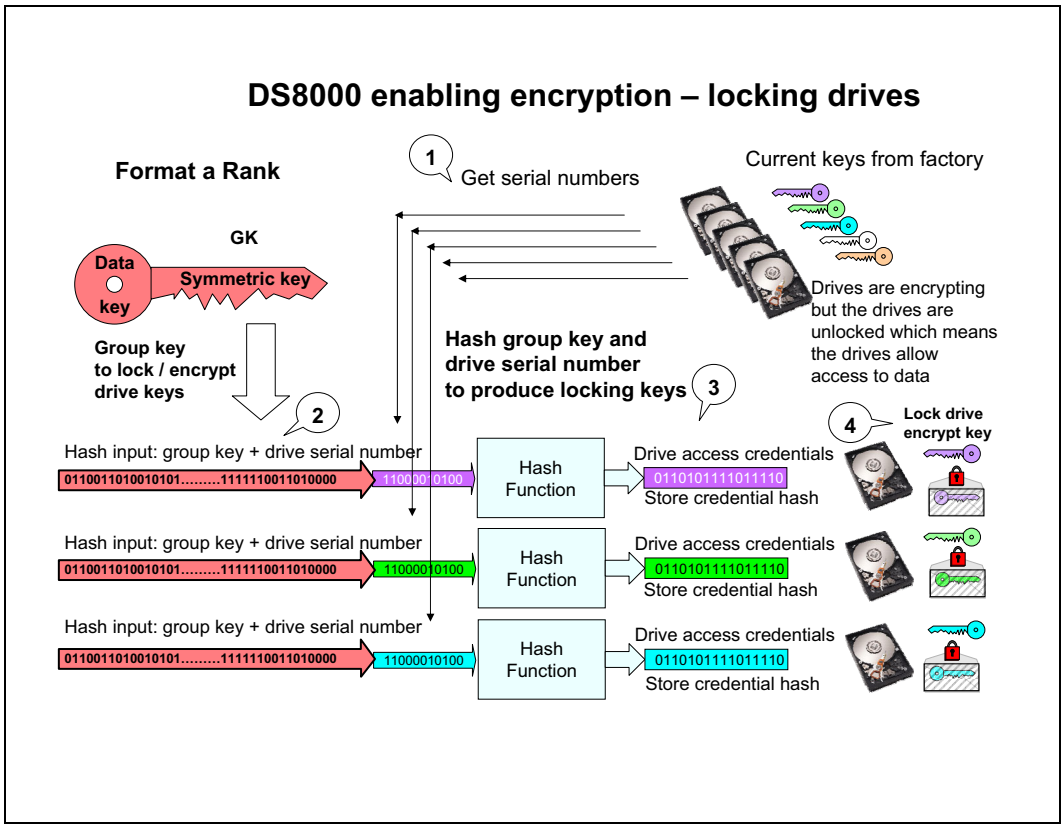


Figure 3-8 Set up encryption: Lock the drives

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 3-9 on page 37.

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 3-9 on page 37, 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 3-9. (The DS8000 first must 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).

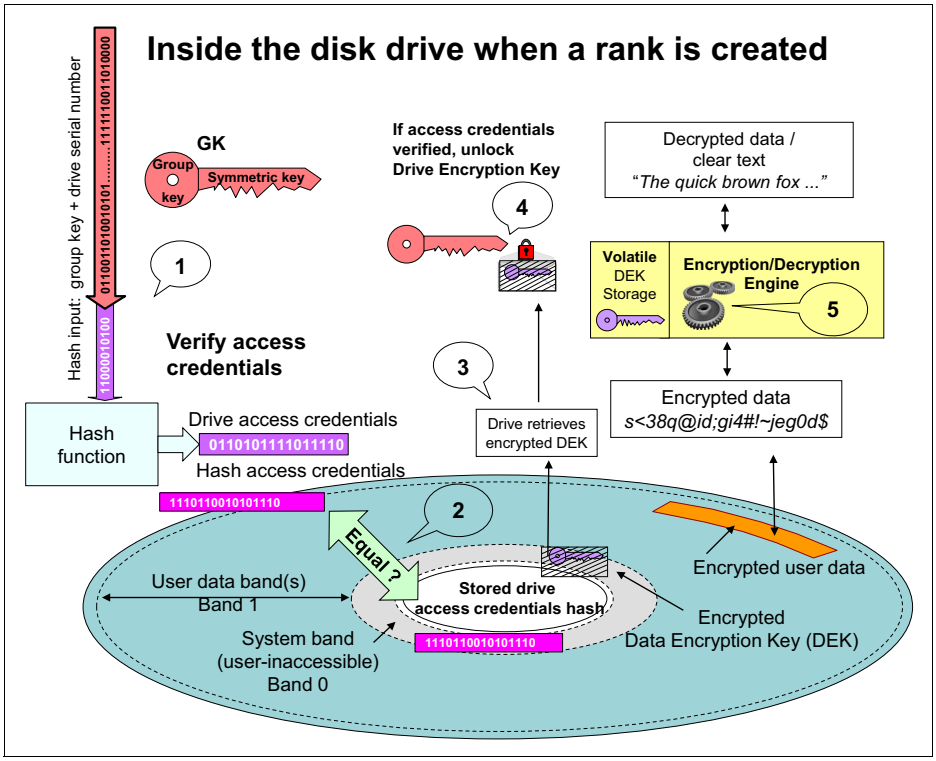


Figure 3-9 Unlock drives

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 action also happens when an encryption-disabled rank is deleted.

In a cryptographic erasure, a new DEK is generated in each disk drive. See Figure 3-10. 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.

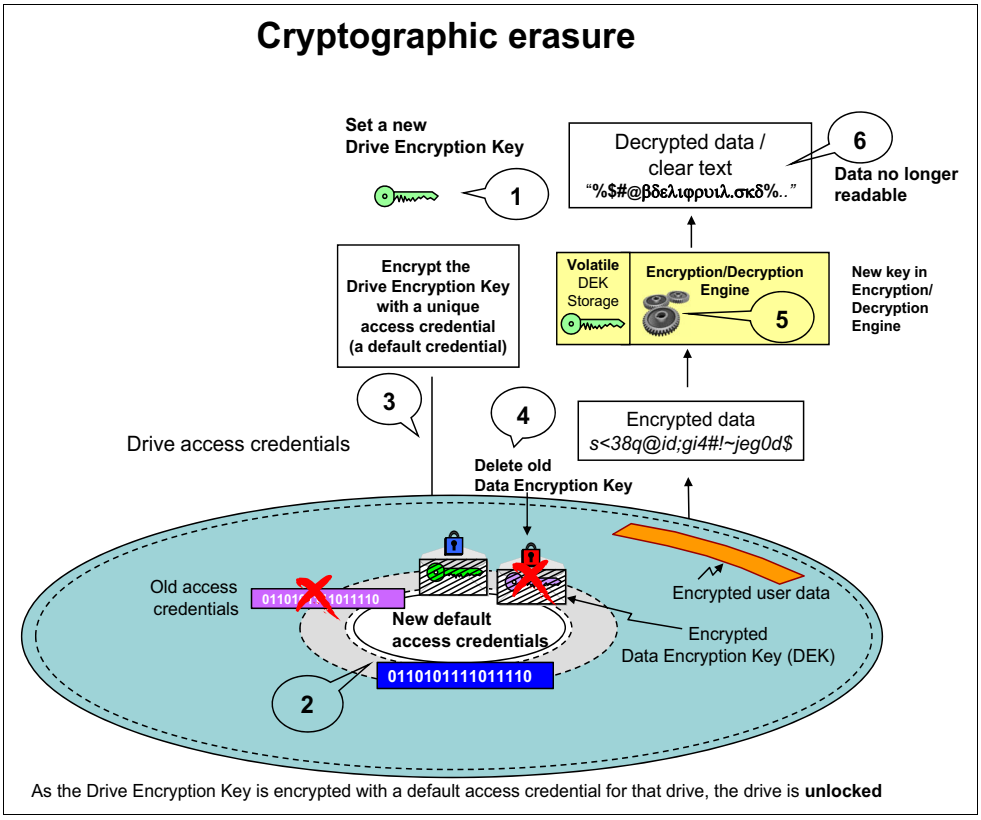


Figure 3-10 Cryptographic erasure

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 3-11. 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.

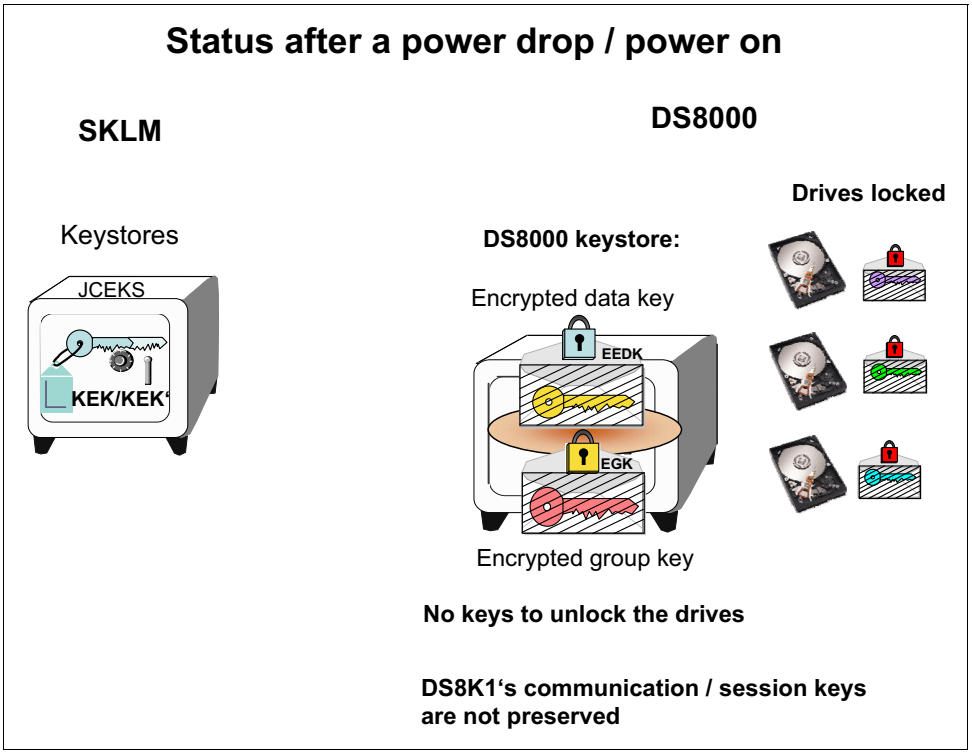


Figure 3-11 DS8000 status after a power-off/-on

The DS8000 must do the following steps (see Figure 3-12) 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.

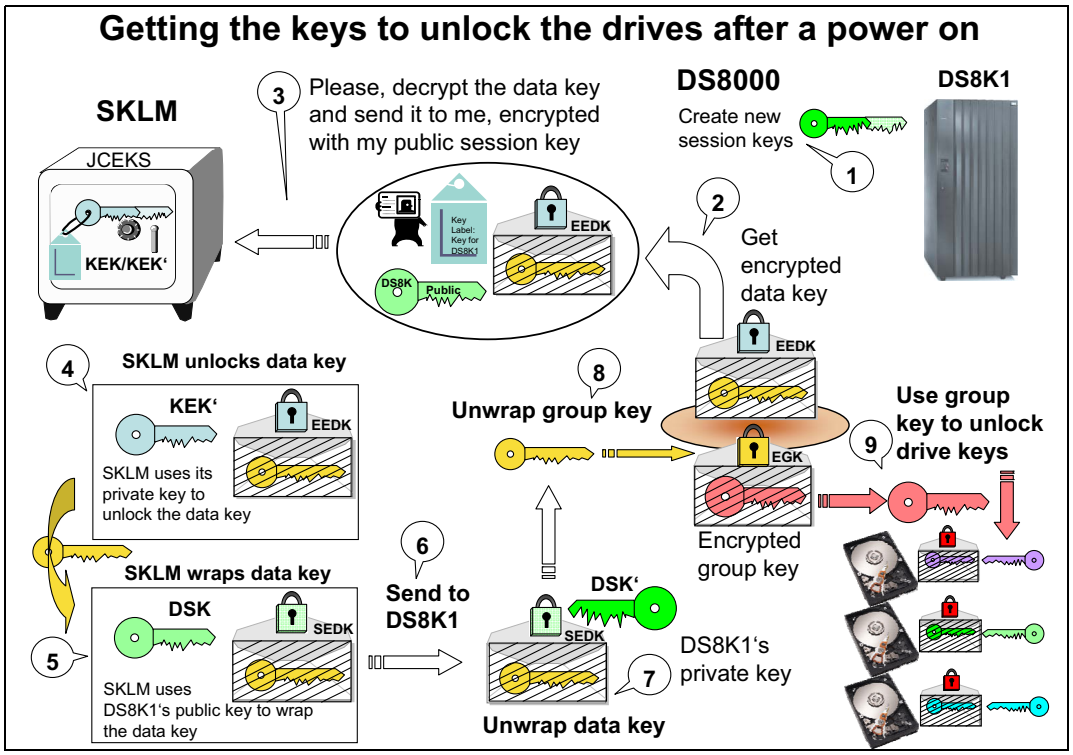


Figure 3-12 Steps to regain access to data after a power-on

3.1.2 Key management by using KMIP

In this section, we describe how the key servers manage and create encryption keys. These keys are used by the DS8000 during encryption group and rank creation and DS8000 power-on time.

> **Important:** Key negotiation and authentication between the key manager and DS8000 occur at DS8000 encryption configuration and power-on time only. Traffic use does not increase in an encrypted DS8000 at run time that is created by key negotiation.

The key manager server that uses KMIP uses the direct key method to serve keys to encryption-enabled DS8000 while IBM Security Key Lifecycle Manager is using IBM Proprietary Protocol 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 DK is created by the key server.

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

For more information about how to set up the IBM Security Key Lifecycle Manager with UID and SSL certificates for DK delivery, see “Creating an SSL/KMIP server certificate” on page 95.

For more information about how to set up the Gemalto SafeNet KeySecure (KS) servers with UID and SSL certificates for DK delivery, see 5.4.2, “Gemalto SafeNet KeySecure configuration” on page 117.

For more information about how to set up the Thales Vormetric Data Security Manager (DSM) server with UID and SSL certificates for DK delivery, see 5.4.3, “Configuring Thales Vormetric Data Security Manager” on page 133.

For more information about how to set up the Thales CipherTrust Manager server with UID and SSL certificates for DK delivery, see 5.4.4, “Configuring Thales CipherTrust Manager” on page 140.

Now, the user (Storage Administrator) can use the DS8000 GUI or DS CLI to register the key server on the DS8000. For more information about how to perform this action, see 5.5.3, “DS8000 configuration for data at rest encryption in DS GUI” on page 158.

Next, an encryption group must be created by using the DS8000 GUI or DS CLI. As part of creating the encryption group, you must specify the encryption protocol KMIP. For more information about how to perform this action, see 5.5.3, “DS8000 configuration for data at rest encryption in DS GUI” on page 158.

> **Note:** The DS8000 supports only one DAR encryption group.

The encryption group creation process features the following steps (see Figure 3-13):

1. A user requests that an encryption group is created.
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 universally unique 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 process.

> **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, which is 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, UUID, and 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 verification is successful.

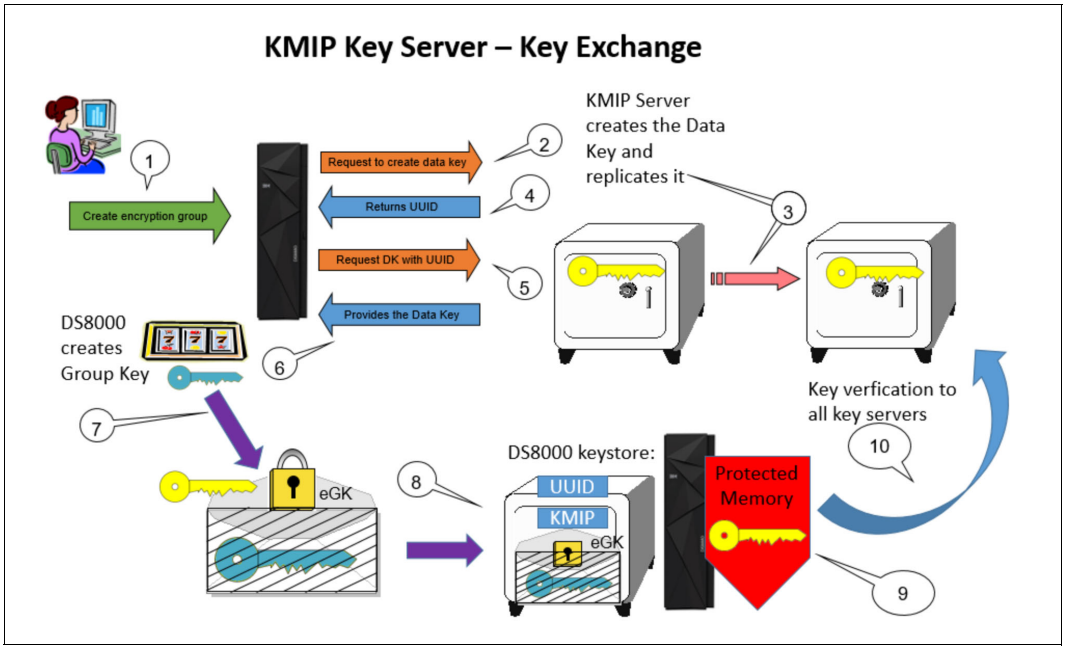


Figure 3-13 KMIP key server key exchange

After the encryption GK is created, the remaining drive encryption steps are the same as the steps that are used with IBM Security Key Lifecycle Manager using IBM Proprietary Protocol because the internal DS8000 encryption mechanism did not change. For more information, see 3.1.1, “Key management for IBM Proprietary Protocol with IBM Security Key Lifecycle Manager” on page 31. 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 cannot access all of these keys. Instead, it must first get the unlock DK from KMIP key server.

Because 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.

3.2 Encryption deadlock

The key server platform provides the operating environment for the key server application to run in, access its keystore on persistent storage, and 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 temporarily, or worse, permanently. This situation is referred to as *encryption deadlock*.

Important (encryption deadlock): Any data that is required to make the 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.

A temporary encryption deadlock and a permanent encryption deadlock feature the following differences:

- Temporary encryption deadlock

  The temporary encryption deadlock indicates a situation where the DS8000 cannot access its disk devices because the key 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, consider 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 move 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 key 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 4, “Planning and guidelines for IBM DS8000 encryption” on page 63, and Chapter 5, “IBM DS8000 encryption implementation” on page 73.

3.3 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), use the DS8000 to create an RK, a Security Administrator can unlock a DS8000 without the involvement of a key server. It is also possible to *disable* RK management.

Restriction: An RK can be created for DAR encryption *only*. It is not supported for TCT or IBM Fibre Channel Endpoint Security.

Important (creating or disabling an RK): An RK can be created during the DAR encryption enablement process only. You cannot create an RK when a DS8000 is 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 DAR 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 must 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.

3.3.1 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 3-14.

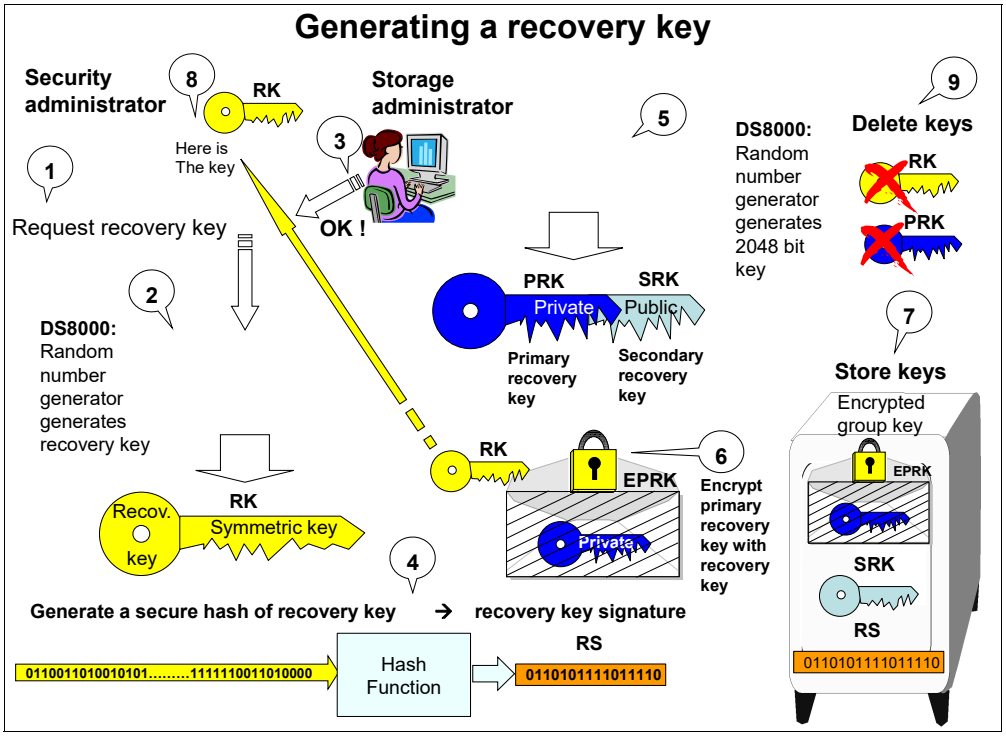


Figure 3-14 Generate a recovery key

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 that is 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. The DS8000 wraps the PRK with the RK to produce the encrypted primary recovery key (EPRK).
7. The EPRK, SRK, and 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 described here (the Security Administrator must reinput the RK).
9. The DS8000 deletes the PRK and the RK.

When you configure an encryption group with an RK defined, you must complete the following steps in addition to the steps (see Figure 3-15 on page 47) that are shown in Figure 3-7 on page 35:

1. The storage facility wraps the GK with the SRK to produce the encrypted group recovery key (EGRK).
2. The EGRK is stored 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.

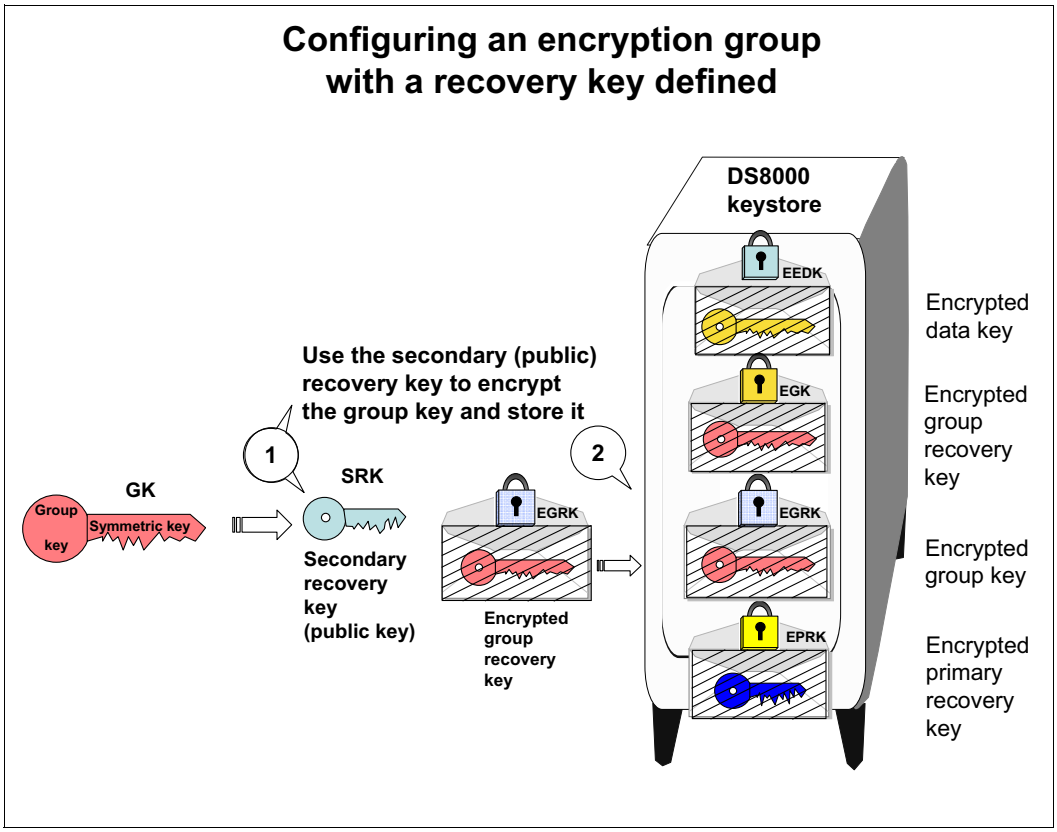


Figure 3-15 Set up encryption with a recovery key defined

Using a recovery key to unlock a DS8000

After a power-off and power-on, the DS8000 attempts to contact all other configured key servers to obtain the required key if it cannot obtain the required DK from a key server.

On a DS8000 with an RK configured, an option exists to allow a Security Administrator enter 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 3-16). The RK process includes the following steps:

1. The DS8000 cannot communicate with any key server.
2. The DS8000 allows the RK to be entered.
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.

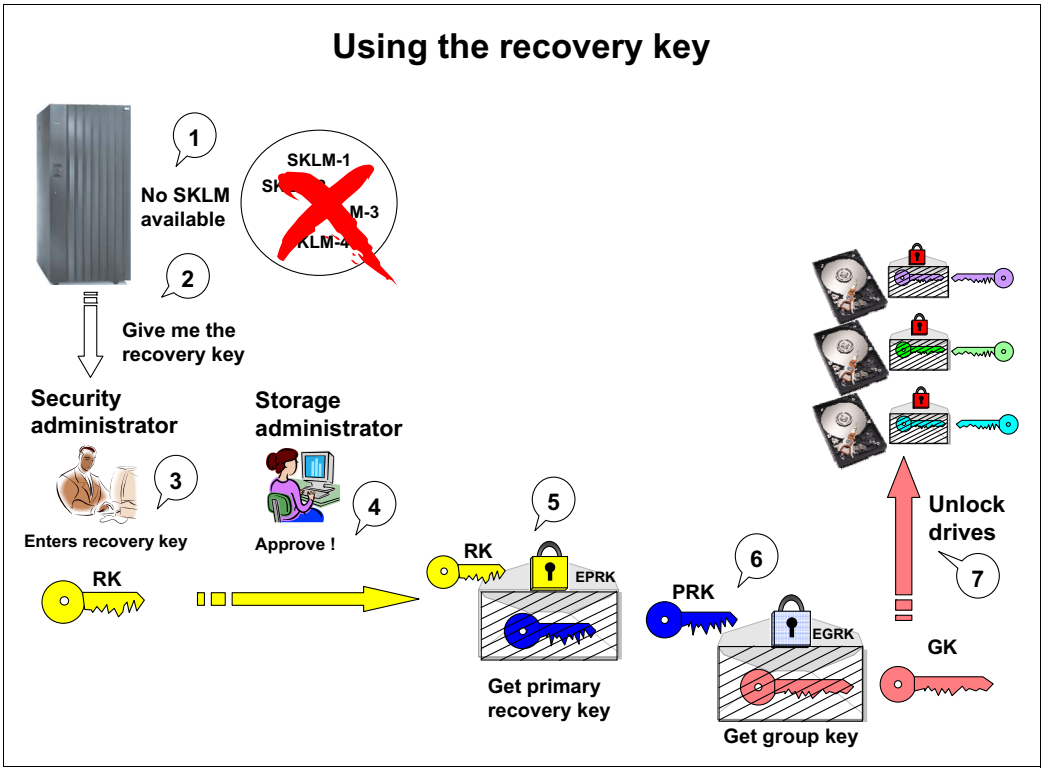


Figure 3-16 Use the recovery key

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 any time 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 3-17 on page 49). Only then, it generates a new RK. For example, rekeying the RK is not possible on a DS8000 that was stolen and placed in a separate environment.

During the rekey operation, the following steps are completed:

1. The DS8000 sends the EEDK and its public key to Key Manager Server and requests a rekey validation.
2. Key Manager attempts to decrypt the DK.
3. If Key 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.

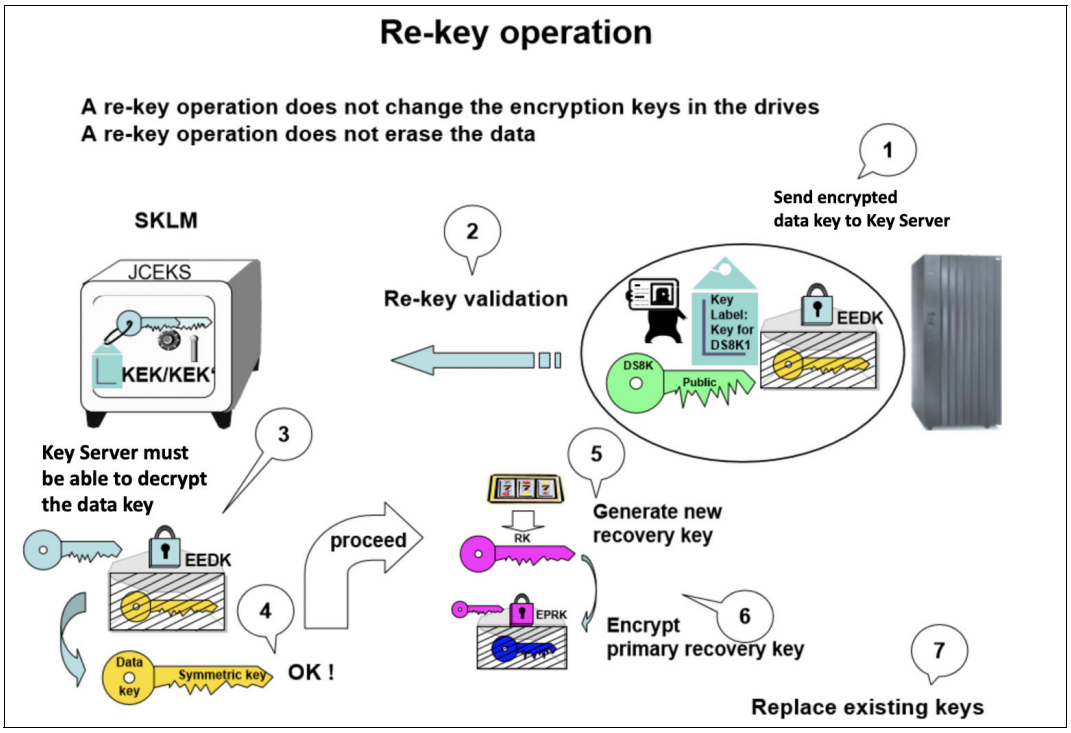


Figure 3-17 Rekeying the recovery key

3.3.2 Disabling or enabling a recovery key

The RK can be optionally disabled or configured before establishing an initial logical configuration. This section describes the steps 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 process must be done as your first task before the encryption group is defined. The process of disabling the RK includes the following steps:

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

The encryption group can now be defined. For more information, see 5.6, “Configuration for TCT encryption” on page 179.

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 process 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 45. For more information, see 5.6, “Configuration for TCT encryption” on page 179.

3.4 Dual key server support (IBM Proprietary Protocol only)

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

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

Note: Consider the following points:

- Having a key server platform in secure key mode on z/OS platform for the DS8000 is not at all common. Typically, the z/OS runs on volumes that are on a DS8000, which is encrypted. This situation increases the chance of running into a deadlock situation. Having a second DS8000 that is not encrypted to run the key server also is not common.
- In an IBM Security Key Lifecycle Manager Multi-Master configuration, two key servers are at least deployed and keys are automatically replicated.

A key server operating in secure key mode typically does not support exporting 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 3-18 on page 51):

- 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 Figure 3-18 on page 51 and Figure 3-19 on page 52).

- The public key from key label 2 is exported to platform IBM Security Key Lifecycle Manager for UNIX (abbreviated as “SKLM-UNIX” in Figure 3-18 and Figure 3-19 on page 52).

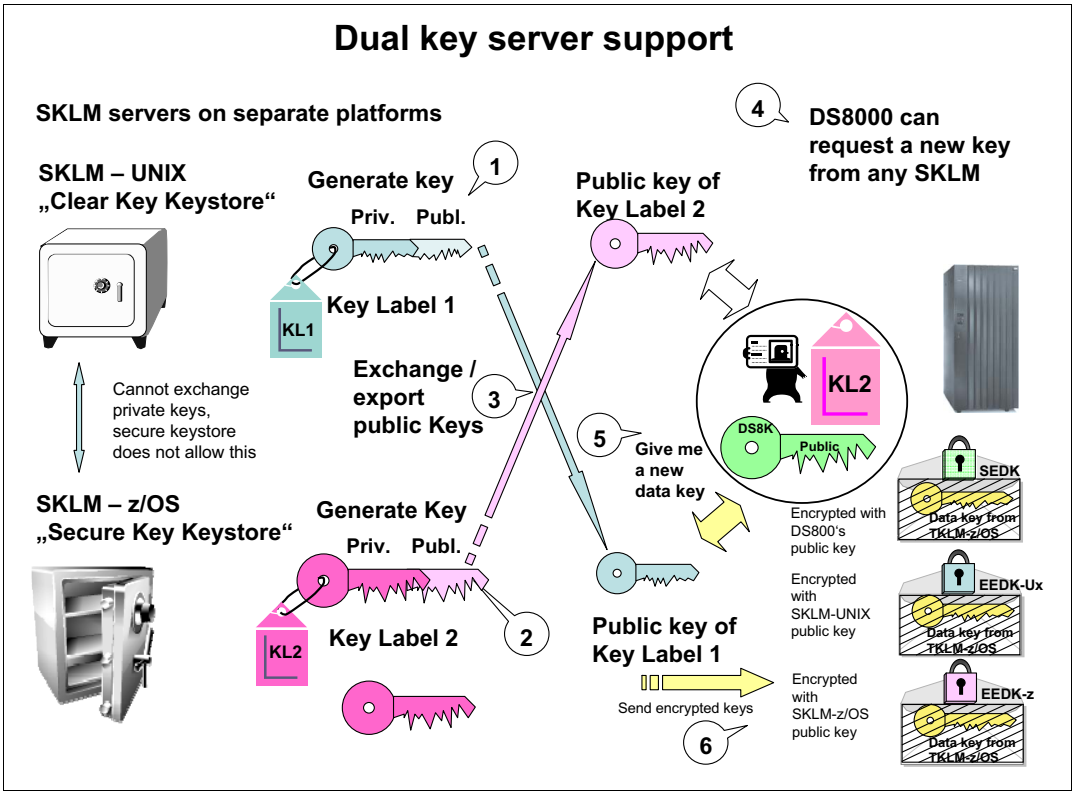


Figure 3-18 Dual key server support

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 that is associated with each key label. Therefore, the DS8000 now includes two separate EEDKs.

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 3-18:

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, wraps the DK with its own public key to produce EEDK with Z (EEDK-z), and wraps the DK with IBM Security Key Lifecycle Manager for the UNIX public key to produce EEDK 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 3-19), 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.

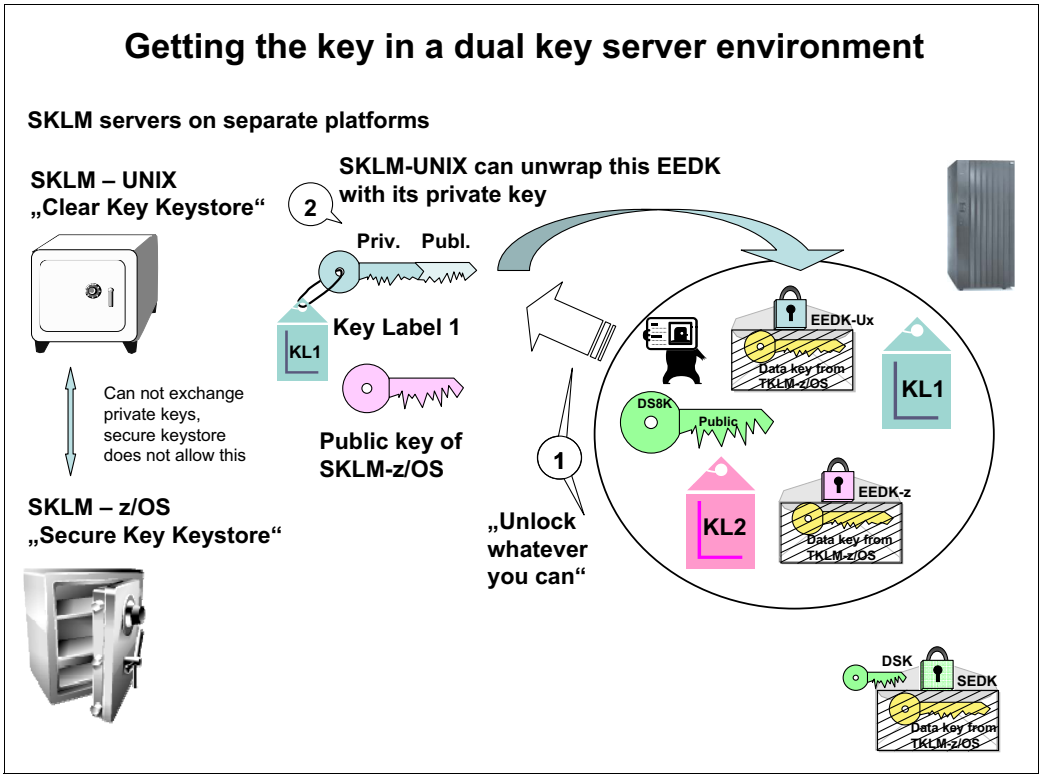


Figure 3-19 Get the recovery key in a dual server environment

In the example that is shown in Figure 3-19, 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 server.

3.5 DS8000 TCT encryption Key Management using KMIP

As described in 3.1, “DS8000 data at rest disk encryption” on page 28, DAR encryption is implemented in the DS8000 by using external key servers. This approach provides a physical and logical isolation between the encryption keys and the DS8000 and thus protects data if the DS8000 or any of its drives were stolen.

However, DAR encryption does not protect data as it is transferred to or from the cloud when TCT is used.

Starting with DS8000 Release 8.5, you can encrypt data as it is transmitted between the DS8000 and the cloud. This encryption mechanism also uses external key servers.

This section describes how the IBM Security Key Lifecycle Manager and Gemalto SafeNet KS servers manages and creates the encryption keys that are used by the DS8000 during encryption group and cloud server connection configuration.

It also describes the major components that are used with TCT encryption and behavior during normal and abnormal operation.

In the direct key model, a DK is stored on the key server. The DK is created by 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.

For more information about TCT functions, see *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381.

TCT encryption components

The major components for TCT encryption are as follows:

- Key Manager:

  The Key Manager creates and stores the DK with the association to the Device Group.

  The DK is an Advanced Encryption Standard (AES) 256 symmetric wrapping key that is identified by a DS8000 device-specific UUID and is stored in the cache of the encrypting DS8000 after initial retrieval from Key Manager as well. It wraps the Cloud Encryption Key (CEK) to obtain Encrypted Cloud Encryption Key (ECEK).

- DS8000:

  The DS8000 stores the DK in the cache and its specific UUID in the KR. It generates and stores a specific CEK in the Encryption Engine. Consider the following points:

  – The CEK is an AES 256 symmetric key that is used to encrypt / decrypt data that is stored in the cloud and will be destroyed after use and wrapping.

  – The UUID is created by manufacturing and is unique for every DS8000. It is stored with the Cloud Data Object and in the KR of the DS8000. It identifies the DK uniquely on the key server.

- Data Object:

  The Data Object is the actual data that is stored in the Cloud Data Storage. It includes the encrypted customer data, the specific UUID from the encrypting DS8000 and the ECEK.

  The ECEK is the result of wrapping the CEK with the DK.

Figure 3-20 illustrates these components.

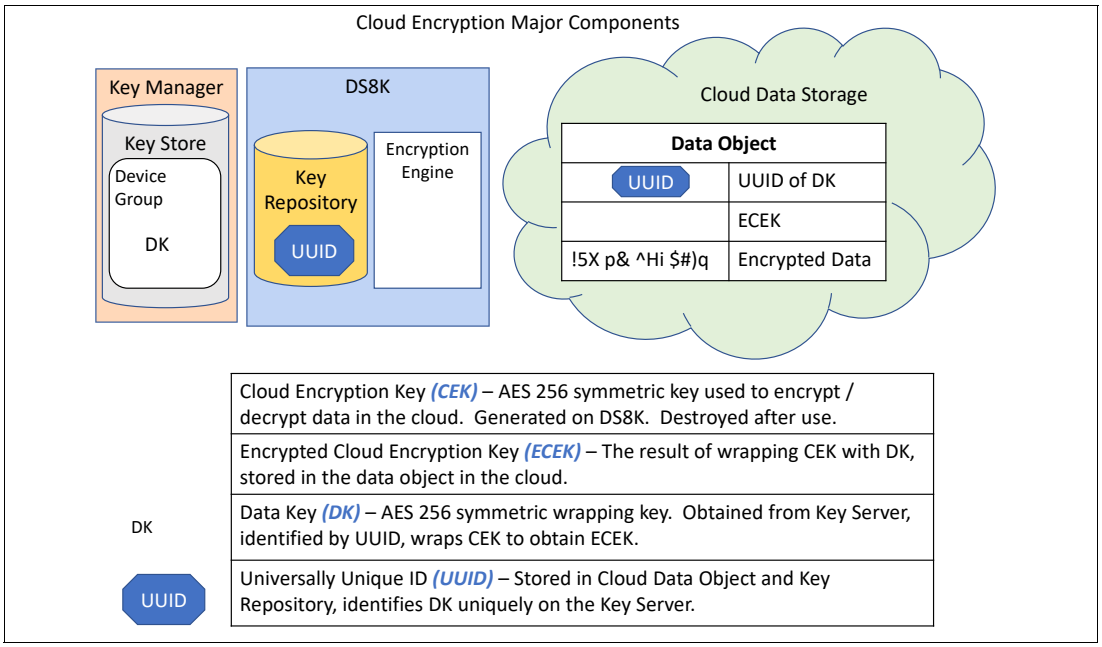


Figure 3-20 Cloud encryption major components

Encrypted writes to the cloud

Every encrypted data object write operation to the cloud is handled by the encrypting DS8000. The DS8000 adds its unique identifier to the data object and encrypts the data with a new unique CEK every time a write happens. All DKs are stored in the encryption key managers. The unencrypted CEK is destroyed after use.

Writes for encrypted cloud follow this high-level write sequence:

1. DS8K sends UUID to the Key Manager and requests a DK.
2. DS8K receives the DK from Key Manager.

> **Note:** Creating the DK happens during creation of the encryption key group. This is a one-time process. For more information about creating the encryption key group, see *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381.

3. DS8K uses a CEK to encrypt data.
4. DS8K wraps the CEK with the DK to obtain the ECEK.
5. DS8K stores its UUID, the ECEK, and the Encrypted Data in the Cloud Storage Object.

Figure 3-21 on page 55 illustrates the write sequence.

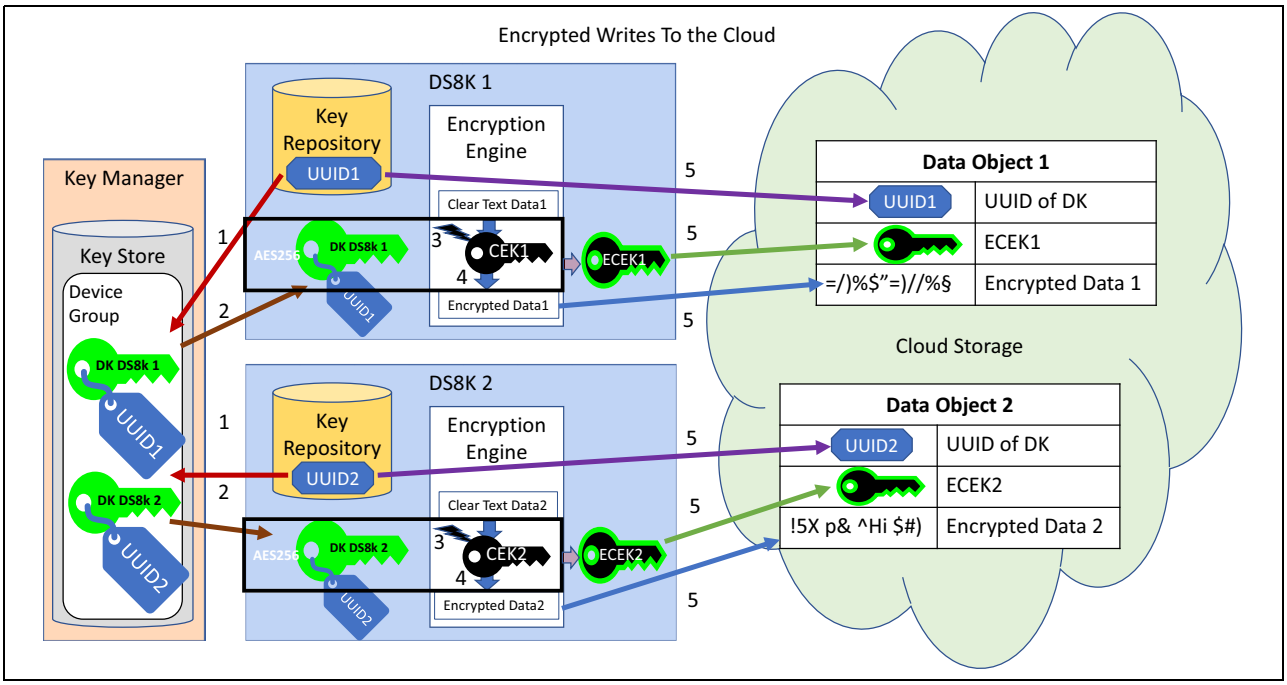


Figure 3-21 Encrypted writes to the cloud

In addition to the encrypted data, every Data Object in the cloud that is written by a specific DS8000 includes the unencrypted UUID of the DS8000 and the CEK. This key is encrypted by the DK that remains in the key server and in the encrypting DS8000.

Encrypted reads from the cloud

A DS8000 that was not the encrypting system might read an encrypted data object from the cloud. This might happen during normal operations or in a failure scenario where the original encrypting DS8000 is unavailable at the time of the read request.

The DS8000 with the read request must be part of the encryption environment, with access to the same key servers that the writing DS8000 had. The reading DS8000 must be able to obtain the correct wrapping DK from the key server to unwrap the CEK and to decrypt the data object. It stores the foreign UUID in its own KR and the foreign DK, requested from the key server during the read request, in the cache. If the UUID of the Data Object read matches the UUID stored in the KR, the DK is retrieved from Cache if it was initially retrieved from the key manager.

Reads for encrypted cloud follow this high-level read sequence:

1. DS8K 1 reads in Data Object 2 from Cloud Storage and DS8K 2 reads in Data Object 1 from Cloud Storage.
2. DS8K 1 sends UUID2 to Key Manager and DS8K 2 sends UUID1 to Key Manager.
3. DS8K 1 receives DK DS8k 2 with UUID2 from Key Manager. DS8K 2 receives DK DS8K 1 with UUID1 from Key Manager.
4. DS8K 1 unwraps CEK2 and DS8K 2 unwraps CEK1.
5. DS8K 1 decrypts Encrypted Data 2 and DS8K 2 decrypts Encrypted Data 1.

Figure 3-22 illustrates the read sequence.

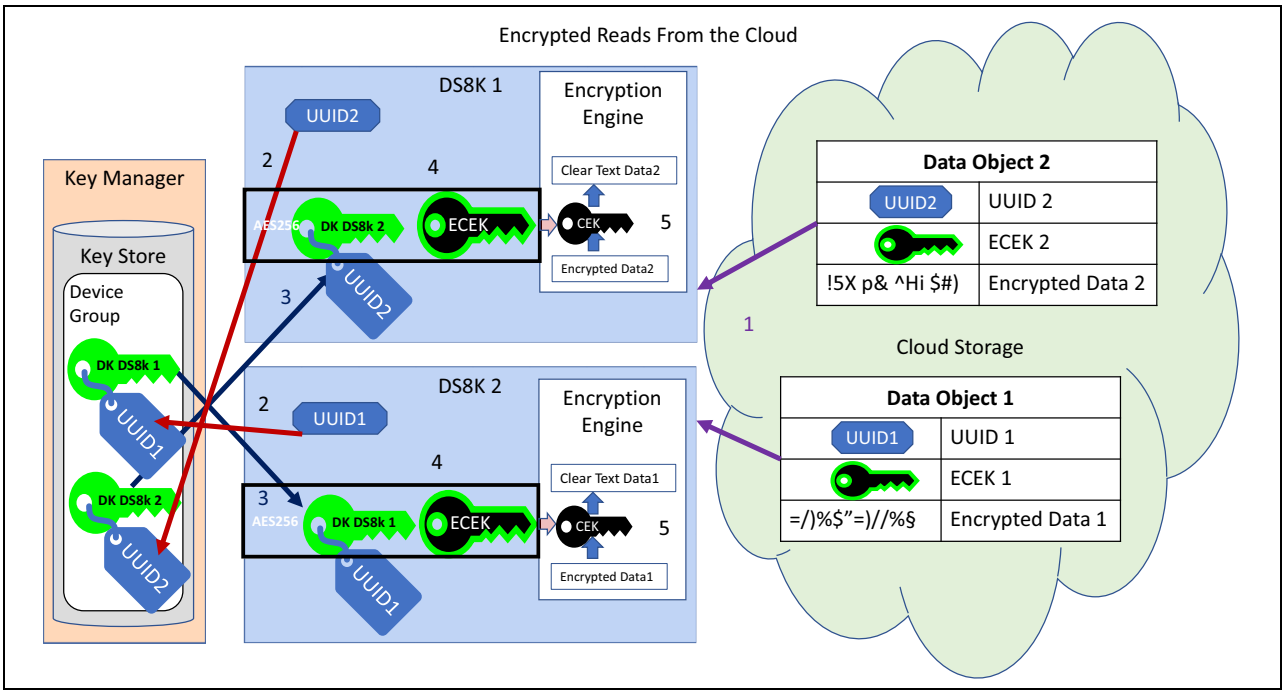


Figure 3-22 Encrypted reads from the cloud

DS8K 2 is now able to read all Data Object encrypted by DS8K 1 and vice versa without an extra key retrieval from the key server each time.

3.6 DS8000 endpoint encryption key management using KMIP

With DAR encryption, your data is protected when stored on FDE drives. With TCT, your data is protected when being transferred between the cloud when using TCT. Now with IBM Fibre Channel Endpoint Security, your data can be protected when transferred between the DS8900F storage system and an IBM z15.

Starting with DS8000 Release 9.0, you can use IBM Fibre Channel Endpoint Security encryption to encrypt data as it is transmitted between an IBM z15 Central Processor Complex (CPC) and the DS8900F. This encryption mechanism also uses external key servers.

This section describes how the IBM Security Key Lifecycle Manager manages and creates the encryption keys that are used by the DS8000 during encryption group and server connection configuration.

> **Important:** For IBM Fibre Channel Endpoint Security, we recommend IBM Security Key Lifecycle Manager V4.0.0.2 or later.

This section also describes the major components that are used with IBM Fibre Channel Endpoint Security encryption and behavior during normal and abnormal operations.

For more information about IBM Fibre Channel Endpoint Security encryption functions, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

IBM Fibre Channel Endpoint Security encryption components

IBM Fibre Channel Endpoint Security involves three major components:

- Two endpoint devices:
  - A z15, IBM Z CPC, also referred to as the *initiator*.
  - An IBM DS8900F storage system, also referred to as the *target*.

  IBM Fibre Channel Endpoint Security is set up between individual IBM FICON® or Fibre Channel (FC) port pairs of these devices. For FICON, the ports are logically connected through the definitions in the *input/output control data set* (IOCDS) of the host system. IBM Fibre Channel Endpoint Security requires no changes to the IOCDS.
- An *external key manager* that maintains the shared secrets that identify the trusted relationships between endpoint devices.

Figure 3-23 shows the IBM Fibre Channel Endpoint Security infrastructure.

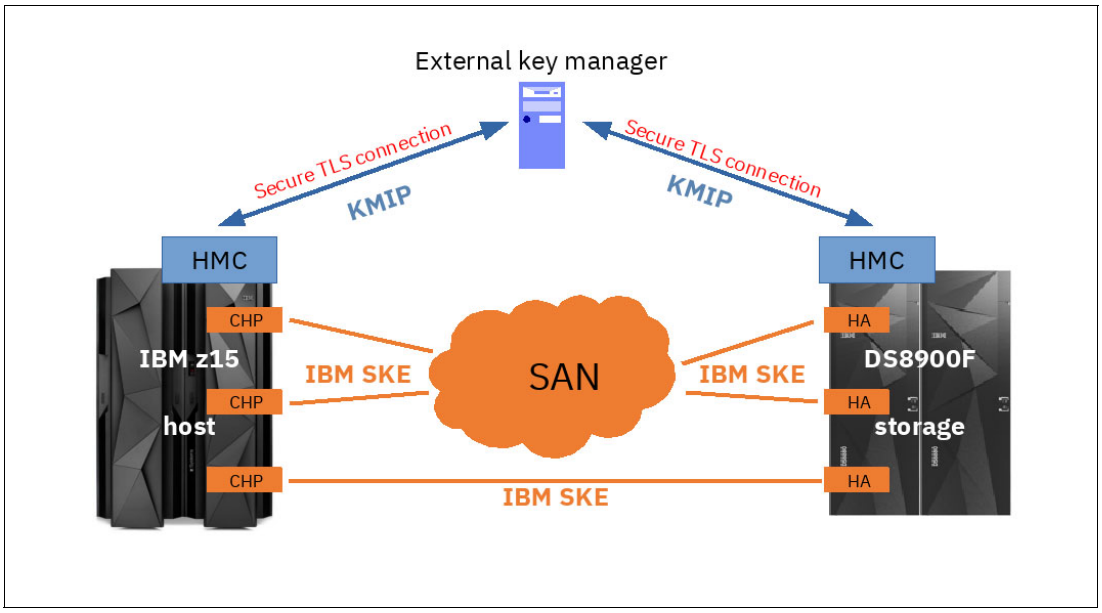


Figure 3-23 Main IBM Fibre Channel Endpoint Security components

External key manager

The external key manager maintains the shared secrets that associate pairs of host and storage system, or initiators and targets. Such a shared secret is called a Device Authentication Key (DAK). It is stored securely in the external key manager keystore. To make sure that a DAK is always available when needed, the external key manager must be redundant and configured for continuous availability.

Important: If IBM Fibre Channel Endpoint Security is in place and enforced, the external key manager is a crucial component during the start of an IBM Z CPC or any connected DS8900F storage system. If IBM Fibre Channel Endpoint Security is unavailable, FICON connections between host and storage fail to start and data cannot be accessed.

The endpoint devices retrieve the DAK from the external key manager:

- When IBM Fibre Channel Endpoint Security is set up for the first time (for example, at power-up).
- When the DAK is renewed according to the specified IBM Fibre Channel Endpoint Security policies.

During normal operation, each endpoint device maintains a copy of the DAK in its *local key manager* (LKM) to avoid excessive external key manager traffic.

Note: The LKM function is implemented in the IBM Z firmware.

The external key manager and endpoint devices use the industry-standard KMIP for their communication. IBM developed an extension to the KMIP protocol that adds support for peer-to-peer device groups, which you can use to store the trusted association of two devices (peers). The peers are the IBM Fibre Channel Endpoint Security endpoints, with the IBM Z system being the owner of the group, and the DS8900F storage system the partner. The FC worldwide node names (WWNNs) of the endpoint devices are used to provide unique identification. One such device group is created and maintained by the external key manager for each IBM Z CPC and DS8900F pair.

Figure 3-24 shows an example with two device groups, which associates an IBM z15 with two DS8900F systems.

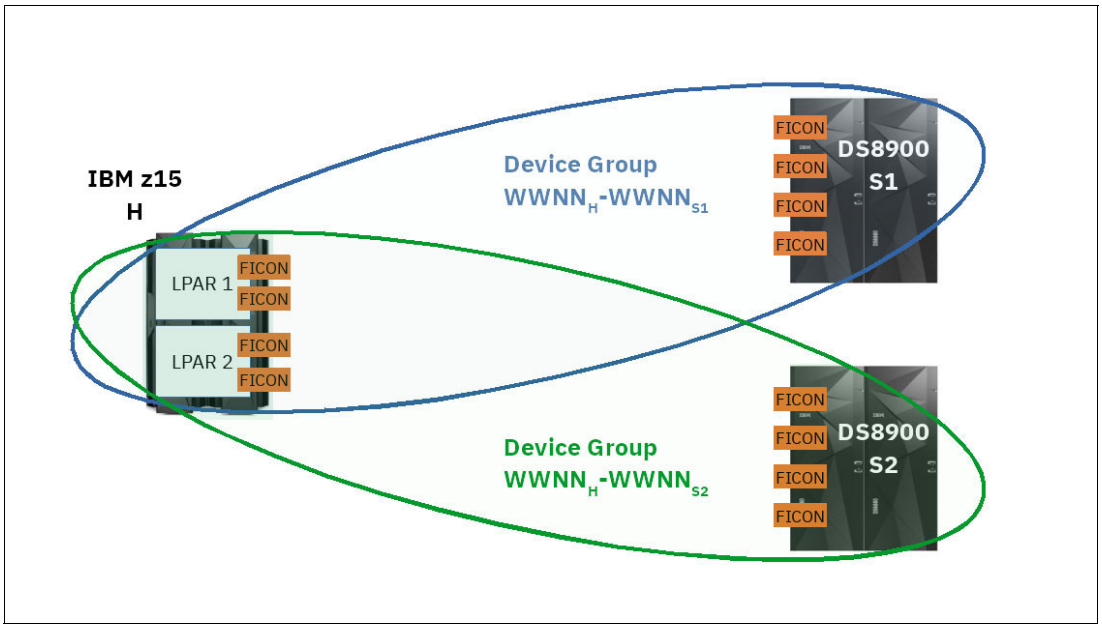


Figure 3-24 Peer to peer device group example

The external key manager is composed of the names of the peer-to-peer device groups from the WWNNs of the peers (endpoint devices) in the group. The device group contains the security credentials (certificates) that the external key manager needs to communicate with each of the peers and their common DAK. The peer-to-peer device group and initial DAK are created on the external key manager automatically at the request of the IBM Z CPC.

When endpoints must retrieve the DAK from the external key manager, they provide the WWNNs of both peers and identify themselves with their certificate. The external key manager then pulls the DAK from the matching device group and passes it to the requesting peer. The device certificates are presented to the external key manager by the HMCs (IBM Z and DS8900F) on behalf of the initiator and target, respectively. This process is out-of-band communication.

Note: IBM Security Key Lifecycle Manager that is configured in Multi-Master mode is the only external key manager solution that is supported for IBM Fibre Channel Endpoint Security.

Endpoints: IBM Z CPC

Note: An IBM z15 with at least one encryption capable FICON adapter (FICON Express 16SA feature) is required for IBM Fibre Channel Endpoint Security.

From an FC or FICON perspective, the IBM Z CPC is the initiator of all I/O operations and IBM Fibre Channel Endpoint Security configuration activities. You can configure and enable IBM Fibre Channel Endpoint Security on an IBM Z CPC when the following requirements are met:

- The IBM Fibre Channel Endpoint Security feature is activated.
- The CP Assist Cryptographic Facility (CPACF) feature is activated (however, the Fibre Channel Security Endpoint Security feature cannot be ordered without CPACF enablement).
- At least one encryption capable FC adapter is installed.

To enable IBM Fibre Channel Endpoint Security for an IBM Z CPC, you must define only the external key manager (IBM Security Key Lifecycle Manager) servers IP addresses or hostnames, and port numbers) in the corresponding configuration window in the IBM Z HMC. The IBM Fibre Channel Endpoint Security firmware that is running on the IBM Z CPC performs all the necessary steps to set up secure communication to the external key manager by using the HMC as the communication gateway and user interface.

The IBM Fibre Channel Endpoint Security firmware in the IBM Z CPC retrieves the DAK from the external key manager when needed and stores the DAK in the LKM. The local copy of the DAK is used for normal operations to avoid excessive external key manager traffic. The LKM uses the DAK in the authentication sequence with target peers and starts a key renewal of the DAK when it is due.

This part of the IBM Z firmware runs in an encapsulated container to ensure that the secrets it keeps cannot be compromised, for example, when a dump or trace is generated. It does not use client memory or processing power because it runs with system internal resources.

Endpoints: IBM DS8900F storage system

Storage systems are targets for I/O operations and IBM Fibre Channel Endpoint Security configuration requests. A DS8900F receives an IBM Fibre Channel Endpoint Security configuration request from an IBM Z CPC and acts on it. The DS8900F does not start endpoint protection on its own.

To support IBM Fibre Channel Endpoint Security, a DS8900F system must access the same external key manager (IBM Security Key Lifecycle Manager servers) as the IBM Z CPC. You can define the IBM Security Key Lifecycle Manager servers to the DS8900F through its HMC by using the GUI or DS CLI.

In addition to providing the IBM Security Key Lifecycle Manager servers' IP addresses and port numbers, you must ensure that the following credentials for secure communication between DS8900F and IBM Security Key Lifecycle Manager are in place:

- Export communication certificates from the IBM Security Key Lifecycle Manager servers and import them to the DS8900F.
- The factory-installed default communication certificates of a DS8900F or Z CPC are known and trusted by the IBM Security Key Lifecycle Manager servers. You do not have to take any further action if you intend to use those certificates.
- If you intend to use another certificate, you must install it on the DS8900F and import it to the IBM Security Key Lifecycle Manager servers.

Because the WWNNs of the endpoints are used to associate them with each other, you also must ensure that any certificate you provide for an endpoint contains its WWNN in the **Subject Alternative Name** field.

Note: If a DS8900F is configured for another type of encryption (DAR or TCT), the IBM Security Key Lifecycle Manager servers also can be used as external key manager for IBM Fibre Channel Endpoint Security if they meet the IBM Fibre Channel Endpoint Security requirements.

Similar to the IBM Z CPC, the HMC acts mainly as a user interface and communication gateway in this case. Communication to the external key manager is started by the DS8900F server nodes. Both the IBM Z CPC and DS8900F server nodes also have their own LKM to keep the DAK during normal operations.

After you complete the steps to configure a DS8900F for IBM Fibre Channel Endpoint Security, it contacts the external key manager and tests whether it can perform all necessary KMIP operations.

Note: For these tests, the DS8900F creates a special peer-to-peer device group with the external key manager. In the IBM Security Key Lifecycle Manager list of device groups, you can identify it by its name. This name consists of the letter "D", which is followed by the DS8900F WWNN (repeated twice), which indicates that it is both the owner and partner of the group.

Fibre Channel endpoints

Until now, we only looked at the endpoint devices (host and storage) as a whole. However, IBM Fibre Channel Endpoint Security is a function that protects data flowing between the following FC port pairs:

- Host or initiator ports
- Storage or target ports

The port pairs perform the authentication and encryption set up individually. They communicate inband over FC and use the IBM Secure Key Exchange (SKE) protocol, which was developed by IBM based on the industry standard Fibre Channel Security Protocols 2 (FC-SP 2).

3.6.1 IBM Fibre Channel Endpoint Security settings and policies

If all hardware and software requirements are met and you successfully defined the external key manager to both endpoint devices (initiator and target), you can enable IBM Fibre Channel Endpoint Security on a storage system host adapter (target) port level. A target port can be configured to one or more initiator ports by the IOCDS. The policy setting affects all possible connections to this target port from any configured initiator ports. You can set each port individually to one of the following IBM Fibre Channel Endpoint Security policies:

Disabled The target port does not signal IBM Fibre Channel Endpoint Security capability to the initiators. Therefore, all attached initiator ports do not try to set up IBM Fibre Channel Endpoint Security. The FC endpoint pairs act as if IBM Fibre Channel Endpoint Security did not exist.

Enabled The target port signals IBM Fibre Channel Endpoint Security capability to the initiators. If an attached initiator port is IBM Fibre Channel Endpoint Security capable, it tries to set up IBM Fibre Channel Endpoint Security. However, the port pair can set up a connection, regardless of whether the authorization succeeds. If a host port is not IBM Fibre Channel Endpoint Security capable, or if the necessary connection to IBM Security Key Lifecycle Manager has not been set up yet, it cannot start the IBM Fibre Channel Endpoint Security authorization sequence, but can also connect to the target without it. This policy is also called *Audit Mode* because you can use it to verify the IBM Fibre Channel Endpoint Security configuration without affecting access to data.

Enforced The target port signals IBM Fibre Channel Endpoint Security capability to the initiators. If an attached initiator port is IBM Fibre Channel Endpoint Security-capable, it tries to set up IBM Fibre Channel Endpoint Security. If the authorization succeeds, the port pair can connect. If the authorization fails, the port pair cannot set up a connection. If a host port is not IBM Fibre Channel Endpoint Security-capable or if the necessary connection to IBM Security Key Lifecycle Manager is not yet set up, the connection also fails.

Whenever you change the IBM Fibre Channel Endpoint Security policy of a target port, this port goes offline (drop light) and the IBM Fibre Channel Endpoint Security negotiation starts from the beginning. This way, you can switch IBM Fibre Channel Endpoint Security on and off while the systems are in operation.

A short period occurs in which the affected port pairs cannot transfer data until the connections are established again. This interruption is handled by the z/OS Input/Output Supervisor (IOS) multipathing, and is transparent to the running applications.

More details: For more information about implementation, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

4

Planning and guidelines for IBM DS8000 encryption

This chapter describes planning for an IBM DS8000 encryption-capable storage system.

This chapter includes the following topics:

- 4.1, “About certificates” on page 64
- 4.2, “Planning and implementation process flow” on page 65
- 4.3, “Encryption-capable DS8000 ordering and configuration” on page 66
- 4.4, “Licensing” on page 67
- 4.5, “Advice for encryption in storage environments” on page 67
- 4.6, “Multiple key managers for redundancy” on page 71

Note: The focus of this chapter is for data at rest (DAR) encryption when IBM Security Key Lifecycle Manager is used as the key server. If you use another supported external key server, see the appropriate vendor documentation.

© Copyright IBM Corp. 2009, 2021. All rights reserved.

4.1 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 8.1and later.

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 (SP) 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 IBM Security Key Lifecycle Manager V3.0.0.2 and Gemalto SafeNet KeySecure (KS), to the DS8000 by Secure Sockets Layer (SSL) session and a specific username.

The Gen 1 certificates are supported for all machines in the DS8000 series except DS8000 machines that were included with Release 8.1 and later. 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.

For a DS8000 with Release 8.1 and later that has the Gen 2 certificate:

- If you need TLS 1.2 support with DS8000 Release 8.1 and later, you must use or upgrade to IBM Security Key Lifecycle Manager V2.6 and later because TLS 1.2 is not supported by Tivoli Key Lifecycle Manager Version 2.x.
- If you do not require TLS 1.2, DS8000 Release 8.1 and later with Gen 2 certificate can still use IBM Proprietary Protocol by way of TCP, which is supported by Tivoli Key Lifecycle Manager Version 2.x.
- If KMIP in combination with IBM Security Key Lifecycle Manager is required, you must upgrade to IBM Security Key Lifecycle Manager V3.0.0.2 or later.

DS8900 includes a Gen 2+ and Gen 3 certificate. Consider the following points:

- Gen 2+ certificates include 112-bit security strength and were introduced in Release 9.0. The Gen 2+ certificates meet the requirements of the NIST SP 800-131a: Transitions Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.

  Gen 2+ certificates are the same as Gen 2 certificates, but include another use of the **Subject Alternative Name** field. The use of this field to provide the Fibre Channel (FC) worldwide node name (WWNN) is standardized. Gen 2+ is the active certificate when the DS8900 is shipped.

- Gen 3 certificates feature 128 - 192-bit security strength. The Gen 2+ certificates meet the requirements of the NIST SP 800-131a beyond the year 2031. The Gen 3 certificate is included with the DS8900 system, but is dormant. The IBM Security Key Lifecycle Manager V4.0 server automatically trusts a DS8900 Gen 3 certificate.

4.2 Planning and implementation process flow

Figure 4-1 shows the planning and implementation process for an encryption-capable DS8000. The details for this process are described in subsequent sections of this chapter. Figure 4-1 shows the overall decision flow and outcomes.

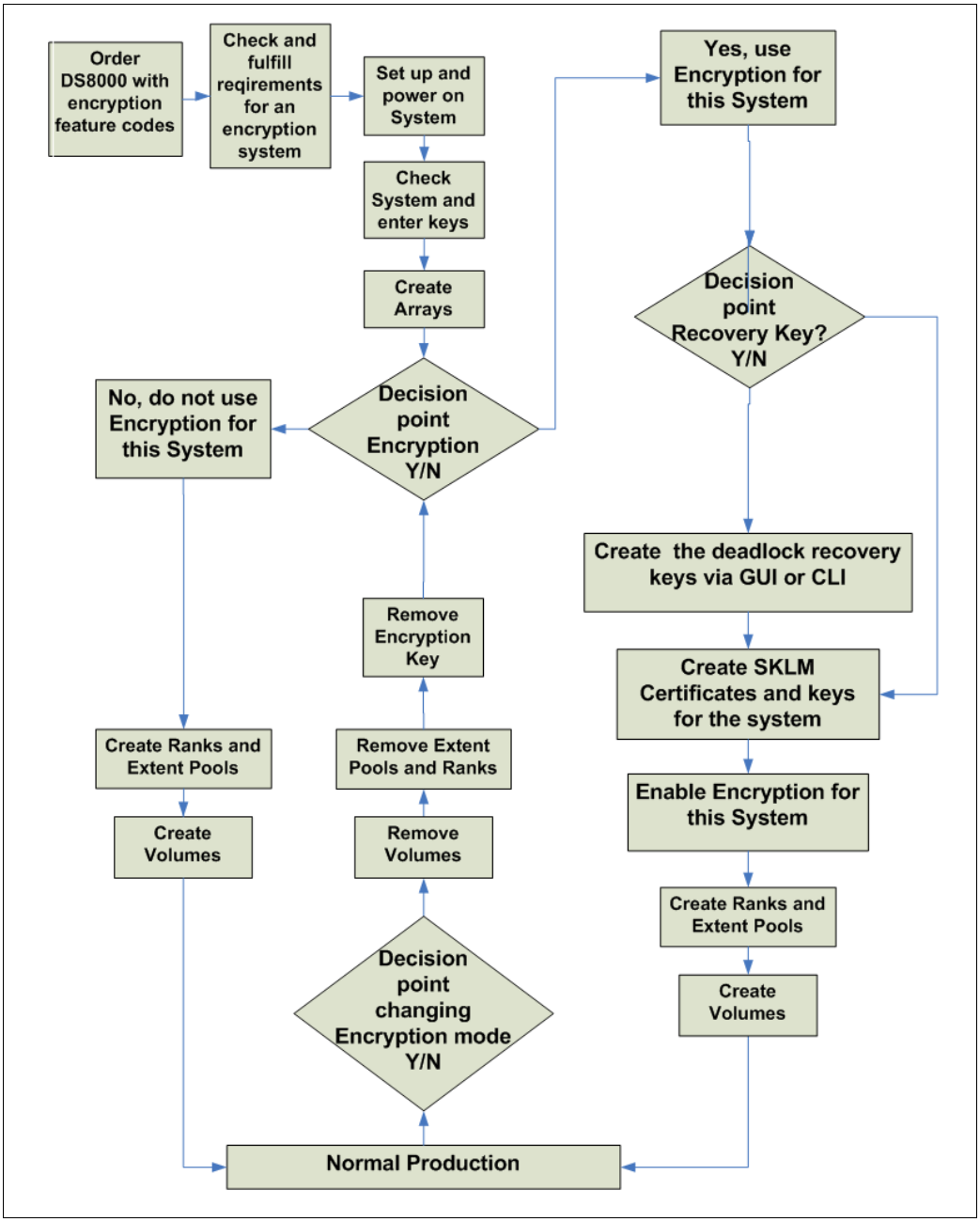


Figure 4-1 Encryption implementation planning flow

4.3 Encryption-capable DS8000 ordering and configuration

To enable encryption on a DS8000 system, the Full Disk Encryption (FDE) feature is required. All the supported DS8000 models are equipped with FDE drives (hard disk or flash drives) by default.

When you plan to use IBM Security Key Lifecycle Manager as your external key server, you must acquire an isolated key server. For more information about the hardware and operating system requirements, see Detailed system requirements for a specific product.

You must still perform the specific tasks to enable or disable encryption, as described in Chapter 5, “IBM DS8000 encryption implementation” on page 73.

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 and the IBM service representative installs the DS8000.
2. After the storage system is made available to the client, the client must download the activations keys from the DSFA website and activate them. Encryption is part of the DS8000 base license since Release 8.0.
3. You must enable DAR encryption.
4. For DAR encryption, create a deadlock recovery key (RK) by using the GUI or DS Command-Line Interface (DS CLI).
5. Now that the system can be recovered at any time by using the RK, 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.
6. You can now configure pools and assign arrays.

Notes: Consider the following points:

- All arrays and storage pools on an encryption-capable DS8000 must be configured with the same encryption group attribute. This option is available when DS CLI is used. (By default, the GUI creates encrypted storage pools if DAR is enabled.) The first storage pool 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 arrays and storage pools must be unconfigured. Unconfiguring an encryption-enabled array causes any data that was stored on the array to be **cryptographically erased** and later, overwritten to reinitialize the array.

4.4 Licensing

When ordering the IBM Security Key Lifecycle Manager, consider the following points:

- Number of IBM Security Key Lifecycle Manager servers per data center
- Amount of data to be encrypted (if you plan to use DAR encryption)
- Number of IBM Z servers

Note: In the past, IBM Security Key Lifecycle Manager was licensed based on the quantity of drives to be encrypted.

The following components must be ordered:

- For DAR encryption:
  - IBM Security Key Lifecycle Manager Basic Edition:
    - For Master-Clone (one per instance of IBM Security Key Lifecycle Manager server and Clone)
    - For Multi-Master (one per instance of IBM Security Key Lifecycle Manager server)
  - IBM Security Key Lifecycle Manager for RAW Decimal Terabyte Storage Resource Value Units (RVUs). Quantity = X where X is the RVU calculation. For more information, see License usage metrics.
- For IBM Fibre Channel Endpoint Security: IBM Security Key Lifecycle Manager Z-platform for KMIP client (one for each IBM Z system)
- For Transparent Cloud Tiering (TCT): Same as for DAR, based on the raw capacity of the DS8900F

IBM Security Key Lifecycle Manager server software entitlements must be purchased for production and high-availability and disaster recovery environments (HA/DR). No special licensing is needed for HA/DR. If the data is encrypted and IBM Security Key Lifecycle Manager servers the encryption keys, the server must be licensed.

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

After the IBM Security Key Lifecycle Manager software is licensed, it is available for download through IBM Passport Advantage. For more information about IBM Passport advantage, see Passport Advantage Express Overview.

4.5 Advice for encryption in storage environments

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

4.5.1 Using LDAP authentication

Ideally, a best 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 (UID) for both roles in the DS8000.

4.5.2 Availability

Consider the following considerations and best practices:

- DS8000

  The DS8000 must be configured with the dual Hardware Management Console (HMC) option to provide redundant access to the client network. DS8900F is always delivered with dual HMCs.

- 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.

4.5.3 Encryption deadlock prevention

Keep the following considerations and best 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 RK 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 best practice is to establish the independence of each recovery site from the other recovery sites.
  - If the RK management is enabled, the client must have a documented process to handle and maintain the deadlock RKs 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 RK 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*.

    Note: Consider the following points:
    - The DS8000 requires at least one isolated key server to be configured, but it is a best practice to use two for redundancy.
    - For TCT encryption and IBM Fibre Channel Endpoint Security encryption, IBM Security Key Lifecycle Manager must be set up in a Multi-Master configuration. This type of configuration requires one IBM Security Key Lifecycle Manager license per server.

    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 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 more 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 best practice and provides extra 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 6.1, “Rekeying the data key for data at rest encryption” on page 214.

- 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 RK 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 best 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 nonzero 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, which reflects a degraded key server environment.

The following conditions are monitored and reported:

- If the DS8000 cannot receive a required data key (DK) 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 DK 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 DKs that are configured on the DS8000 is verified at 8-hour intervals. Loss of the ability to unwrap a configured DK 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 DKs that are configured on the DS8000 at 8-hour intervals. If detected, this condition is reported to the client and to IBM.

4.6 Multiple key managers for redundancy

Whether you use the IBM Security Key Lifecycle Manager product or any other supported external key managers, such as the Thales CipherTrust Manager or Gemalto SafeNet KS, you must be certain that at least one key server always remains available. You need some redundancy, which can be achieved by deploying multiple key servers:

- IBM Security Key Lifecycle Managers

  To ensure continuous key and certificate availability to encrypting devices, configure your IBM Security Key Lifecycle Manager servers in a Multi-Master setup.

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

  This setup 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 system.

- Gemalto SafeNet KS

  To ensure continuous key and certificate availability to encrypting devices, configure your Gemalto Safenet KS servers in a clustered setup.

- Thales Vormetric Data Security Manager (DSM)

  To ensure continuous key and certificate availability to encrypting devices, configure your Thales Vormetric DSM servers in a clustered setup.

- Thales CipherTrust Manager

  To ensure continuous key and certificate availability to encrypting devices, configure your Thales CipherTrust Manager servers in a clustered setup.

5

IBM DS8000 encryption implementation

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

This chapter includes the following topics:

- 5.1, “Installing IBM Security Key Lifecycle Manager V4.0” on page 74
- 5.2, “WebSphere, Java, and IBM Security Key Lifecycle Manager hardening” on page 83
- 5.3, “Migrating from an earlier version of IBM Security Key Lifecycle Manager to IBM Security Key Lifecycle Manager V4.0” on page 85
- 5.4, “Key manager setup” on page 93
- 5.5, “Configuring data at rest” on page 151
- 5.6, “Configuration for TCT encryption” on page 179
- 5.7, “Configuration for IBM Fibre Channel Endpoint Security” on page 188
- 5.8, “Data at rest encryption and Copy Services functions” on page 194
- 5.9, “NIST SP 800-131a requirements for key servers” on page 194
- 5.10, “Migrating certificates” on page 199
- 5.11, “Using a custom-generated Gen 1 or Gen 2 certificate” on page 208

© Copyright IBM Corp. 2009, 2021. All rights reserved.

5.1 Installing IBM Security Key Lifecycle Manager V4.0

The IBM Security Key Lifecycle Manager V4.0 installation, including all prerequisites, is described at IBM Security Key Lifecycle Manager 3.0.0.

> **Note:** At the time of writing, IBM Security Key Lifecycle Manager is being rebranded as IBM Security Guardium Key Lifecycle Manager Version 4.1. For more information, see IBM Security Guardium Key Lifecycle Manager.

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

- Runtime environment:
  - IBM WebSphere® Application Server V9.0.5
  - IBM WebSphere SDK Java Technology Edition V8.0.5.37
- Database: IBM Db2 Workgroup Server Edition V11.1.4.4
- IBM Security Key Lifecycle Manager V4.0

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 5-1.

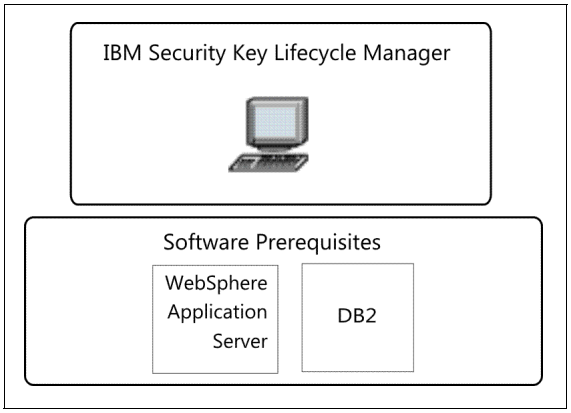


Figure 5-1 IBM Security Key Lifecycle Manager base architecture

> **Note:** This section describes an IBM Security Key Lifecycle Manager V4.0 installation on a Linux platform.

5.1.1 Before you start the installation

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

1. Make sure that a supported operating system is used for installation with the required libraries and operating system kernel settings. For more information, see IBM Security Key Lifecycle Manager 4.0.0.
2. Extract both installation images (disk1 and disk2) that you previously obtained from IBM Passport Advantage. Use the same subfolder to extract the images, as shown in Example 5-1. Keep the paths as short as possible.

Example 5-1 Extracted installation images

```
[root@sklmdemo74 sklm]# ls -l
drwxr-xr-x. 8 root root 4096 Nov 29  2019 disk1
drwxr-xr-x. 3 root root   35 Nov 29  2019 disk2
```

5.1.2 Silent mode installation on Linux

A silent installation is a non-interactive installation that 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 contains encrypted password information. For more security, delete the response file immediately after the installation of IBM Security Key Lifecycle Manager.

You must add *encrypted* passwords to the relevant elements of the response file. Use the IBM Installation Manager Utility to create encrypted passwords.

Complete the following steps:

1. Go 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 5-2 for the syntax to encrypt a password.

Example 5-2 Create the encrypted password

```
/disk1/im/tools # ./imcl encryptString myPassw0rd
HUTvoorFkpZvxu48WXv8qQ==
```

2. IBM Security Key Lifecycle Manager includes sample response files that you can use as templates 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 5-3.

Example 5-3 Create a response file copy

```
[root@sklmdemo74 disk1]# cp SKLM_Silent_Linux_Resp.xml myresp.xml
[root@sklmdemo74 disk1]# ls -l *.xml
-rwxr-xr-x. 1 root root 5956 Nov 17 23:01 myresp.xml
-rwxr-xr-x. 1 root root 6153 Nov 29  2019 SKLM_Silent_Linux_Mig_25_Resp.xml
-rwxr-xr-x. 1 root root 6153 Nov 29  2019 SKLM_Silent_Linux_Mig_26_Resp.xml
-rwxr-xr-x. 1 root root 6159 Nov 29  2019 SKLM_Silent_Linux_Mig_27_Resp.xml
-rwxr-xr-x. 1 root root 6235 Nov 29  2019 SKLM_Silent_Linux_Mig_301_Resp.xml
-rwxr-xr-x. 1 root root 6231 Nov 29  2019 SKLM_Silent_Linux_Mig_30_Resp.xml
-rwxr-xr-x. 1 root root 5956 Nov 29  2019 SKLM_Silent_Linux_Resp.xml
-rwxr-xr-x. 1 root root  825 Nov 29  2019 SKLM_Uninstall_Linux_Resp.xml
-rwxr-xr-x. 1 root root  624 Nov 29  2019
TKLMPasswordPolicy.xml
```

3. Add the encrypted password that you created to the response file, as shown in Example 5-4. The example uses the same password for Db2, WebSphere Application Server Admin, and SKLMAdmin.

Example 5-4 Password modifications in the response file

```
<data key=''user.DB2_ADMIN_PWD,com.ibm.sklm40.db2.lin.ofng'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key=''user.CONFIRM_PASSWORD,com.ibm.sklm40.db2.lin.ofng'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
...
<data key='user.WAS_ADMIN_PASSWORD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.WAS_ADMIN_CONF_PWD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
...
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.SKLM_ADMIN_CONF_PWD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
```

Note: Tags that are mentioned in Example 5-4 are specific to the response file of the operating system where the installation is being done. Do not copy tags directly, but instead use the appropriate response file and change the password value.

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

Example 5-5 Repository

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

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

5. Start the installation, as shown in Example 5-6.

Example 5-6 Silent installation

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

5.1.3 Graphical user interface mode installation on Linux

IBM Security Key Lifecycle Manager provides a GUI installation program. IBM Installation Manager is used to install IBM Security Key Lifecycle Manager and its components. It presents a series of windows that prompt for the information that is required for installation. For more information, see IBM Security Key Lifecycle Manager 4.0.0.

To install IBM Security Key Lifecycle Manager in GUI mode, complete the following steps:

1. Go to the IBM Security Key Lifecycle Manager directory from which the file was extracted. Go to `disk1` and start the installation in GUI mode by running **`launch.sh`** on non-Windows systems and **`launch.bat`** on Windows systems. Figure 5-2 shows an example.

```
[root@sklmdemo74 sklm]# cd disk1
[root@sklmdemo74 disk1]# ls
ad                              SKLM_Silent_Linux_Mig_25_Resp.xml
diskTag.inf                     SKLM_Silent_Linux_Mig_26_Resp.xml
documentation                   SKLM_Silent_Linux_Mig_27_Resp.xml
im                              SKLM_Silent_Linux_Mig_301_Resp.xml
install.sh                      SKLM_Silent_Linux_Mig_30_Resp.xml
launchpad.sh                    SKLM_Silent_Linux_Resp.xml
md                              SKLM_Uninstall_Linux_Resp.xml
PRS                             TKLMPasswordPolicy.xml
silent_install.sh               toc
silent_install_withoutIM.sh     uninstallSKLM_linux.sh
silent_uninstallSKLM_linux.sh
[root@sklmdemo74 disk1]# ./install.sh
Default locale is English
Using default value for locale
No pre-installed IBM Installation Manager found on the system.
Installing IBM Security Key Lifecycle Manager 4.0
```

Figure 5-2 Start the installation in graphical user interface mode

2. On the Welcome window, information regarding all the packages is shown (see Figure 5-3). Installing IBM Security Key Lifecycle Manager also installs IBM Installation Manager, IBM Db2, and IBM WebSphere Application Server, which includes IBM Java and IBM Security Key Lifecycle Manager. Click **Next** to proceed.

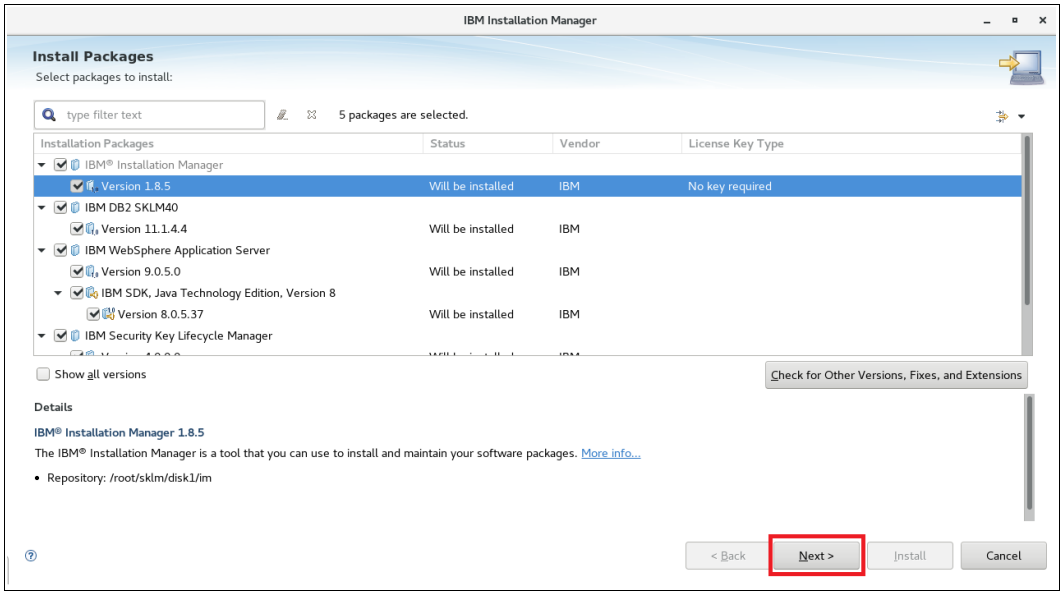


Figure 5-3 Installation: Welcome window

3. Read the license agreements that are shown in the next window and accept the licenses by clicking the **I accept the terms in the license agreements** option, and click **Next** to proceed.
4. Select the installation location for IBM Installation Manager. As a best practice, keep the default location for the installation. Click **Next** to proceed.
5. Select the installation location for IBM Db2, IBM WebSphere Application Server, and IBM Security Key Lifecycle Manager. Click **Next** to proceed.
6. In the Translation packages window, click **Next** to proceed. You can ignore the warning that is shown on the top of the page regarding a 64-bit version in use.
7. In the Feature Selection window, make sure that all the options are selected. Click **Next** to proceed.
8. In the IBM Db2 configuration window (see Figure 5-4 on page 79), provide an appropriate name for a new Db2 administrator user. This user is used to run the IBM Security Key Lifecycle Manager application and the Db2 database postinstallation. Make sure to provide a password that matches the operating system password requirements. Click **Next** to proceed.

> **Important:** Note the Db2 username and password because they will be required during operations like upgrades and migration.

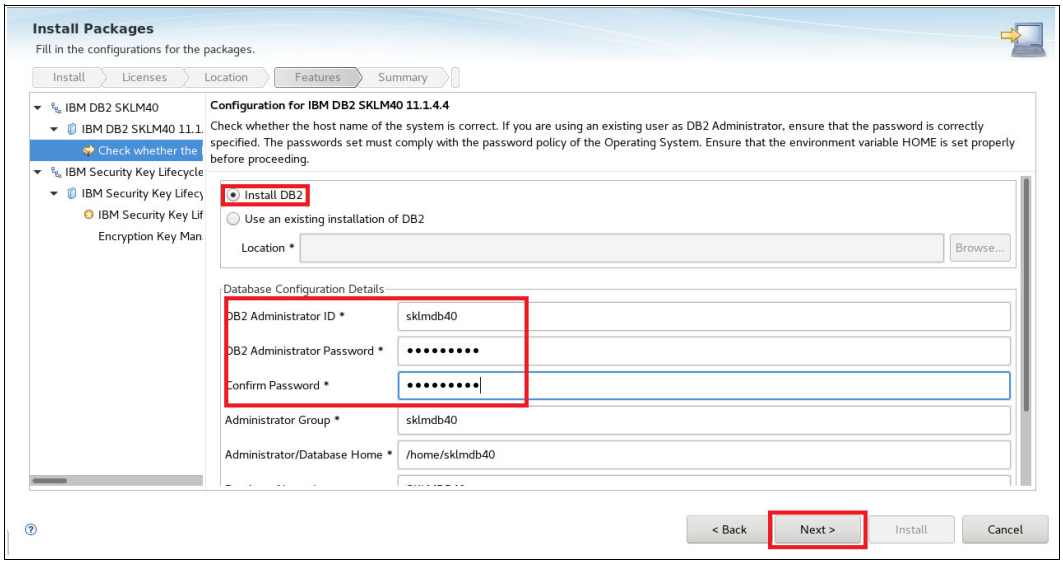


Figure 5-4 Installation: Db2 configuration window

9. In the IBM Security Key Lifecycle Manager configuration window (see Figure 5-5), provide the appropriate passwords for the `wasadmin` and `SKLMAdmin` users. Ensure that the passwords follow a proper password policy. For more information, see IBM Security Key Lifecycle Manager 4.0.0.

   Click **Next** to proceed.

> **Important:** The usernames and passwords are required for many operations, such as IBM Security Key Lifecycle Manager user creation, restarting IBM Security Key Lifecycle Manager services, configuring Multi-Master cluster, and upgrades and migrations. Also, note the port numbers.

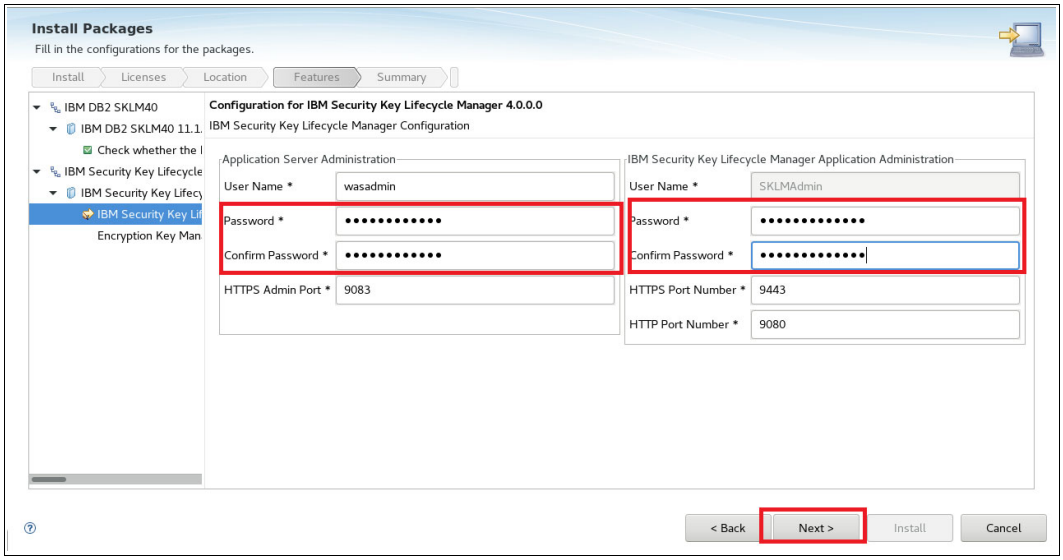


Figure 5-5 Installation: IBM Security Key Lifecycle Manager configuration window

10. In the Encryption Key Manager migration window, click **Next** to proceed.

11. In the Preinstall summary window (see Figure 5-6), review your choices before you install the product package. To change a selection, click **Back** to revise your selections. Click **Install** to begin the installation.

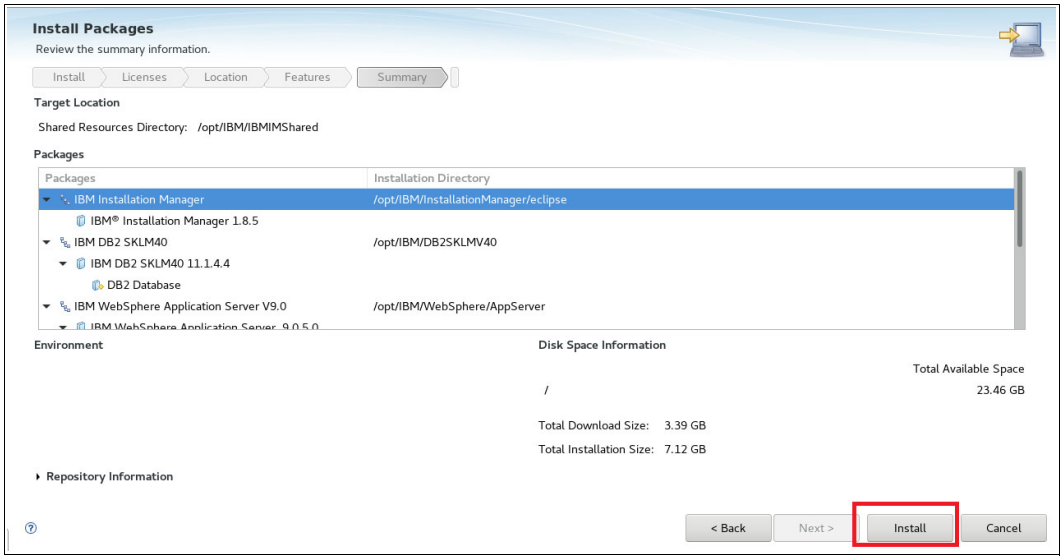


Figure 5-6 Installation: Summary window

12. A progress indicator shows the percentage of the installation that is complete. When the installation process is complete, a message confirms the completion of the process (see Figure 5-7). Click **Finish** to complete the installation and close the installation wizard.

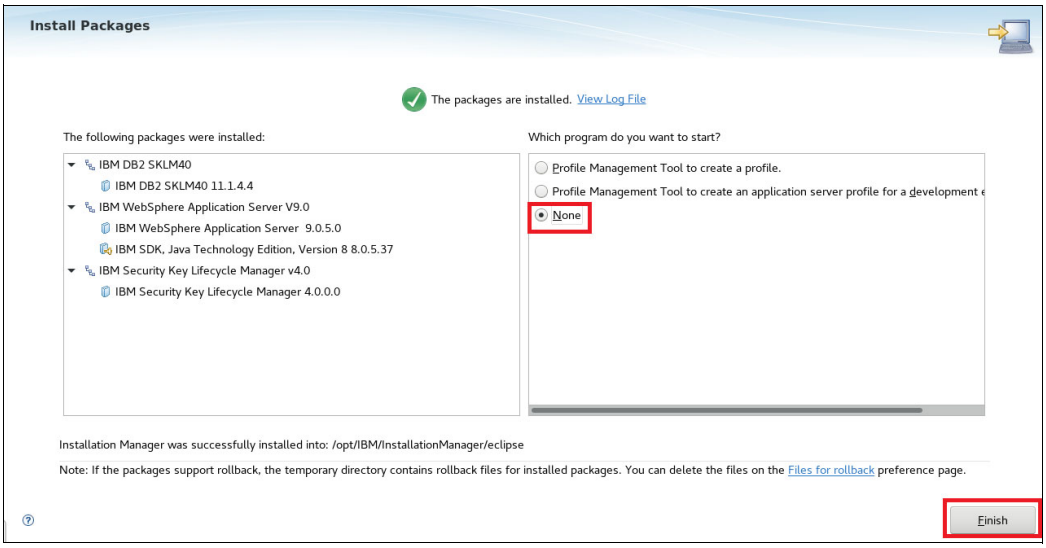


Figure 5-7 Installation complete window

5.1.4 Verifying the IBM Security Key Lifecycle Manager installation

After successful installing IBM Security Key Lifecycle Manager by using silent mode installation or graphical user mode installation, you should verify that IBM Security Key Lifecycle Manager properly installed by completing the following steps:

1. Start a browser.
2. Log in to IBM Security Key Lifecycle Manager (see Figure 5-8) by using the `SKLMAdmin` user and password that were set during the installation at the following address:

   `https://<ip address>:<port>/ibm/SKLM/jsp/Main.jsp`

   Important: IBM Security Key Lifecycle Manager V4.0 by default uses port 9443 for IBM Security Key Lifecycle Manager GUI and Rest APIs.

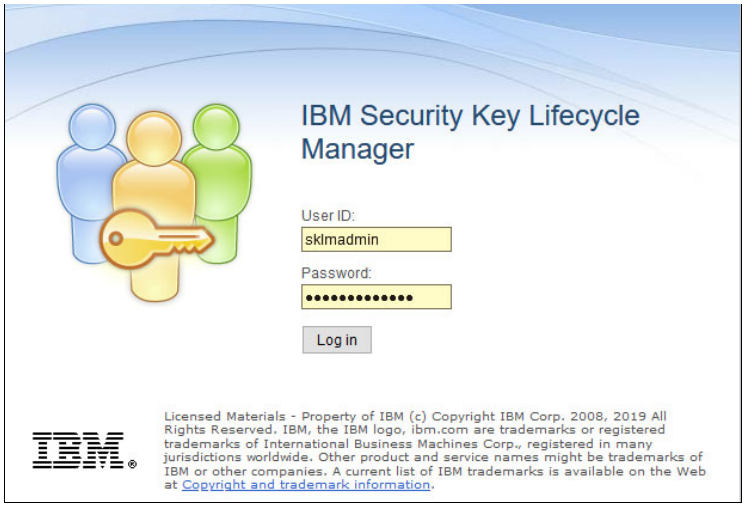


Figure 5-8 IBM Security Key Lifecycle Manager Login window

3. On the IBM Security Key Lifecycle Manager welcome window, click the circled question mark besides the username, and then click **About** (see Figure 5-9).

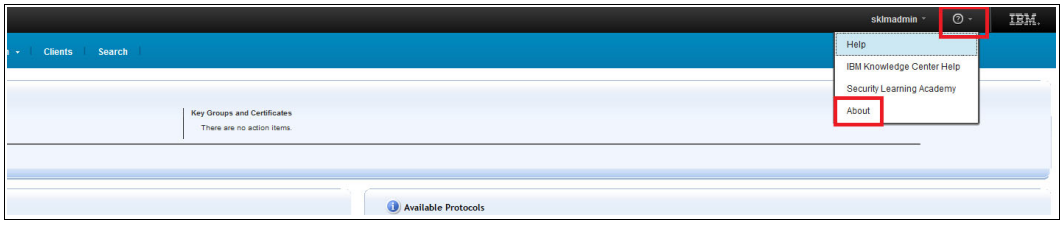


Figure 5-9 IBM Security Key Lifecycle Manager About menu

4. The installed version of IBM Security Key Lifecycle Manager and its bundled software is displayed (see Figure 5-10).

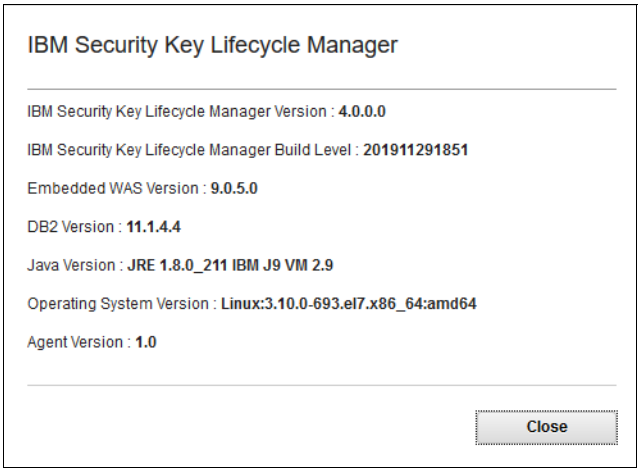


Figure 5-10 IBM Security Key Lifecycle Manager Version information

5.1.5 Installing Fix Pack 2 (or later) for IBM Security Guardium Key Lifecycle Manager Version 4.0

The fix packs for IBM Security Key Lifecycle Manager V4.0 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.

Fix Packs can be installed in silent mode, too.

Complete the following steps:

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

   Go to the IBM Installation Manager directory. The password encryption tool is available in `/opt/IBM/InstallationManager/eclipse/tools`. See Example 5-7 for the syntax for encrypting a password.

   Example 5-7 Create the encrypted password

   ```
   /opt/IBM/InstallationManager/eclipse/tools # ./imcl encryptString myPassw0rd
   HUTvoorFkpZvxu48WXv8qQ==
   ```

2. The IBM Security Key Lifecycle Manager Fix Pack 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 5-8.

   Example 5-8 Password modifications in the sample response file

   ```
   <data key='user.WAS_ADMIN_ID,com.ibm.sklm40.linux' value='wasadmin'/>
   <data key='user.WAS_ADMIN_PASSWORD,com.ibm.sklm40.linux'
   value='HUTvoorFkpZvxu48WXv8qQ=='/>
   <data key='user.SKLM_ADMIN_USER,com.ibm.sklm40.linux' value='SKLMAdmin'/>
   ```

```
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
<data key='user.DB_ADMIN_USER,com.ibm.sklm40.linux' value='sklmdb40'/>
<data key='user.DB_ADMIN_PASSWORD,com.ibm.sklm40.linux'
value='HUTvoorFkpZvxu48WXv8qQ=='/>
```

3. Start the update after you make the installer executable file, as shown in Example 5-9. You do not have to specify the response file.

Example 5-9 Install the fix pack

```
/sklm/fp1 # chmod +x ./silent_updateSKLM.sh
/sklm/fp1 # ./silent_updateSKLM.sh /opt/IBM/InstallationManager
/opt/IBM/WebSphere/AppServer wasadmin myPassw0rd
```

5.2 WebSphere, Java, and IBM Security Key Lifecycle Manager hardening

The WebSphere Application Server, Java, and IBM Security Key Lifecycle Manager itself are not set to provide the maximum level of security by default. They can be hardened on request to comply with the US government security standard NIST Special Publication (SP) 800-131 and to pass security scanners.

Perform the steps in this section on all installed IBM Security Key Lifecycle Manager servers that you are about to install.

5.2.1 WebSphere Application Server hardening

To comply with the US government NIST SP 800-131 security standard, you can configure the WebSphere Application Server to support the Transport Layer Security (TLS) 1.2 protocol.

For the full procedure, see Configuring Federal Information Processing Standard Java Secure Socket Extension files.

If you cannot access the Integrated Solutions Console from the browser after changing the Secure Sockets Layer (SSL) protocols to TLS 1.2, the browser might not be configured to support the protocol or does not support the protocol.

For more information about Firefox, see this website.

5.2.2 Java hardening

To force Java to use TLS 1.2, log in to each IBM Security Key Lifecycle Manager server as root and change the `java.security` file by completing the following steps:

1. Go to the Java security directory as shown in Example 5-10.

Example 5-10 Java security directory

```
[root@sklma ~]# cd /opt/IBM/WebSphere/AppServer/java/8.0/jre/lib/security/
```

2. Create a backup of the file java.security as shown in Example 5-11.

Example 5-11 Backup of the file java.security

```
[root@sklma security]# cp java.security java.security.org
```

3. Edit the file java.security and look for jdk.tls.disabledAlgorithms as shown in Example 5-12.

Example 5-12 Edit the java.security file

```
[root@sklma ~]# vi java.security
```

4. Append TLSv1 and TLSv1.1 to jdk.tls.disabledAlgorithms as shown in Example 5-13.

Example 5-13 jdk.tls.disabledAlgorithms modifications

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, MD5withRSA, DH keySize < 768,
3DES_EDE_CBC, DESede, \ EC keySize < 224, TLSv1, TLSv1.1
```

5.2.3 IBM Security Key Lifecycle Manager hardening

The configuration steps for changing IBM Security Key Lifecycle Manager to use TLS 1.2 to activate the Federal Information Processing Standard (FIPS) and to enable the requirement to use SHA2 signatures are shown here:

1. Go to the IBM Security Key Lifecycle Manager config directory as shown in Example 5-14.

Example 5-14 IBM Security Key Lifecycle Manager config directory

```
[root@sklma ~]# cd /opt/IBM/WebSphere/AppServer/products/sklm/config
```

2. Edit the SKLMConfig.properties file as shown in Example 5-15.

Example 5-15 Edit SKLMConfig.properties

```
[root@sklma ~]# vi SKLMConfig.properties
```

3. IBM Security Key Lifecycle Manager V4.0 runs in TLS 1.2 mode by default. Ensure that the line `TransportListener.ssl.protocols` is present, as shown in Example 5-16.

Example 5-16 SKLMConfig.properties changes 1

```
TransportListener.ssl.protocols=TLSv1.2
```

4. Add the following lines to SKLMConfig.properties as shown in Example 5-17.

Example 5-17 SKLMConfig.properties changes 2

```
requireSHA2Signatures=true
fips=on
```

5. Restart the IBM Security Key Lifecycle Manager Server process as shown in Example 5-18.

Example 5-18 IBM Security Key Lifecycle Manager restart

```
Navigate to:
[root@sklma ~]# cd /opt/IBM/WebSphere/AppServer/bin/

To stop the server, run:
```

```
[root@sklma bin]#  ./stopServer.sh server1 -username wasadmin -password
MyPassw0rd

To start the server, run:
[root@sklma bin]#  ./startServer.sh server1
```

5.3 Migrating from an earlier version of IBM Security Key Lifecycle Manager to IBM Security Key Lifecycle Manager V4.0

IBM Security Key Lifecycle Manager provides two mechanisms to migrate from a former version of IBM Security Key Lifecycle Manager:

- Inline migration: When IBM Security Key Lifecycle Manager V4.0 is installed on the same machine where an earlier version of IBM Security Key Lifecycle Manager is installed. Migration does not remove the earlier version of IBM Security Key Lifecycle Manager.
- Cross migration: When IBM Security Key Lifecycle Manager V4.0 is installed on a different machine than the earlier version of IBM Security Key Lifecycle Manager and data is migrated by using migration scripts.

For more information about supported paths, see IBM Security Key Lifecycle Manager 4.0.0.

5.3.1 Migration by using inline migration

With the inline migration method, you install IBM Security Key Lifecycle Manager V4.0 on the same machine where an earlier version of IBM Security Key Lifecycle Manager is installed. In this method, data from an earlier version of IBM Security Key Lifecycle Manager is migrated to IBM Security Key Lifecycle Manager V4.0 during the installation process. So, the installation process performs two tasks:

1. Installs IBM Security Key Lifecycle Manager V4.0.
2. Migrates data from an earlier version of IBM Security Key Lifecycle Manager to IBM Security Key Lifecycle Manager V4.0.

After the installation completes, the earlier version of IBM Security Key Lifecycle Manager remains on the machine. Also, there will be two folders in the `/opt/IBM/WebSphere` folder where the `AppServer` folder contains the data of the earlier version of IBM Security Key Lifecycle Manager and the `AppServer_1` folder contains the data of IBM Security Key Lifecycle Manager V4.0.

To install IBM Security Key Lifecycle Manager in GUI mode, complete the following steps:

1. Make sure that the supported operating system that is used for installation has the required libraries and operating system kernel settings. For more information, see IBM Security Key Lifecycle Manager V4.0.0.
2. Extract both installation images (disk1 and disk2) that you previously obtained from IBM Passport Advantage. Use the same subfolder to extract the images, as shown in Example 5-19. Keep the paths as short as possible.

Example 5-19 Extracted installation images

```
[root@sklmdemo74 sklm]# ls -l
drwxr-xr-x. 8 root root 4096 Nov 29  2019 disk1
drwxr-xr-x. 3 root root   35 Nov 29  2019 disk2
```

3. Go to the IBM Security Key Lifecycle Manager directory to which the file was extracted. Go to `disk1` and start the installation in GUI mode by running **`launch.sh`** on non-Windows systems and **`launch.bat`** on Windows systems. Figure 5-11 shows an example of starting the installation in GUI mode.

```
[root@sklmdemo74 sklm]# cd disk1
[root@sklmdemo74 disk1]# ls
ad                                SKLM_Silent_Linux_Mig_25_Resp.xml
diskTag.inf                       SKLM_Silent_Linux_Mig_26_Resp.xml
documentation                     SKLM_Silent_Linux_Mig_27_Resp.xml
im                                SKLM_Silent_Linux_Mig_301_Resp.xml
install.sh                        SKLM_Silent_Linux_Mig_30_Resp.xml
launchpad.sh                      SKLM_Silent_Linux_Resp.xml
md                                SKLM_Uninstall_Linux_Resp.xml
PRS                               TKLMPasswordPolicy.xml
silent_install.sh                 toc
silent_install_withoutIM.sh       uninstallSKLM_linux.sh
silent_uninstallSKLM_linux.sh
[root@sklmdemo74 disk1]# ./install.sh
Default locale is English
Using default value for locale
Install Registry exists
pluginName = Equinox Launcher Linux X86_64 Fragment
64 bit version of IBM Installation Manager installed on the system.
Installing IBM Security Key Lifecycle Manager 4.0
```

Figure 5-11 Starting the migration in GUI mode

4. In the Migration confirmation window (Figure 5-12), click **Continue**.

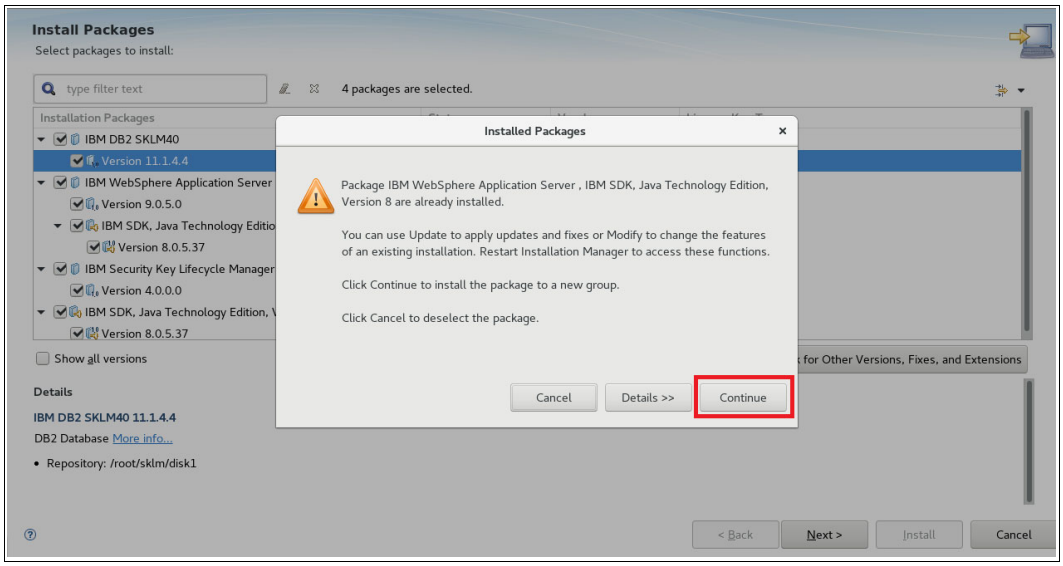


Figure 5-12 Migration confirmation window

5. In the welcome window, information regarding all the packages is shown (Figure 5-13 on page 87). An inline migration-based installation of IBM Security Key Lifecycle Manager also installs IBM Db2, IBM WebSphere Application Server with IBM Java, and IBM Security Key Lifecycle Manager. IBM Installation Manager is not installed again. Click **Next**.

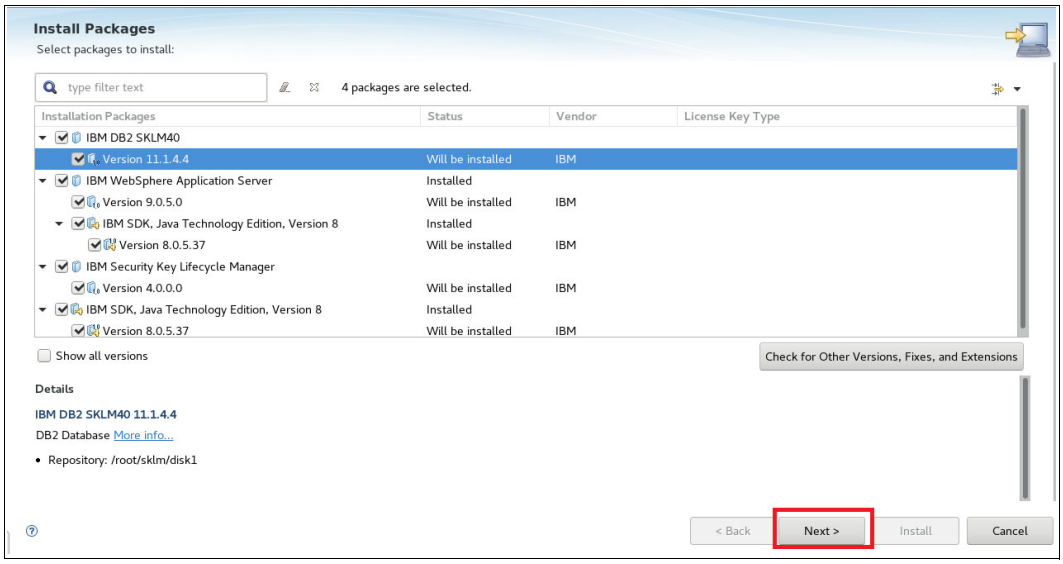


Figure 5-13 Migration welcome window

6. Read the license agreements and accept licenses by selecting **I accept the terms in the license agreements** option (Figure 5-14). Click **Next**.

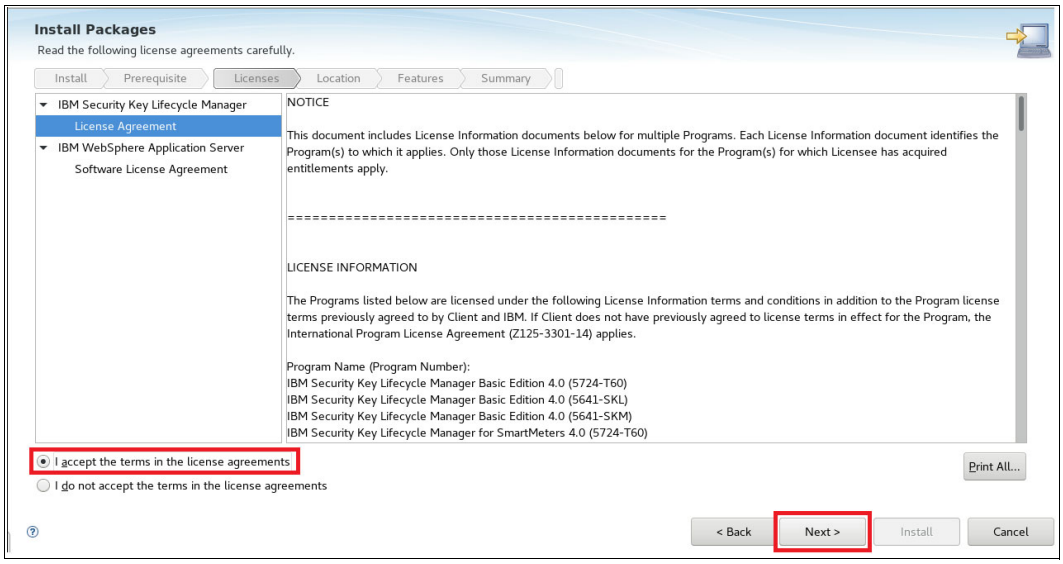


Figure 5-14 Migration License Agreement window

7. Select the installation location for IBM Db2, IBM WebSphere Application Server, and IBM Security Key Lifecycle Manager (Figure 5-15). Click **Next**.

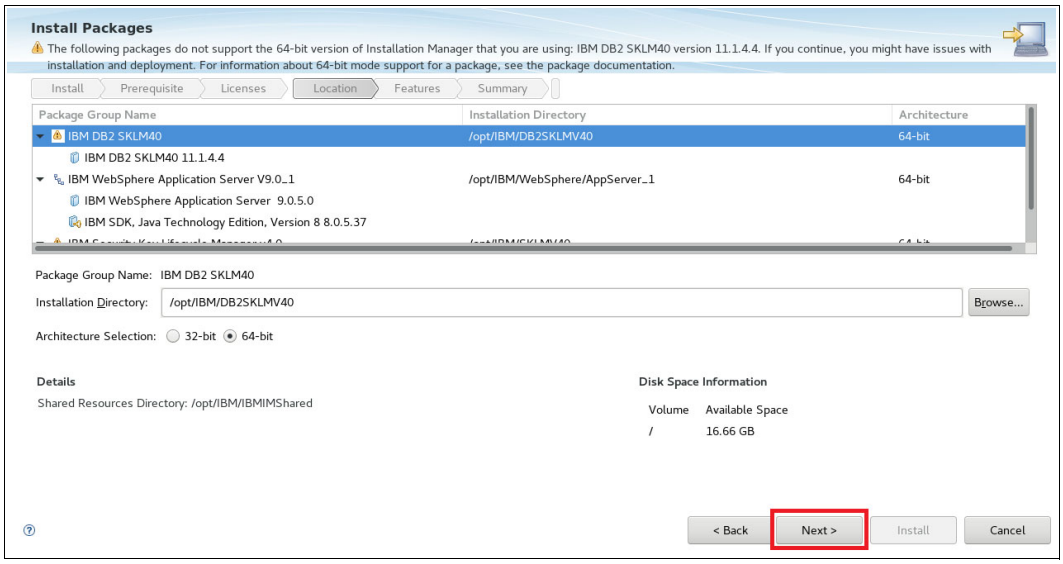


Figure 5-15 Migration installation location window

8. In the Translation packages window (Figure 5-16), click **Next**. You can ignore the warning that is shown on the top of the page regarding the 64-bit version in use.

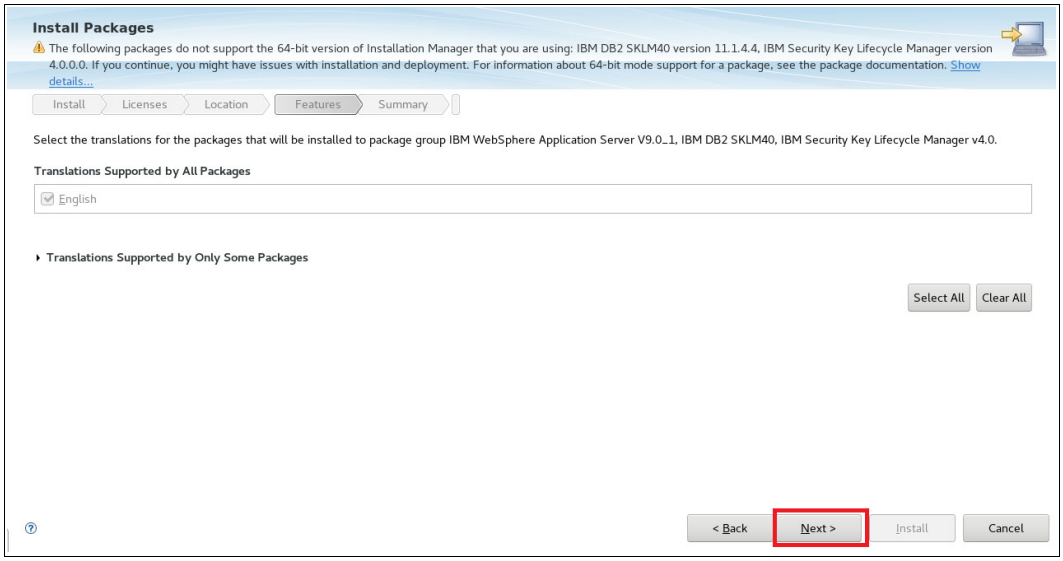


Figure 5-16 Migration translation package window

9. In the Features selection window (Figure 5-17), make sure that all the options are selected. Click **Next**.

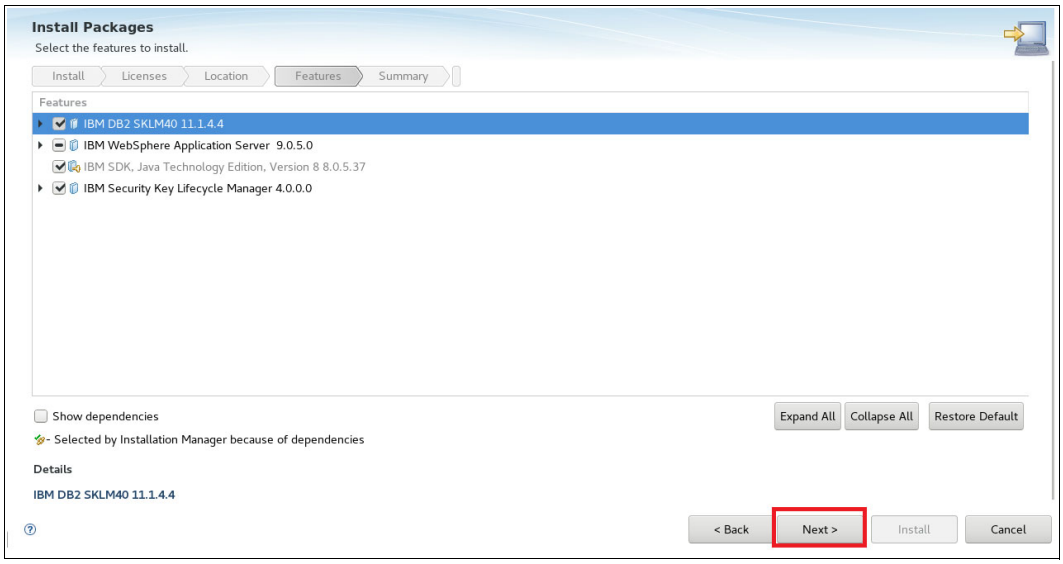


Figure 5-17 Migration feature selection window

10. In the IBM Db2 configuration window (Figure 5-18), provide an appropriate name for the new Db2 administrator user. This user runs the IBM Security Key Lifecycle Manager application and Db2 database postinstallation. Make sure to provide a password that matches the operating system password requirements. Click **Next**.

> **Important:** Note the Db2 username and password because they are required during operations like upgrades and migration.

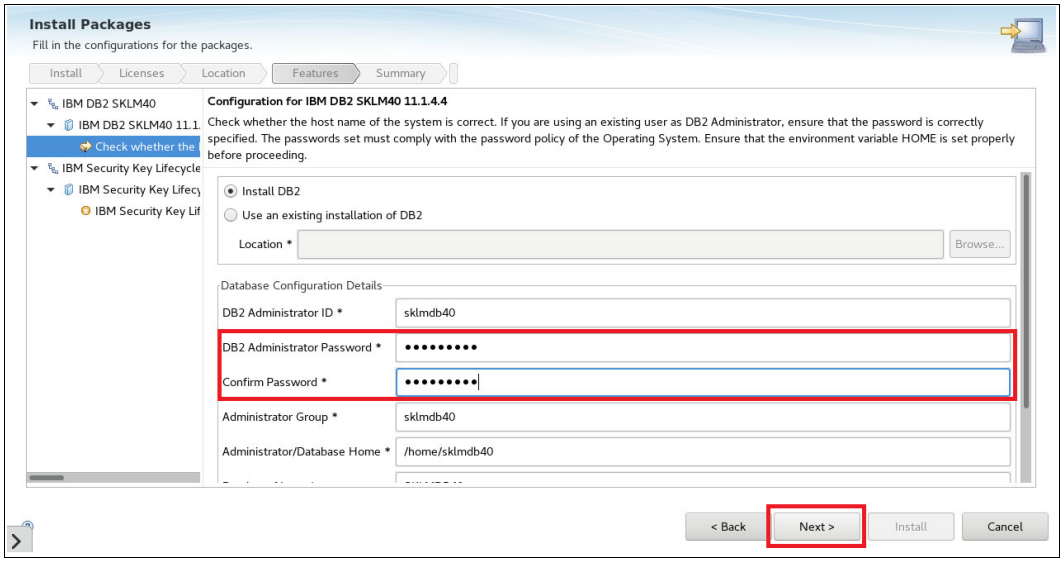


Figure 5-18 Migration Db2 configuration window

11. In the IBM Security Key Lifecycle Manager configuration window (Figure 5-19), provide the following details.

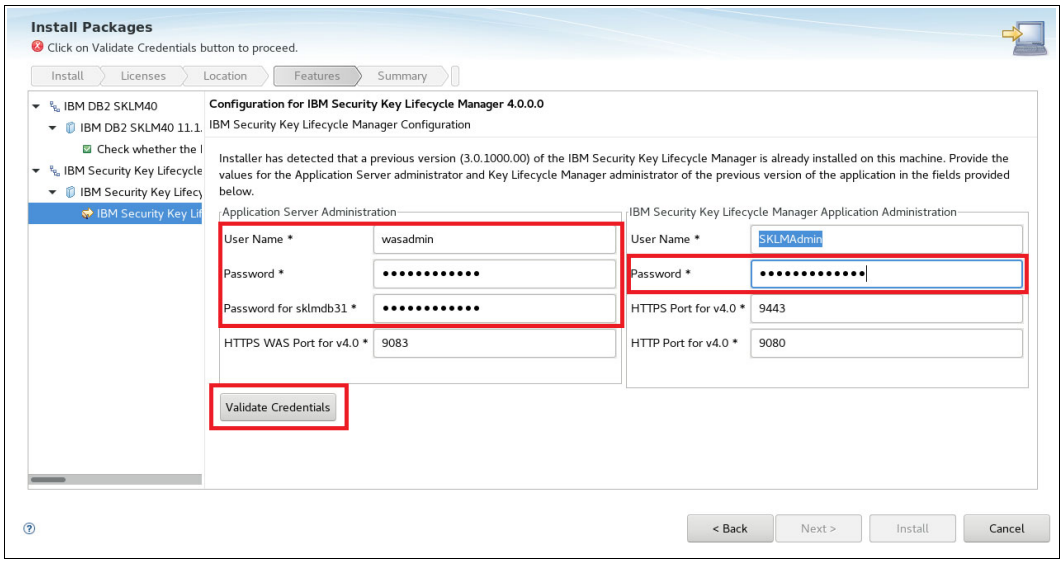


Figure 5-19 Migration IBM Security Key Lifecycle Manager configuration screen to Validate Credentials

– The password for the `wasadmin` user of IBM Security Key Lifecycle Manager V4.0

– The password for the `wasadmin` user of the earlier version of IBM Security Key Lifecycle Manager

– Appropriate password to set for `SKLMAdmin` user in IBM Security Key Lifecycle Manager V4.0

Ensure that the passwords follow a proper password policy. For more information, see IBM Security Key Lifecycle Manager 4.0.0.

Click **Validate Credentials** to validate the passwords.

> **Important:** Note the usernames and passwords because they are required for many operations, such as IBM Security Key Lifecycle Manager user creation, restarting IBM Security Key Lifecycle Manager services, configuring a Multi-Master cluster, and performing upgrades and migrations. Also, note the port numbers.

12. If there are no errors (Figure 5-20 on page 91), click **Next**.

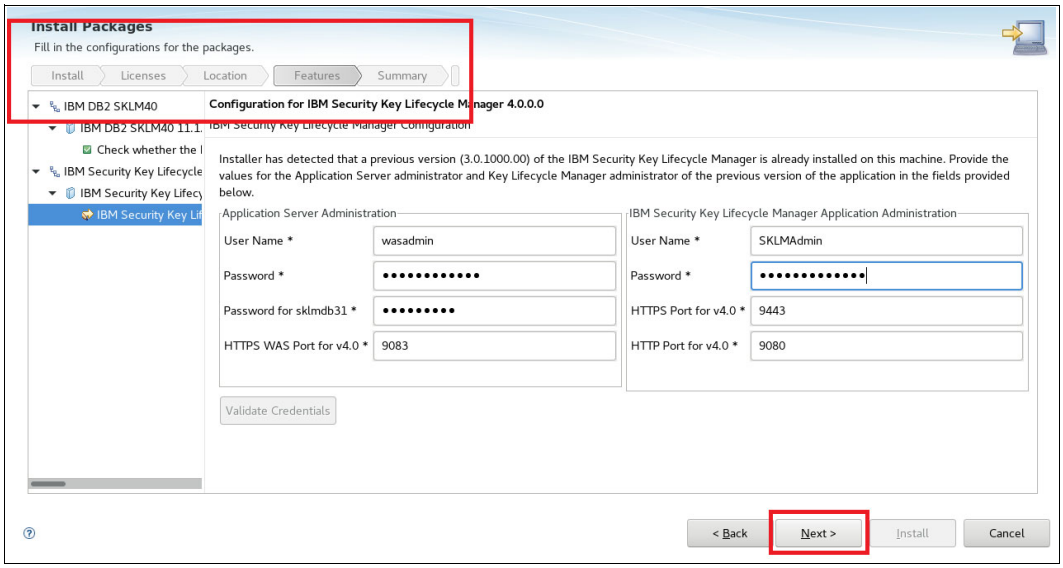


Figure 5-20 Migrate IBM Security Key Lifecycle Manager configuration window

13. In the preinstallation summary window (see Figure 5-21), review your choices before you install the product package. To change a selection, click **Back** to return to your selections. Click **Install** to begin the installation.

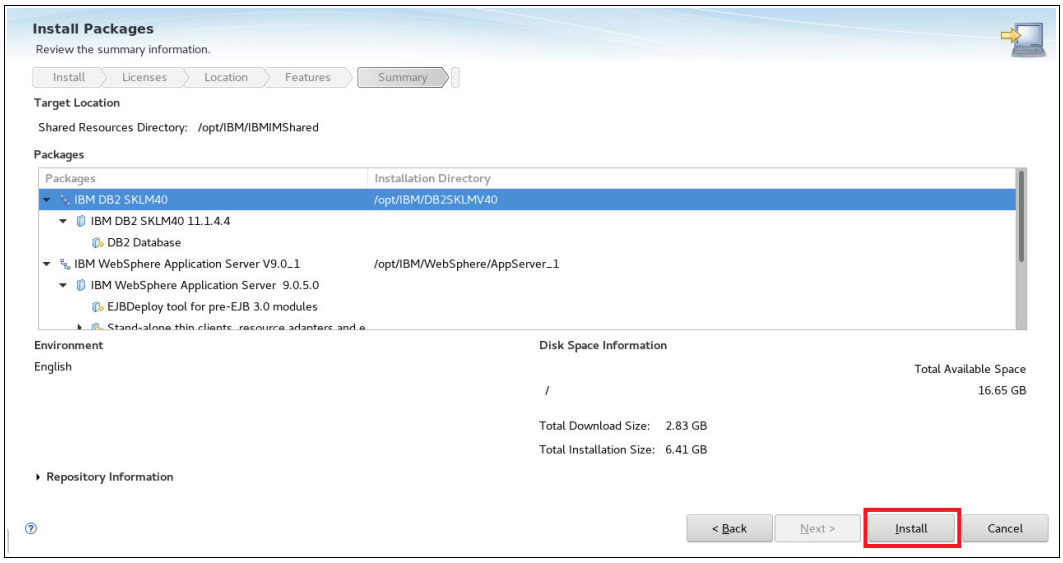


Figure 5-21 Migration summary window

14. A progress indicator shows the percentage of the installation that is complete. When the installation process completes, a message confirms the completion of the process (Figure 5-22). Click **Finish** to complete the installation and close the installation wizard.

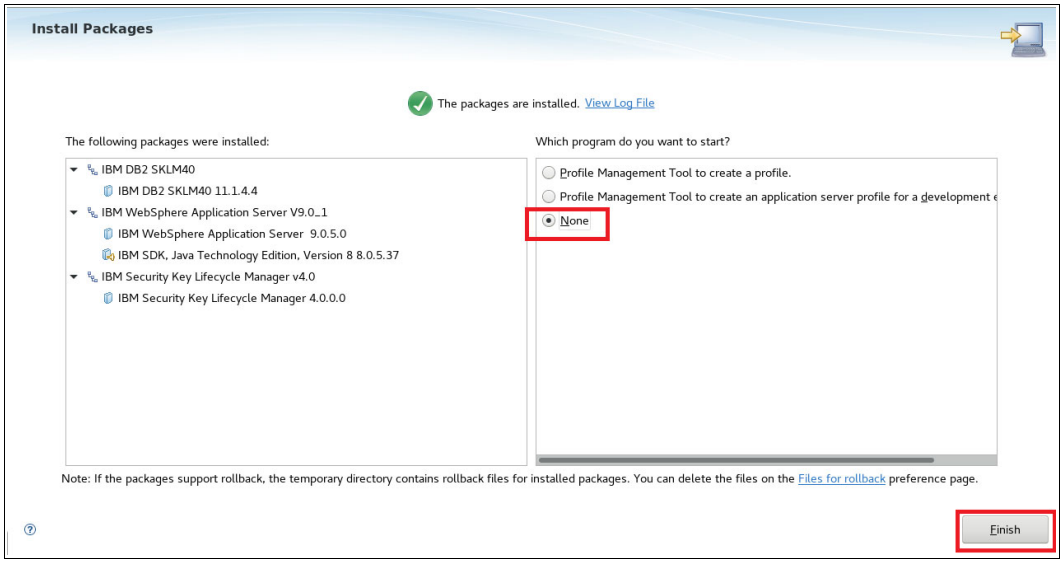


Figure 5-22 Migration complete window

5.3.2 Migration by using cross migration

The cross migration method is the preferred migration method because there is not an earlier version of IBM Security Key Lifecycle Manager on the same machine.

To migrate data from an earlier version of IBM Security Key Lifecycle Manager to IBM Security Key Lifecycle Manager V4.0, complete the following steps:

1. Install IBM Security Key Lifecycle Manager V4.0 on a machine different than the machine where an earlier version of IBM Security Key Lifecycle Manager is installed. For more information, see 5.1, “Installing IBM Security Key Lifecycle Manager V4.0” on page 74.
2. Take a backup from the earlier version of IBM Security Key Lifecycle Manager. For more information, see “Creating a backup” on page 102.
3. A backup file (.jar) is created in the `/opt/IBM/WebSphere/AppServer/products/sklm/data` folder. Copy this backup file to the new machine where IBM Security Key Lifecycle Manager V4.0 is installed. Place the backup file in the `/opt/IBM/WebSphere/AppServer/products/sklm/data` folder.
4. Restore the backup on the IBM Security Key Lifecycle Manager V4.0 machine. For more information, see “Restoring the backup” on page 104.

5.4 Key manager setup

This section describes setting up external key managers like IBM Security Lifecycle Manager and other external key managers.

5.4.1 Configuring IBM Security Key Lifecycle Manager V4.0

IBM Security Key Lifecycle Manager Version V4.0 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 and Security Guardium Key Lifecycle Manager Version 2.x.

Microsoft Windows, Linux, Linux on IBM Z, and IBM AIX operating systems are supported.

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 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.

IBM Security Key Lifecycle Manager includes the following enhancements compared to its predecessors:

- Support for only 64-bit platforms.
- TLS 1.2 (default) 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.
- High availability support by using replication and a Multi-Master cluster.
- DS8000 Key Management Interoperability Protocol (KMIP) support for data at rest (DAR) encryption, Transparent Cloud Tiering (TCT) encryption, and IBM Fibre Channel Endpoint Security. For IBM Fibre Channel Endpoint Security encryption, IBM Security Key Lifecycle Manager V3.0.1.3 is required.

This section describes the procedure to configure IBM Security Key Lifecycle Manager to serve keys to 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 servers must be relatively synchronized.

For more information, see IBM Knowledge Center.

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.
- 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 compliant with NIST SP 800-131a, see 5.9, “NIST SP 800-131a requirements for key servers” on page 194. Modify the IBM Security Key Lifecycle Manager configuration.

Before you configure devices such as the DS8000, the IBM Security Key Lifecycle Manager requires some initial basic configuration:

- IBM Security Key Lifecycle Manager Multi-Master:
  - You must create SSL/KMIP communication certificates on the primary master.
  - Ensure that all other servers joining the Multi-Master cluster are clean servers without any keys or certificates.
  - Ensure that the operating system kernel parameters are correctly set. For other requirements for Multi-Master, see IBM Security Key Lifecycle Manager 4.0.0.
- IBM Security Key Lifecycle Manager Master-Clone Replication:
  - You must create SSL/KMIP communication certificates before setting up the replication cluster.
  - You must establish a replication between the Security Guardium Key Lifecycle Manager servers, or set up a disaster recovery solution through backup and restore.

Consider the following definitions:

- SSL/KMIP Certificate: The IBM Security Key Lifecycle Manager servers and its devices require a SSL/KMIP certificate for secure communication between these entities:
  - Between the servers themselves (when you use replication or in Multi-Master environments).
  - Between the servers and the device, such as the DS8000 HMC. You always must create an SSL/KMIP certificate. Self-signed and third-party (CA) signed certificates are supported. The choice is based on the customer's requirements/policy regarding the types of certificate and authority.
- Backup and Restore: The IBM Security Key Lifecycle Manager creates cross-platform backup files independently of operating systems and directory structure of the server. You can restore the backup files to an operating system that is different from the one it was backed up from. For example, you can restore a backup file that is taken on a Linux system and restore it on a Windows system. Using backup and restore is not supported in KMIP environments or when you use TCT encryption.
- Replication: The DS8000 does not require replication between the IBM Security Key Lifecycle Manager servers when you use IBM Proprietary Protocol through TLS for DAR encryption. However, setting it up prepares the servers for future functions, where replication will be required. Replication is not supported in KMIP environments or when you use TCT encryption.
- Multi-Master: The IBM Security Key Lifecycle Manager instances with Multi-Master configuration achieve continuous availability of data across multiple IBM Security Key Lifecycle Manager deployment environments. This function is new in Version 3 and is required for a DS8880 R8.5, DS8900F R9, and later that uses the KMIP protocol for TCT encryption and encryption of data in flight (EDIF). Multi-Master is the preferred option to use.

Note: Consider that if you move from an older IBM Security Key Lifecycle Manager version to IBM Security Key Lifecycle Manager 3.0 or later, you must use the backup and restore operations for earlier versions of IBM Security Key Lifecycle Manager. Do not continue with creating certificates after that process is complete.

Creating an SSL/KMIP server certificate

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 (Figure 5-23) at the following address:

   `https://<ip address>:<port>/ibm/SKLM/jsp/Main.jsp`

   Important: IBM Security Key Lifecycle Manager V4.0 by default uses port 9443 for the IBM Security Key Lifecycle Manager GUI and REST APIs.

Figure 5-23 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, select **Advanced Configuration** → **Server certificates**, and then **Add**, as shown in Figure 5-24.

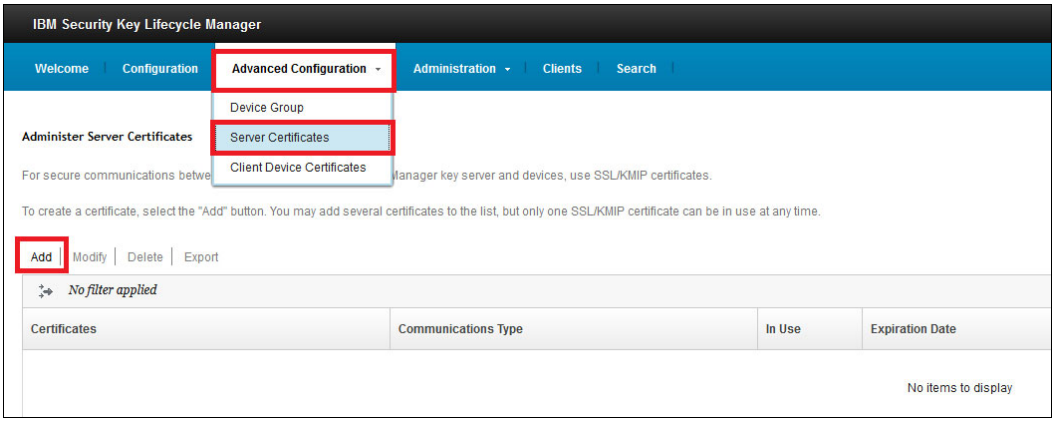


Figure 5-24 IBM Security Key Lifecycle Manager Welcome window

You can create a self-signed server certificate (“Option 1: Creating a self-signed SSL/KMIP server certificate”) or create a certificate that is signed by a third-party provider (“Option 2: Creating a certificate that is signed by a third-party provider” on page 97).

Option 1: Creating a self-signed SSL/KMIP server certificate

This example uses a self-signed certificate. If you use a certificate that is signed by a third-party provider, that is, a certificate authority (CA), go to “Option 2: Creating a certificate that is signed by a third-party provider” on page 97.

Although using an existing certificate from the keystore is possible, it is not a best practice to use the same certificate that encrypts disk data to also protect communication.

Complete the following steps:

1. To create a self-signed SSL certificate, log in to the IBM Security Key Lifecycle Manager Server GUI, and select **Advanced Configuration** → **Server Certificates**, as shown in Figure 5-25.

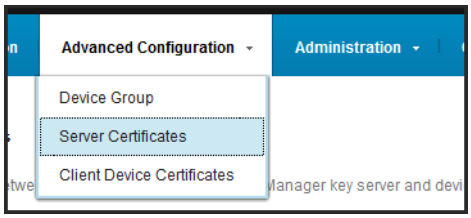


Figure 5-25 Server Certificate

2. Provide a certificate label and a certificate description when you create the certificate.

   Figure 5-26 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 completed when creating the certificate. The validity period determines how long the certificate is valid. The RSA algorithm uses the 2048-bit key.

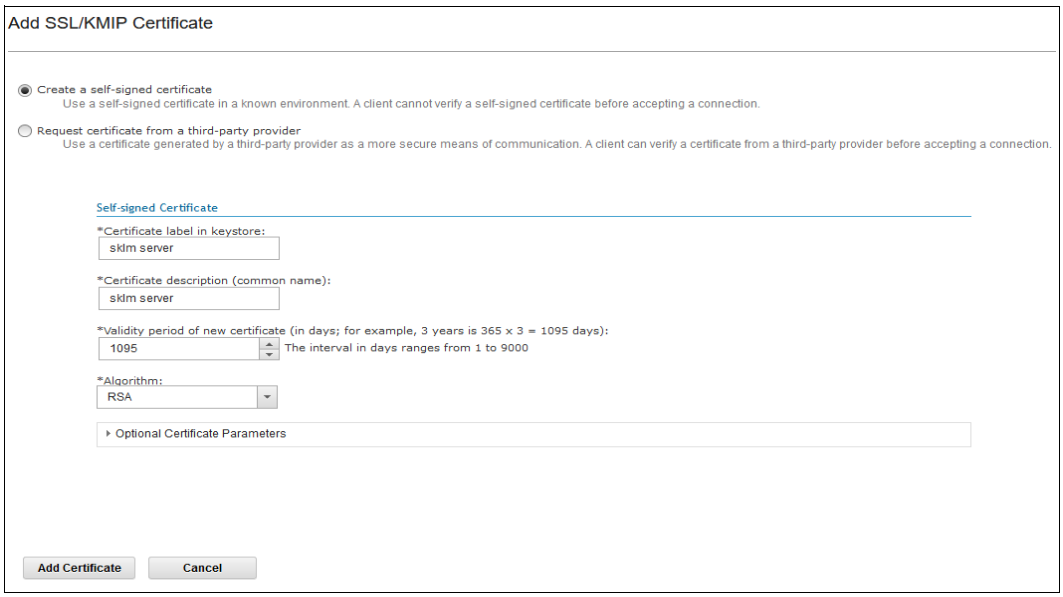


Figure 5-26 SSL/KMIP for Key Serving window

3. After you complete all fields, click **Add Certificate** to create and add the certificate. Figure 5-27 shows that the SSL certificate is created.

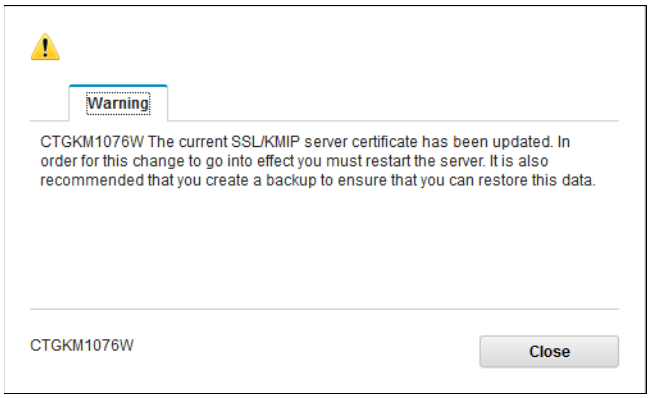


Figure 5-27 SSL certificate created successfully

4. Restart the IBM Security Key Lifecycle Manager. Select **sklmadmin** in the upper right, and then click **Restart Server**, as shown in Figure 5-28.

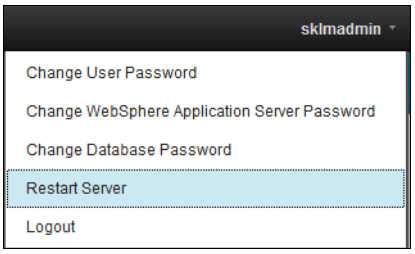


Figure 5-28 Restart the IBM Security Key Lifecycle Manager server

5. Go to “Exporting the SSL/KMIP server certificate” on page 100.

Option 2: Creating a certificate that is signed by a third-party provider

This example uses a certificate that is signed by a third-party CA.

Although using an existing certificate from the keystore is possible, but using the certificate that is used to encrypt disk data to also protect communication is not a best practice.

To have a certificate that is signed by a third-party CA, it is required to start with a certificate request.

Complete the following steps:

1. To create a third-party signed SSL certificate request, log in to the IBM Security Key Lifecycle Manager Server GUI, and select **Advanced Configuration** → **Server Certificates**, as shown in Figure 5-29.

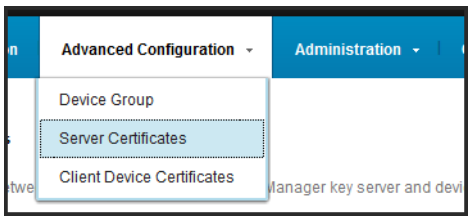


Figure 5-29 Server Certificate

2. Select **Request certificate from a third-party provider** and complete the fields, as shown in Figure 5-30. Then, click **Add Certificate**.

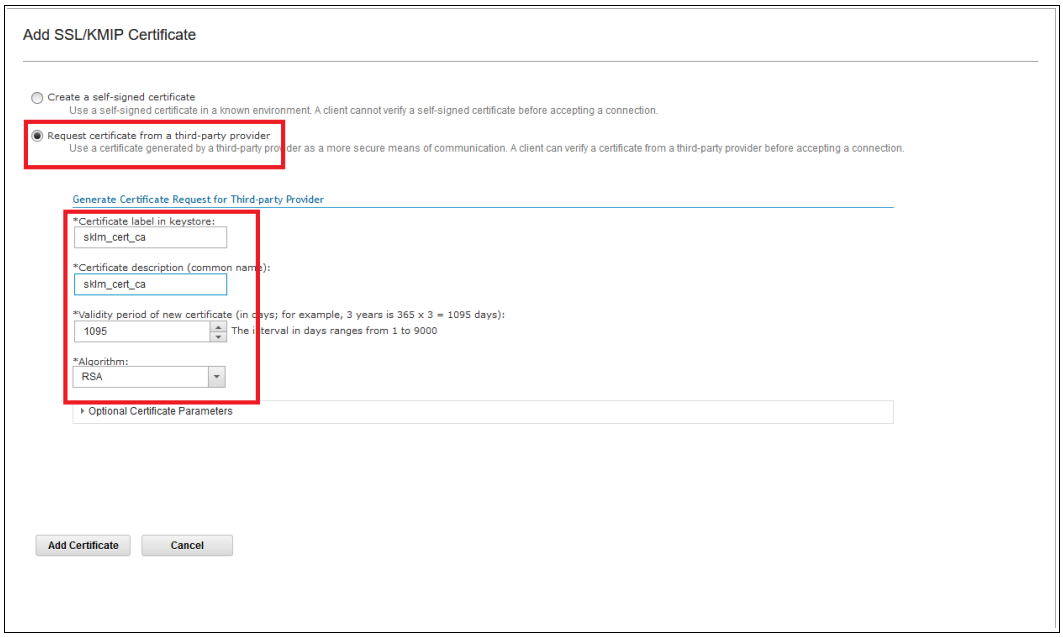


Figure 5-30 IBM Security Key Lifecycle Manager Server certificate from a third party

3. The certificate request is now active, but the status is Pending (Figure 5-31).

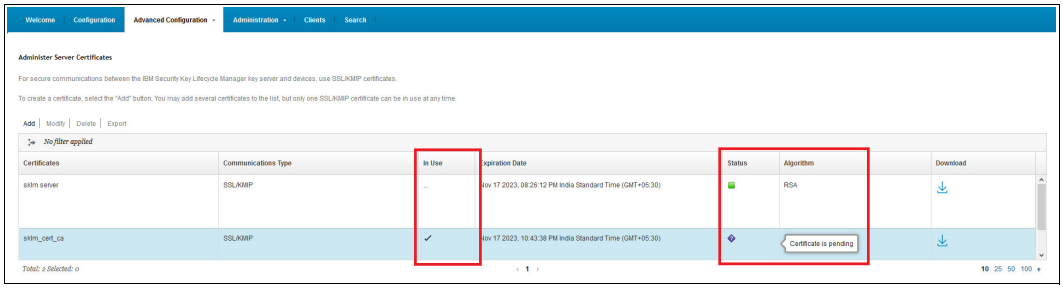


Figure 5-31 Pending certificate

4. The certificate request file is in the directory that is shown in Example 5-20. Manually send this certificate request file to a CA and get it signed. You must import the signed certificate to IBM Security Key Lifecycle Manager later.

> **Note:** If inline migration is performed on this system, then you find the `AppServer` and `AppServer_1` folders in the `/opt/IBM/WebSphere` folder, where the `AppServer` folder is holding data for the older IBM Security Key Lifecycle Manager server and `AppServer_1` holds data for IBM Security Key Lifecycle Manager V4.0.

Example 5-20 CSR location

```
[[root@sklmdemo74 data]# cd /opt/IBM/WebSphere/AppServer_1/products/sklm/data
[root@sklmdemo74 data]# ls -l
-rw-r--r--. 1 sklmdb40 sklmdb40  958 Nov 17 22:23 201117224338-sklm_cert_ca.csr
drwx------. 2 sklmdb40 root        6 Nov 12 02:14 agent
drwx------. 2 sklmdb40 root        6 Nov 12 02:14 restore
```

5. Transfer the signed SSL/KMIP certificate that returned from the CA to `/opt/IBM/WebSphere/AppServer/products/sklm/data/` and import it by clicking **Third-party certificates pending import** under **Action Items** on the welcome page, as shown in Figure 5-32.

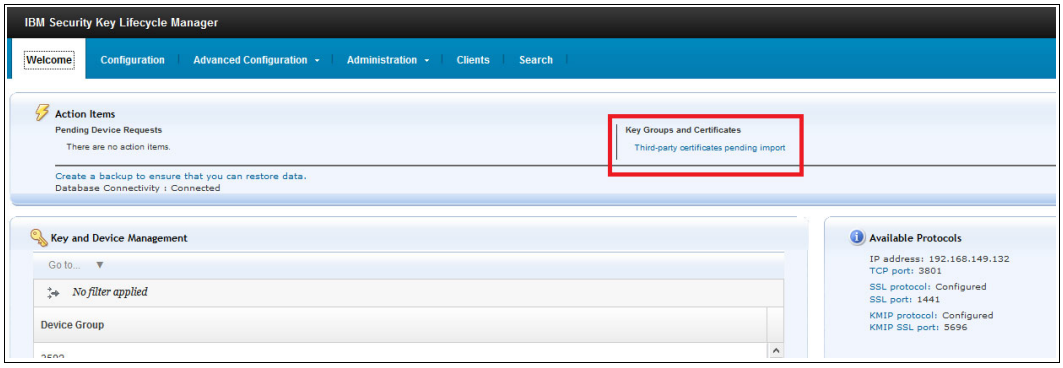


Figure 5-32 Welcome page: Third-party certificate pending import

6. On the Import page, select the pending certificate, right-click it, and select **Import**, as shown in Figure 5-33.

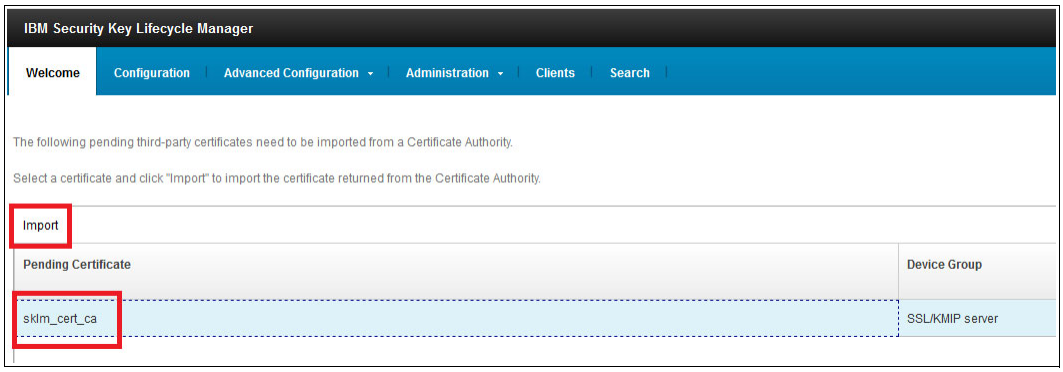


Figure 5-33 Select the signed certificate for import

7. Browse for the signed certificate and click **Import**, as shown in Figure 5-34.

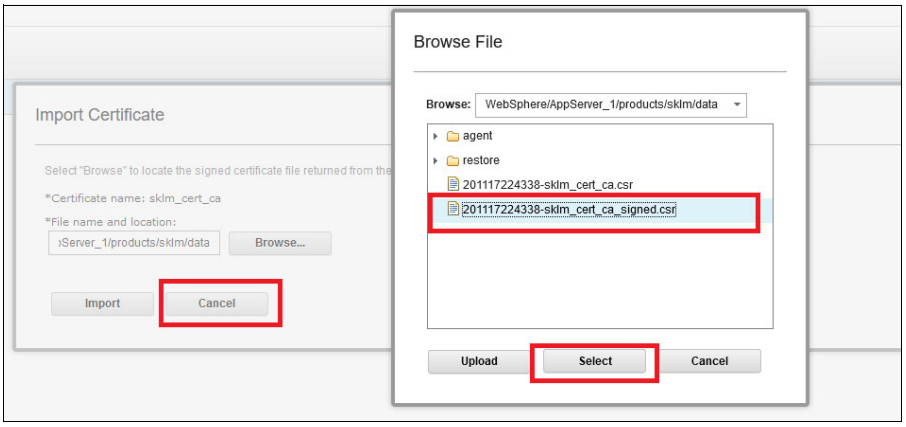


Figure 5-34 Import the signed certificate

8. After the signed SSL/KMIP certificate is imported, the status of the certificate changes to valid, as shown in Figure 5-35.

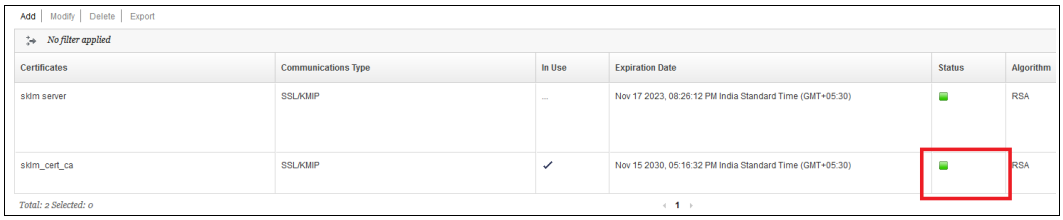


Figure 5-35 Valid certificate

9. Restart the IBM Security Key Lifecycle Manager. Select **sklmadmin** in the upper right, and then click **Restart Server**, as shown in Figure 5-36.

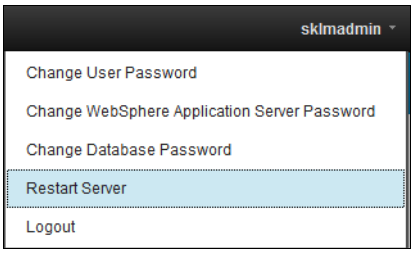


Figure 5-36 Restart IBM Security Key Lifecycle Manager Server

Exporting the SSL/KMIP server certificate

The SSL/KMIP Server certificate for secure communication between the IBM Security Key Lifecycle Manager Server and the DS8000 must be exported to the IBM Security Key Lifecycle Manager servers local hard disk drive (HDD) and transferred to the DS8000 in a later step.

Complete the following steps to export the SSL/KMIP Server certificate:

1. Log in to any IBM Security Key Lifecycle Manager server as SKLMAdmin and select **Advanced Configuration** → **Server Certificates**, as shown in Figure 5-37.

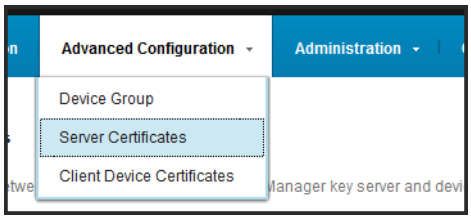


Figure 5-37 Server Certificates

2. Highlight the previously created certificate, and then click the **Download** icon, as shown in Figure 5-38.

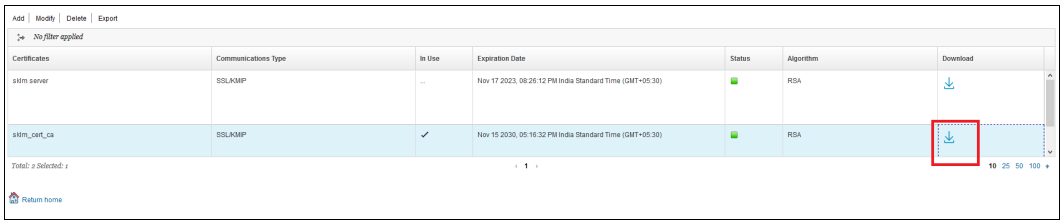


Figure 5-38 Export and download the certificate

3. Click **Download** to download the exported certificate on to your local machine, as shown in Figure 5-39.

Figure 5-39 Download exported certificate

The SSL/KMIP Certificate is now exported. Transfer it to a destination that is accessible when activating encryption in the DS8000. The Certificate must be specified during activation to ensure secure communication between the servers and the DS8000.

Only continue with backup and restore as described in “Backup and restore” on page 101 if Multi-Master will *not* be used. In Multi-Master configurations, the certificate is transferred automatically to the Standby master when you set it up. Continue to set up a Multi-Master environment as described in “Setting up a Multi-Master environment with two IBM Security Key Lifecycle Manager key servers” on page 112.

The Multi-Master environment can be set up in new environments only.

Backup and restore

The IBM Security Key Lifecycle Manager creates cross-platform backup files independently of operating systems and directory structure of the server. You can restore the backup files to an operating system that is different from the one it was backed up from. For example, you can restore a backup file that is taken on a Linux system and restore it on a Windows system.

Important: Using backup and restore is not supported in Multi-Master environments when you use DAR and TCT encryption with KMIP.

Creating a backup

To create a backup, complete the following steps:

1. In the IBM Security Key Lifecycle Manager GUI, select **Administration** → **Backup and Restore**, as shown in Figure 5-40.

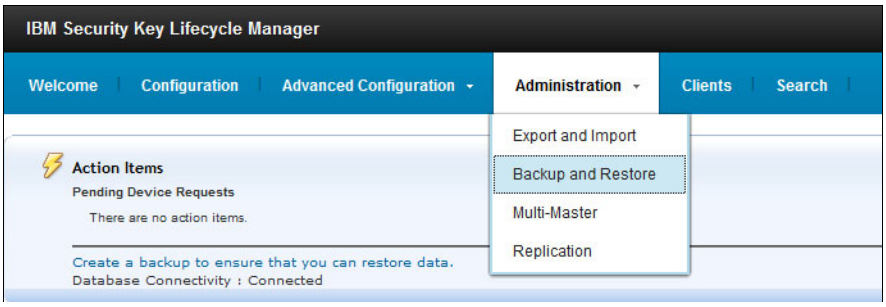


Figure 5-40 Backup and Restore menu

2. Under the **Backup and Restore** tab that is shown in Figure 5-41, next to the **Backup repository location** field, click **Browse** to select a path for the backup.

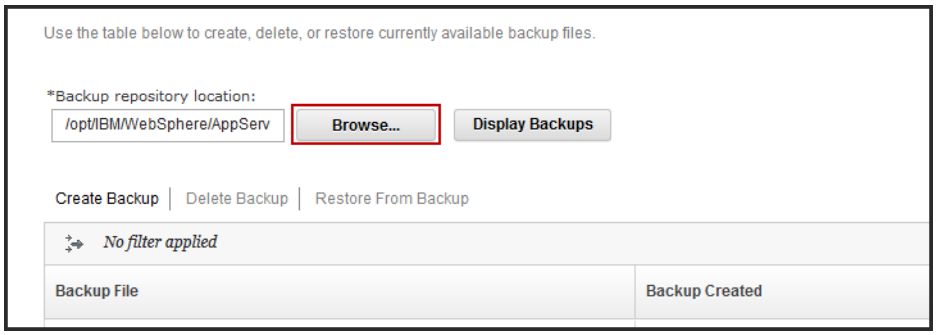


Figure 5-41 Backup and Restore directory

3. In the Create Backup window that is shown in Figure 5-42, enter a password for the backup, and then click **Create Backup**. This password is *required* to use the restore function later.

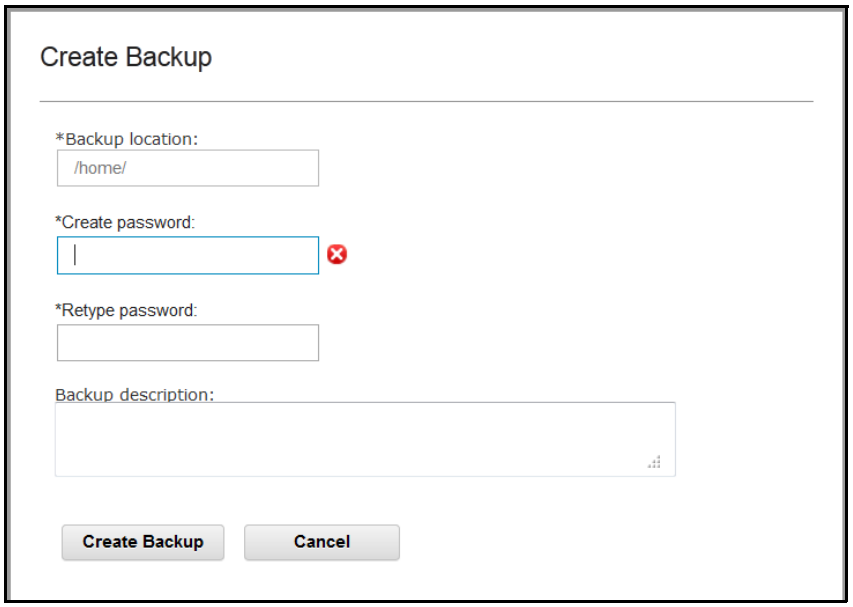


Figure 5-42 Create Backup window to enter the password and backup description

4. When you see the “successfully created” notice that is shown in Figure 5-43, click **Close**.

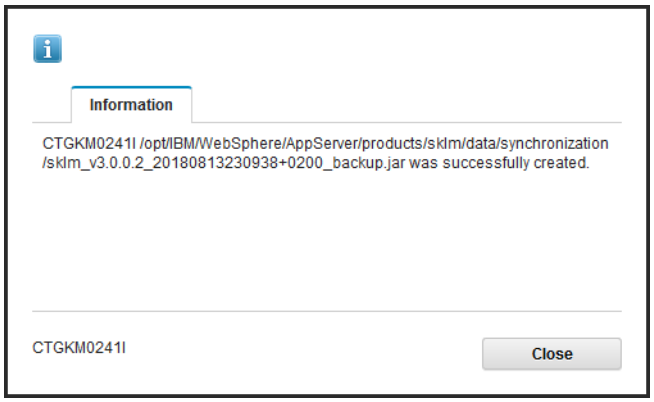


Figure 5-43 Backup was successfully completed

Because this server 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 by setting up remote replication between the IBM Security Key Lifecycle Manager Key servers (see “Setting up remote replication between IBM Security Key Lifecycle Manager key servers” on page 106) or by setting up a Multi-Master environment, as shown in “Setting up a Multi-Master environment with two IBM Security Key Lifecycle Manager key servers” on page 112.

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

- Initial configuration in non Multi-Master environments
- Adding keys or devices without synchronization

- Key or certificate replacement intervals without synchronization
- CA requests without synchronization

Transfer and restore the previously taken backup to all further IBM Security Key Lifecycle Manager Key servers that are installed. Always keep the latest backup in a secure place, such as off the key server on unencrypted storage.

Restoring the backup

To restore the backup, complete the following steps:

1. Log in to the IBM Security Key Lifecycle Manager and go to **Backup and Restore** and then click **Browse** to browse for the previously transferred backup file, as shown in Figure 5-44.

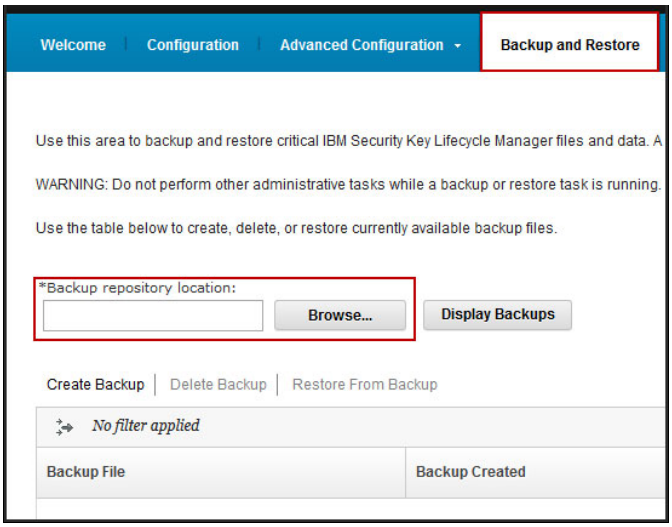


Figure 5-44 Browse for backup

2. Browse to the directory with the stored backup, as shown in Figure 5-45, and click **Select**.

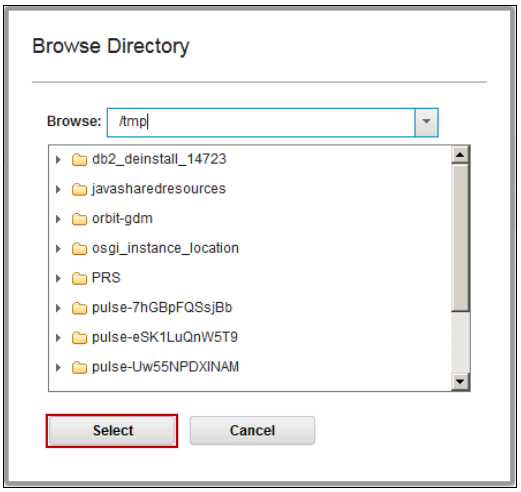


Figure 5-45 Browse to the directory

3. Click **Display Backups** to refresh. The backup appears as shown in Figure 5-46.

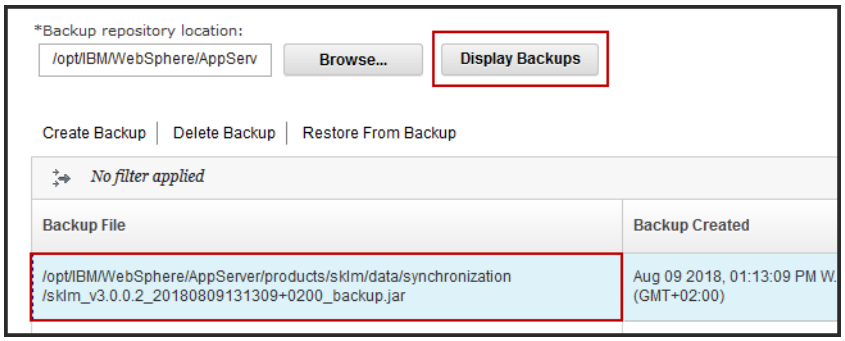


Figure 5-46 Display Backups

4. Select the backup and click **Restore From Backup**, as shown in Figure 5-47.

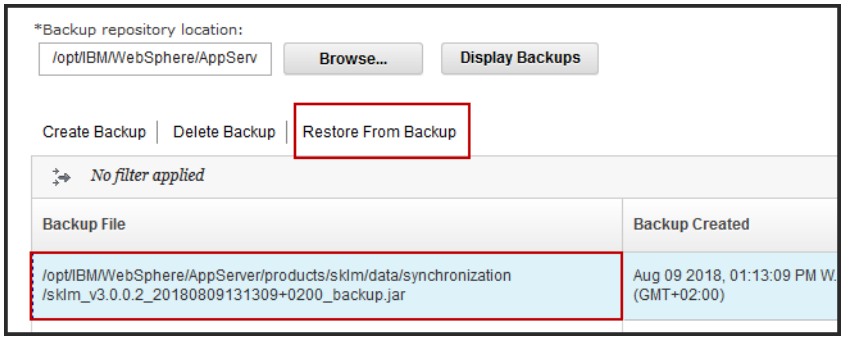


Figure 5-47 Highlight and restore

5. Enter the password that is specified during the backup and click **Restore Backup**, as shown in Figure 5-48.

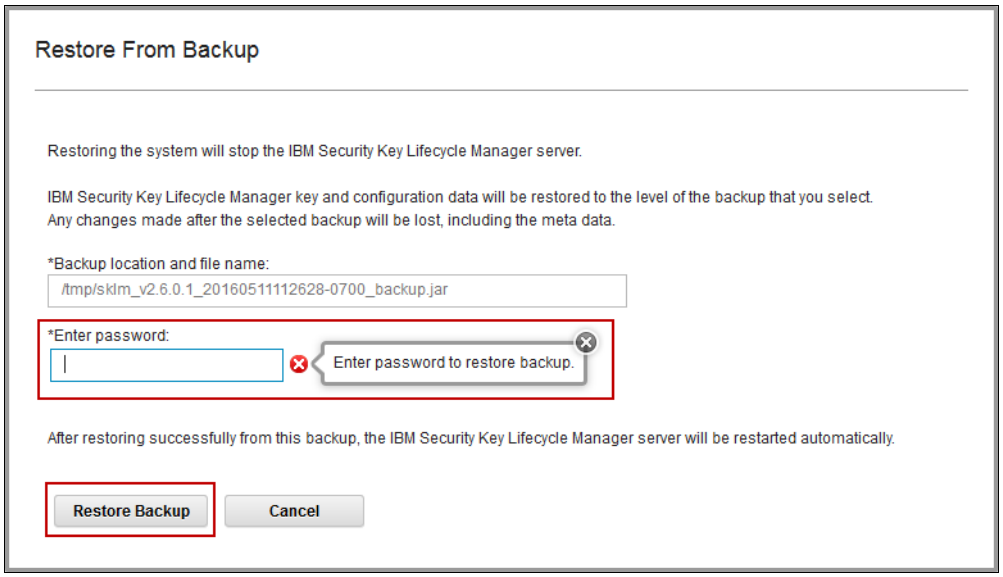


Figure 5-48 Restore backup: Password

6. The following dialog must be confirmed by clicking **OK**:

   "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 5-49.

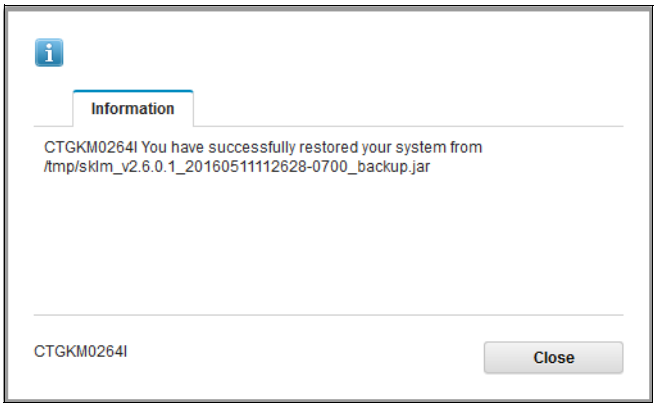


Figure 5-49 Restore Successful

The Security Guardium Key Lifecycle Manager is now set up to serve keys to the DS8000.

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, independently 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 hostname 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 IBM Security Key Lifecycle Manager V3.0.0.

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 **Administration** → **Replication**, as shown in Figure 5-50.

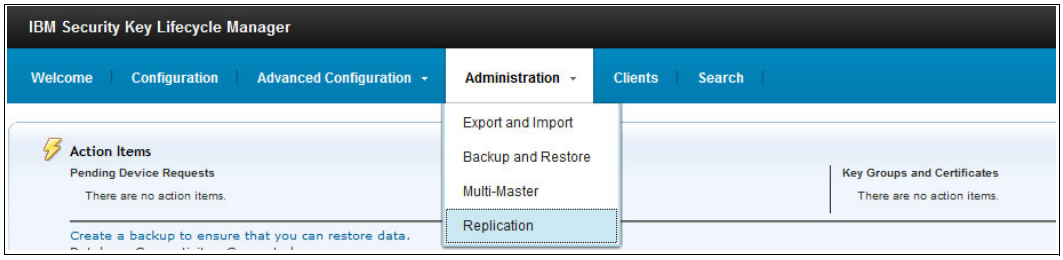


Figure 5-50 Replication menu

2. Change the value for one or more settings of the master server. Select **Master** and confirm “Are you sure to set up this IBM Security Key Lifecycle Manager as Master?” by clicking **OK**, as shown in Figure 5-51.

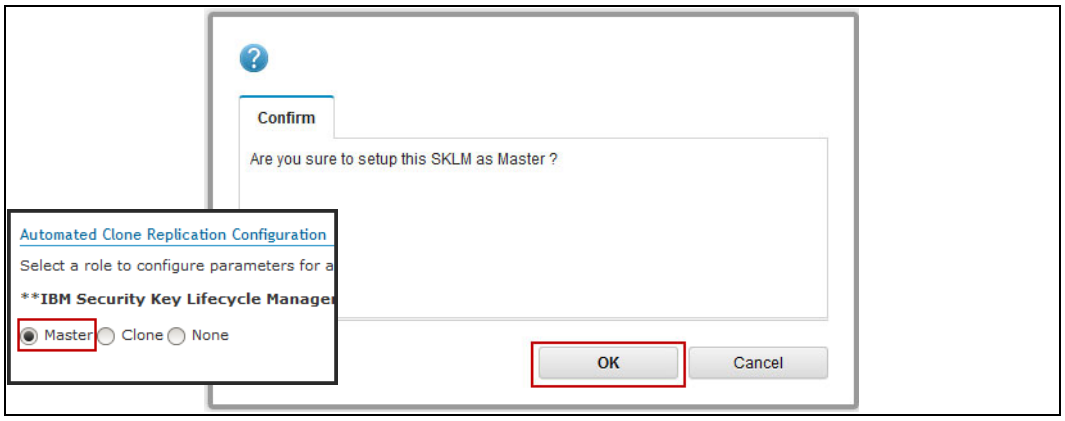


Figure 5-51 Confirm the master

3. Specify the following Basic Properties settings for the master. Then, click **OK**, as shown in Figure 5-52:
   - 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.

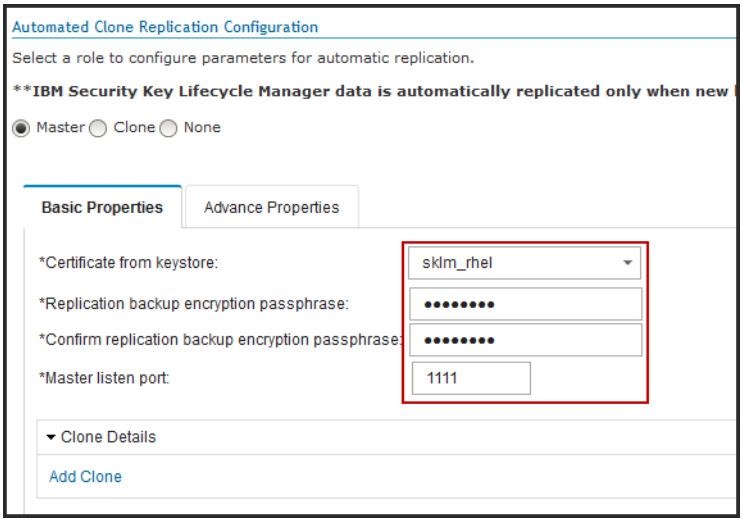


Figure 5-52 Basic Properties (Master)

Note: If you want to set up incremental replication, select the **Incremental Replication** check box 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 select **Configuration** → **Replication**, as shown in Figure 5-53.

Figure 5-53 Replication menu on the clone machine

2. Select **Clone** and change the ports under **Basic Properties** if required. Click **OK**, as shown in Figure 5-54.

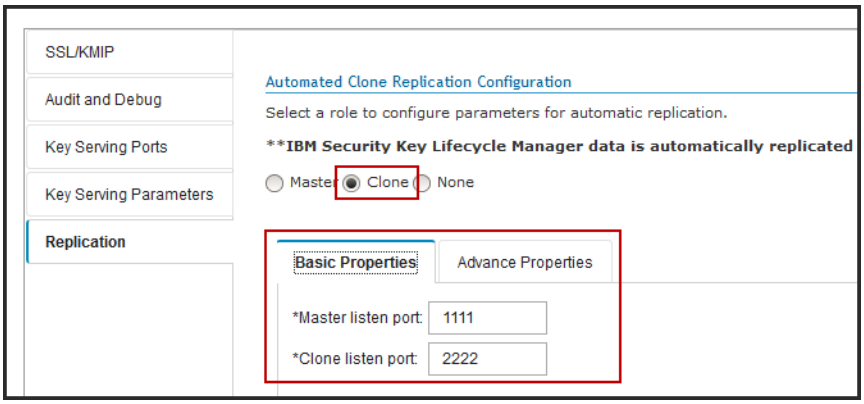


Figure 5-54 Basic Properties (Clone)

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

Configuring the clone in the master key server

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

1. Log in to the master IBM Security Key Lifecycle Manager, and click **Administration** → **Replication** → **Add Clone**, as shown in Figure 5-55.

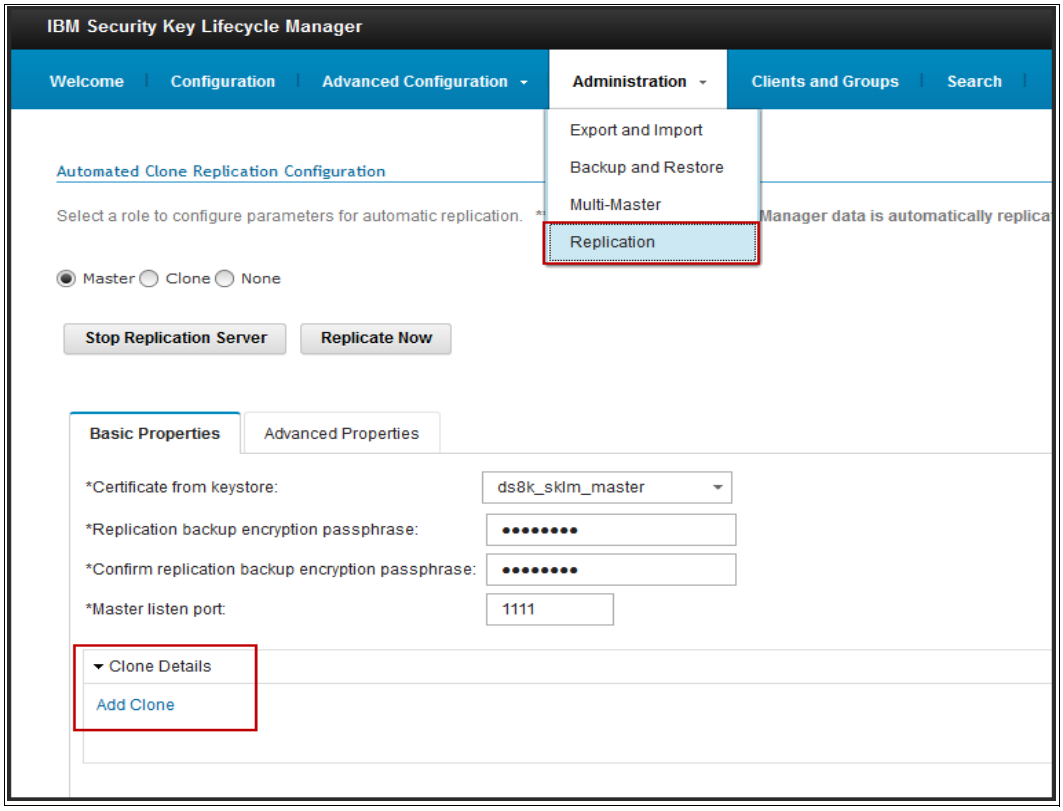


Figure 5-55 Add Clone

2. Add the fully qualified hostname or IP address of the clone, and click **OK**, as shown in Figure 5-56.

Figure 5-56 Insert IP address / hostname

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, as shown in Figure 5-57.

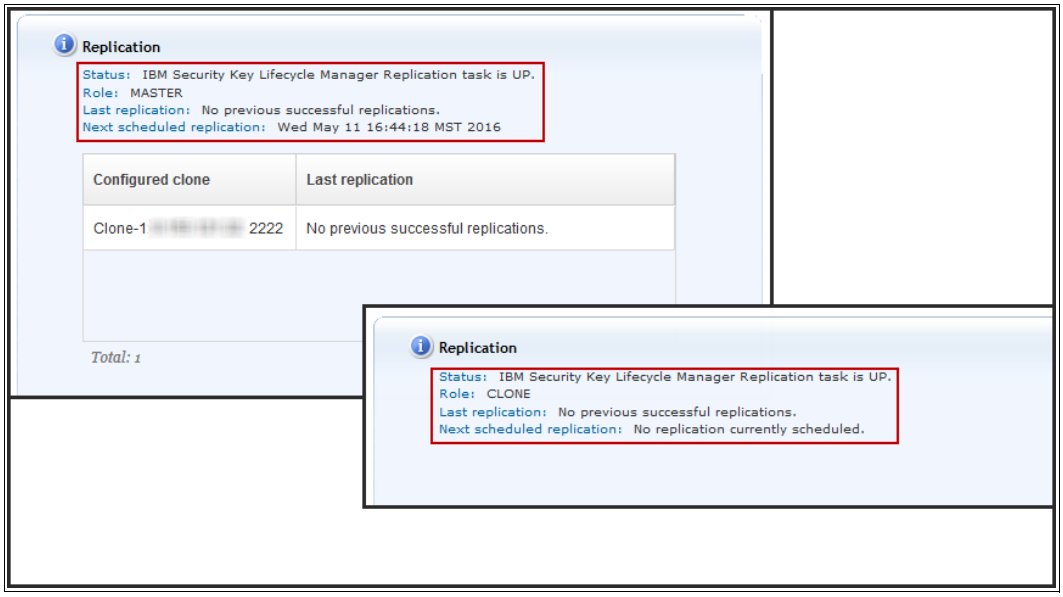


Figure 5-57 Verify the replication status

4. Log in again to the IBM Security Key Lifecycle Manager Replication Master and perform an initial replication. Select **Administration** → **Replication** and select **Replicate Now**, as shown in Figure 5-58.

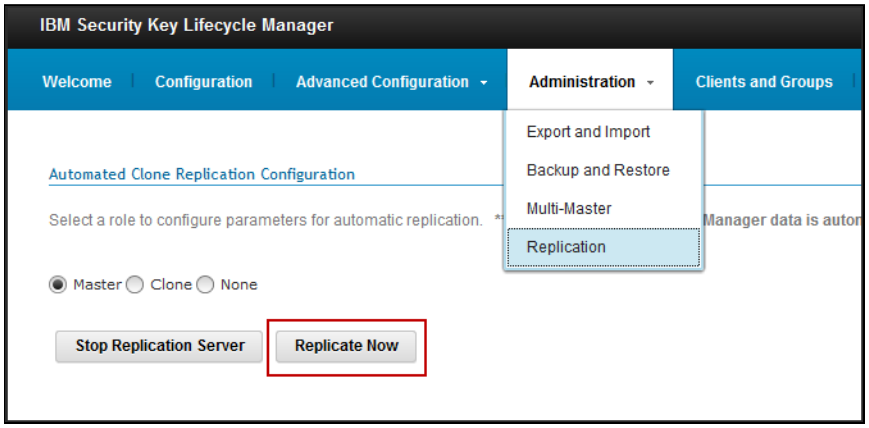


Figure 5-58 Replicate now

5. Confirm the open window by clicking **OK**, as shown in Figure 5-59.

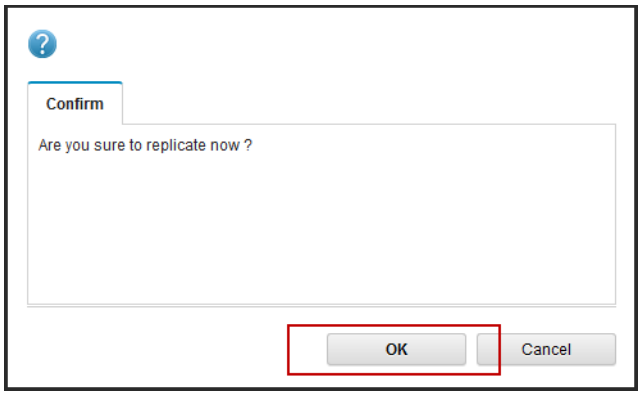


Figure 5-59 Replication confirmation

6. An information message is displayed, as shown in Figure 5-60.

Figure 5-60 Replication successful

7. Return to the Welcome window and verify the replication status. It displays the last replication with the current timestamp, as shown in Figure 5-61.

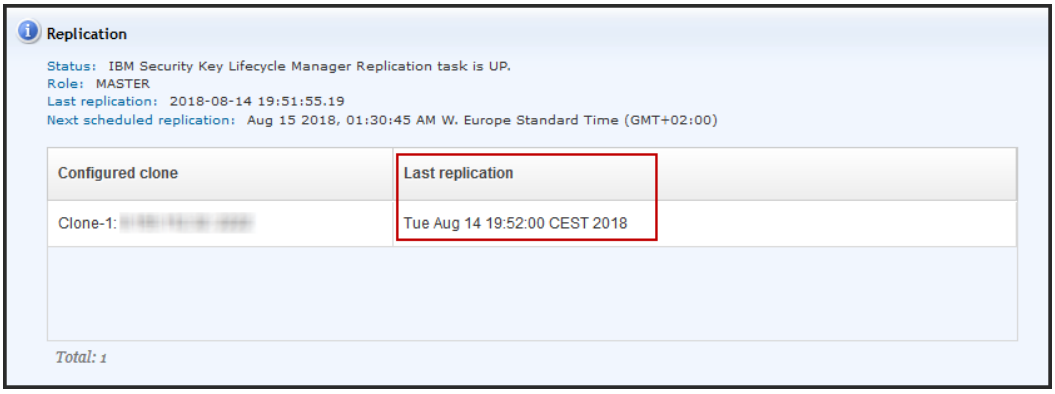


Figure 5-61 Replication status

The IBM Security Key Lifecycle Manager is now set up for replication. All configured servers are ready to serve keys.

Setting up a Multi-Master environment with two IBM Security Key Lifecycle Manager key servers

In a Multi-Master configuration, all IBM Security Key Lifecycle Manager instances in the cluster point to a single data source. This data source is configured for Db2 high availability disaster recovery (HADR) to ensure real-time availability of latest data to all the masters in the cluster.

Db2 HADR configuration is used as single data source for all masters in IBM Security Key Lifecycle Manager Multi-Master cluster. HADR protects against data loss by transmitting data changes from a source database, called primary, to a target database, called the standby. Db2 HADR supports multiple standby databases in your Multi-Master setup.

> **Important:** The Multi-Master setup must be performed *before* any configuration in the IBM Security Key Lifecycle Manager servers, immediately after installation and patching.
>
> A Multi-Master environment is required when you use the KMIP protocol for DAR, TCT encryption, or both.

You must ensure that your computer hostname is configured correctly before you set up IBM Security Key Lifecycle Manager masters for Multi-Master configuration. You can resolve an IP address to a hostname by editing the `etc/hosts` file.

For Db2 HADR configuration, you must update the `/etc/hosts` file in the primary and standby master servers of the cluster to enable host name-to-IP address mapping.

Location of the host file:

- Windows
  `C:\Windows\System32\Drivers\etc\`
- Linux
  `/etc/hosts`

A correct Linux host file is shown in Example 5-21.

Example 5-21 The /etc/hosts file for Multi-Master

```
[root@sklma ]#  cat /etc/hosts
# 127.0.0.1 sklma
0.00.000.01 sklma
0.00.000.02 sklmb
```

Nominate one of the IBM Security Key Lifecycle Manager servers as “Primary Master” and set up the Multi-Master environment on the primary server. The other IBM Security Key Lifecycle Manager servers become “Standby Masters” during the configuration activity.

Creating a SSL/KMIP certificate on the Primary Master

The IBM Security Key Lifecycle Manager servers and its devices require a SSL/KMIP certificate for secure communication between the servers themselves and between the servers and devices, such as the DS8000 HMC. It is always required that you create a SSL/KMIP certificate.

The creation of a SSL/KMIP certificate for Multi-Master configurations corresponds exactly to the procedure described in “Creating an SSL/KMIP server certificate” on page 95.

Do not transfer the created certificate to any standby master. The Multi-Master synchronization process ensures that all servers in the environment are running with the same SSL/KMIP server certificate.

A manual transfer causes the Multi-Master setup to fail. Ensure that the following conditions are met before starting a Multi-Master cluster.

- Only the primary server should have a server certificate.
- Ensure that all other servers that join the Multi-Master cluster are clean without any keys or certificates.
- Ensure that the operating system kernel parameters are correctly set.

For more information, see IBM Security Key Lifecycle Manager 4.0.0.

Adding a standby master to the cluster

An IBM Security Key Lifecycle Manager high-availability solution is implemented by using Multi-Master cluster configuration. The cluster must contain a Primary Master and at least one Standby Master.

The Primary Master was automatically defined in step “Creating a SSL/KMIP certificate on the Primary Master” on page 112.

Complete the following steps to add a Standby Master to the cluster:

1. Log on to the GUI of one IBM Security Key Lifecycle Manager Primary Master Server and select **Administration** → **Multi-Master**, as shown in Figure 5-62.

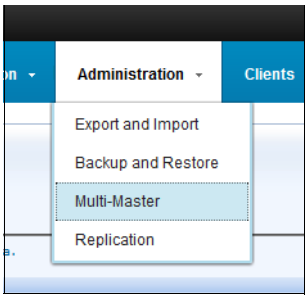


Figure 5-62 Multi-Master menu

2. If the IBM Security Key Lifecycle Manager Multi-Master is not yet configured, click **Multi-Master** for configuration, as shown in Figure 5-63.

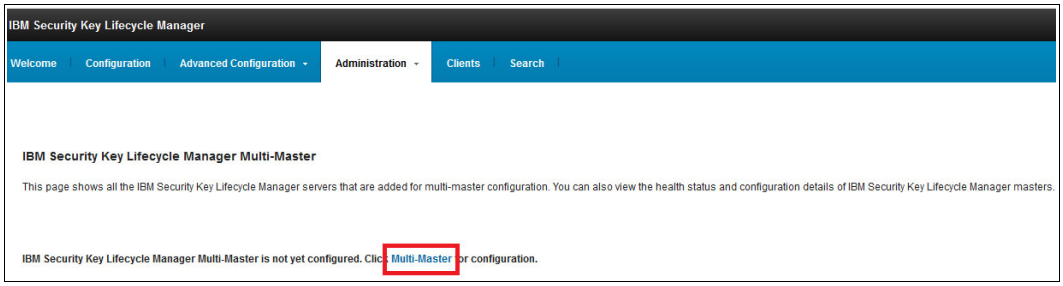


Figure 5-63 Multi-Master configuration

3. A confirmation message appears, as shown in Figure 5-64. The current server automatically becomes a master.

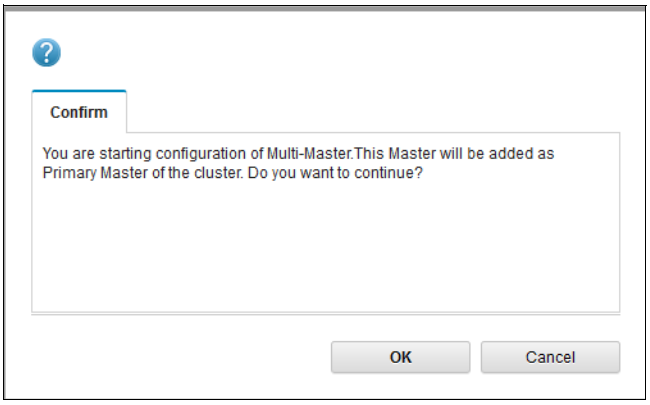


Figure 5-64 Multi-Master confirmation message

4. By default, Agent is not working, so all protocols show as down. This message can be ignored. Click **Add Master** to start cluster creation, as shown in Figure 5-65.

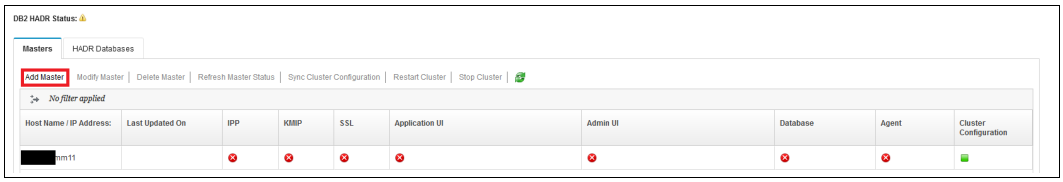


Figure 5-65 Multi-Master: Add Master

5. In the Basic Properties window that is shown in Figure 5-66 on page 115, specify the following information for the standby master that you are adding:
   - **Hostname / IP address**

     Specify the hostname or IP address of the IBM Security Key Lifecycle Manager standby master that is added to the cluster.
   - **IBM Security Key Lifecycle Manager Username**

     Specify the name of the IBM Security Key Lifecycle Manager administrator. The administrator name is displayed by default.
   - **IBM Security Key Lifecycle Manager Password**

     Specify the password for the IBM Security Key Lifecycle Manager server administrator.
   - **WebSphere Application Server Username**

     Specify the WebSphere Application Server login username for the IBM Security Key Lifecycle Manager server administrator profile. The WebSphere Application Server login username is displayed by default.
   - **WebSphere Application Server Password**

     Specify the password for the WebSphere Application Server login username.
   - **UI port**

     Specify the HTTPS port to access IBM Security Key Lifecycle Manager GUI and REST services. The port number is displayed by default.

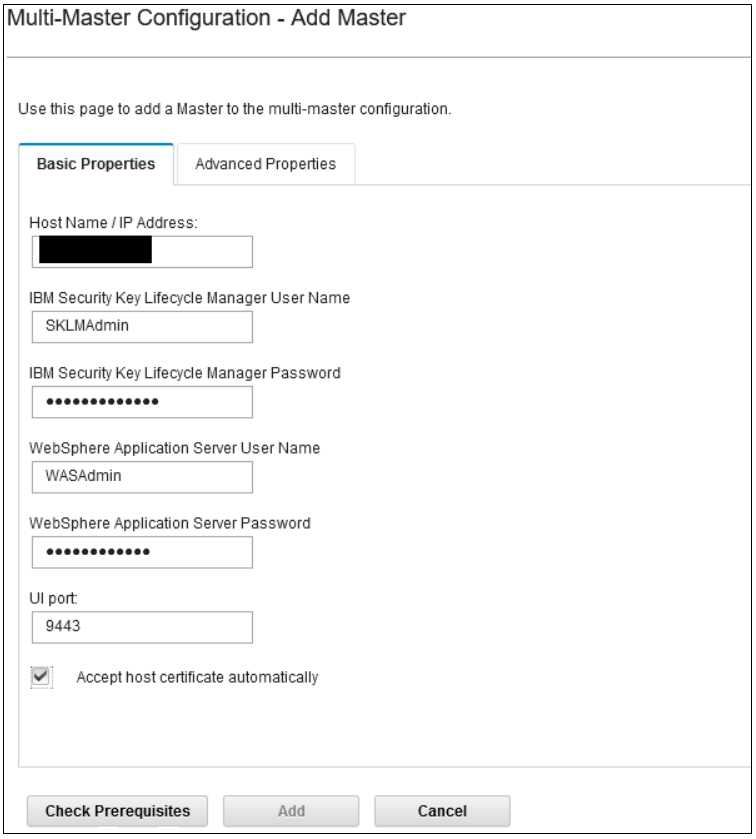


Figure 5-66 Multi-Master - Add Master: Basic Properties

6. Select the **Advanced Properties** tab, as shown in Figure 5-67.

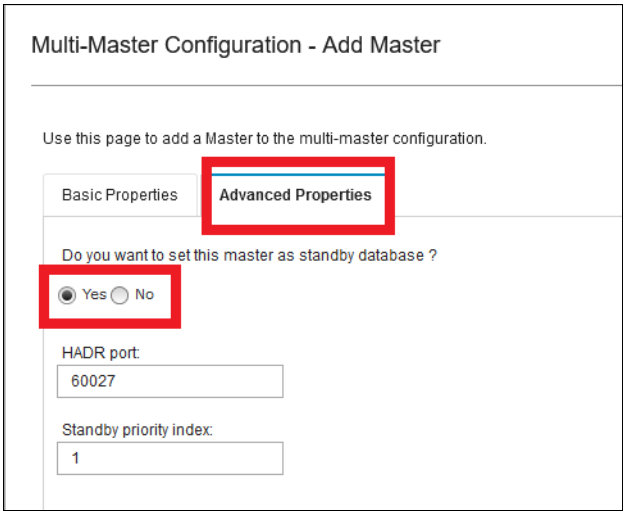


Figure 5-67 Multi-Master - Add Master: Advanced Properties

Specify the following information for the standby master:

– **Do you want to set this master as standby database?**

  Select **Yes** to add the current instance as a standby master to the cluster.

– **HADR port**

  Specify the port number for the standby HADR database to communicate with the primary HADR database. Keep it at the default.

– **Standby priority index**

  Specify the priority index value for the standby database to take over when the primary database is down. You can set the priority index to any value 1 - 3. The standby server with a higher-priority index level (lower number) takes precedence over the lower-priority databases.

7. Click **Check Prerequisite** (as shown in Figure 5-68) to test various prerequisites and communication between the standby master that you are adding and the current primary master. Click **Close** on the information message that is shown when all the prerequisites are met.

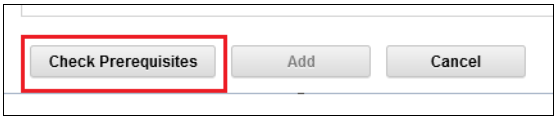


Figure 5-68 Check Prerequisites

8. If the prerequisite checking was successful, click **Add** (as shown in Figure 5-69) to add the Standby Master to the cluster. Confirm the dialog “Are you sure to add this master to the multi-master Cluster?” by clicking **OK**.

Figure 5-69 Add the standby master

9. The HADR Database is now built across the Masters. This process can take up to 10 minutes to complete. A progress window (see Figure 5-70) is shown.

Figure 5-70 HADR progress

10. A confirmation window opens. After clicking **OK**, both IBM Security Key Lifecycle Manager instances are restarted in the background.

11. Log in to all configured IBM Security Key Lifecycle Manager servers and verify the Multi-Master availability. The Welcome window should resemble the example that is shown in Figure 5-71 on page 117.

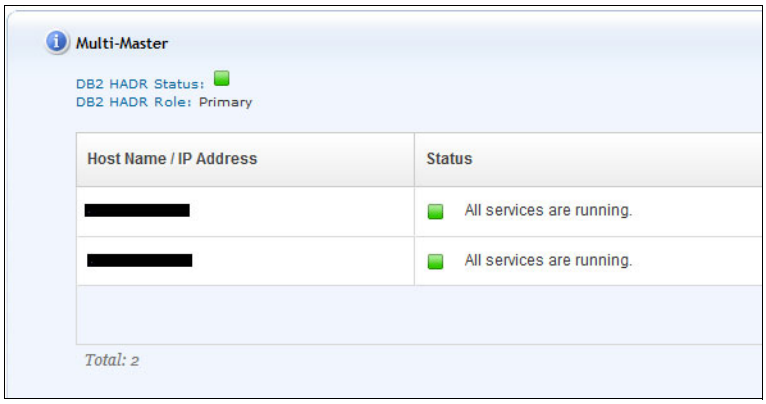


Figure 5-71 Multi-Master status on the welcome window

12. If one of the hosts shows a failed state, which is indicated by a red cross, return to the Multi-Master configuration window, select the host that shows the red cross, and refresh its status, as shown in Figure 5-72.

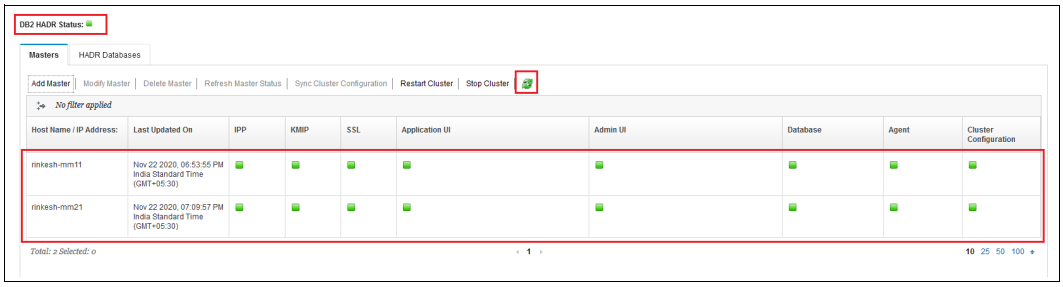


Figure 5-72 Multi-Master detailed status

The IBM Security Key Lifecycle Manager is now set up as a Multi-Master cluster. All configured servers in the HADR database are now ready to serve keys.

5.4.2 Gemalto SafeNet KeySecure configuration

Gemalto SafeNet KeySecure (KS) 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 KS is available as a hardware and virtual software appliance.

In our scenarios, we used Version 8.3.2 RevA. It supports KMIP Version 1.1 and the IBM Security Key Lifecycle Manager does (used with DS8000 Release 8.1 and later), LDAP and Active Directory authentication, and multiple network management protocols.

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

Gemalto SafeNet KS 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 KS, see Cloud Protection and Licensing Solutions.

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

Preparation

When using Gemalto SafeNet KS 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 best practice because it does not meet KMIP standards.

- Client Certificate Authentication used for SSL session only

  This policy applies to DS8000 disk storage systems that are included with Release 8.1 and later 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 (UID)

  This policy applies to DS8000 disk storage systems that are shipped from manufacturing with Release 8.1 and later. A 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 UID. It is the most secure way.

Not using Client Certificate Authentication (policy 1) is not a best 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 KS Configuration, make sure to satisfy the following prerequisites:

- Make sure that you configured two independent key servers in a cluster.
- Configure the recovery key (RK) (see “Creating the recovery key” on page 161).
- Update the DS8000 certificate from Gen 1 to Gen 2 or Gen 3, as described in “Migrating certificates” on page 199.
- Download the root certificate for DS8000 to the client computer from the DS8000 IBM Knowledge Center.
- Policy 3 only: Export the Gen 2 certificate 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 “DS8000 Encryption Communication Certificate (Gen 2) export and usage” on page 119.

Note: DS8000 Release 8.1 and later features a Gen 2 certificate from manufacturing by default. DS8900F R9 features a Gen 2+ certificate that is active and a Gen 3 certificate that is not active.

DS8000 Encryption Communication Certificate (Gen 2) export and usage

1. To export the Gen 2 certificate, follow the procedure in “DS8000 TCT encryption certificate (Gen 2) import (IBM Security Key Lifecycle Manager)” on page 182 and return here.
2. 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 5-22.

   Example 5-22 Extract the UID

```
[root@sklm-rehl64 tmp]# openssl x509 -in smokerlh_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:

   a. Run CertManager, as shown in Figure 5-73.

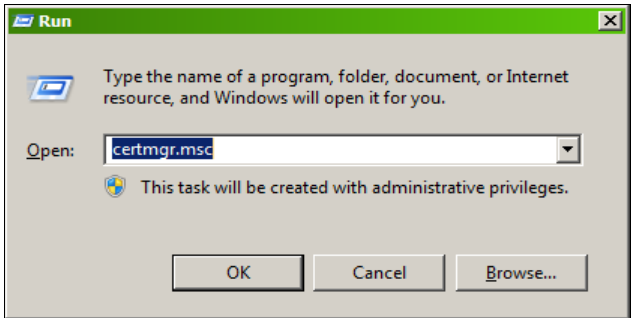


Figure 5-73 Run CertMgr

b. Click **Personal** → **Certificates** and then, click **All Tasks** → **Import**, as shown in Figure 5-74.

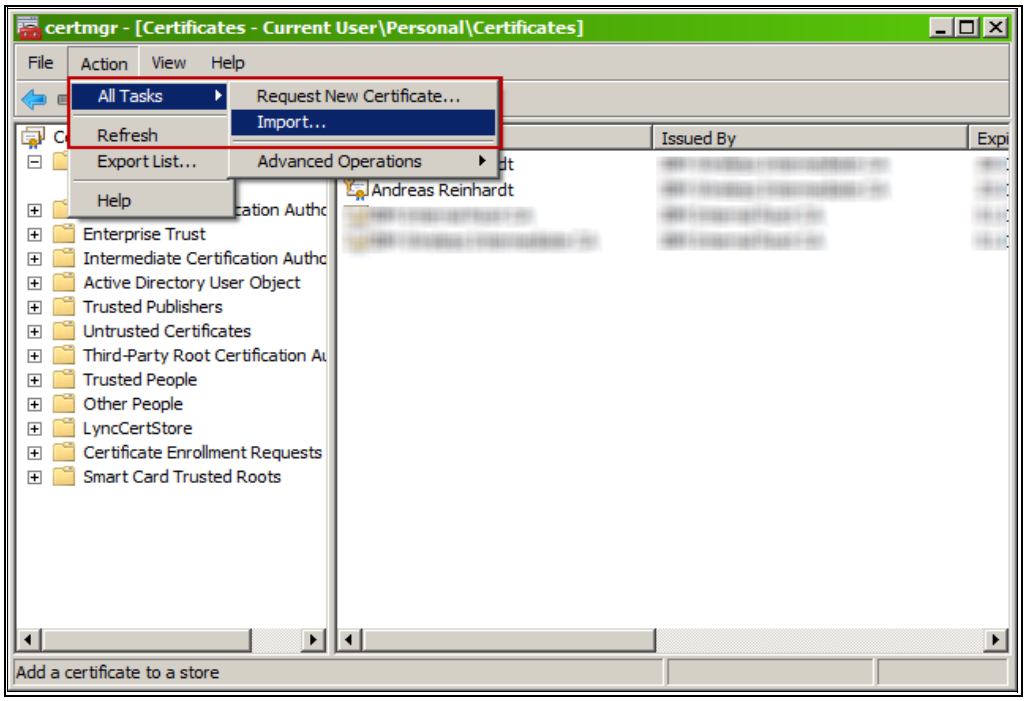


Figure 5-74 Import menu

c. Follow the wizard to import the certificate. Be sure to select **All Files (*.*)** to see the certificate in `.pem` format, as shown in Figure 5-75.

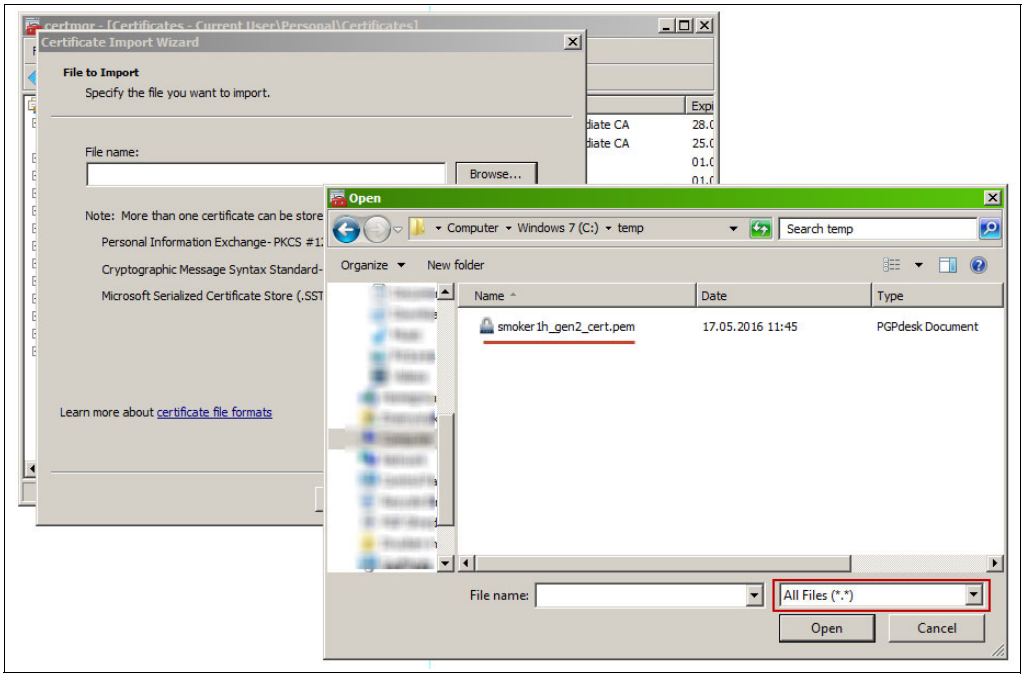


Figure 5-75 Select “All Files (.*)”*

The certificate is now imported, as shown in Figure 5-76.

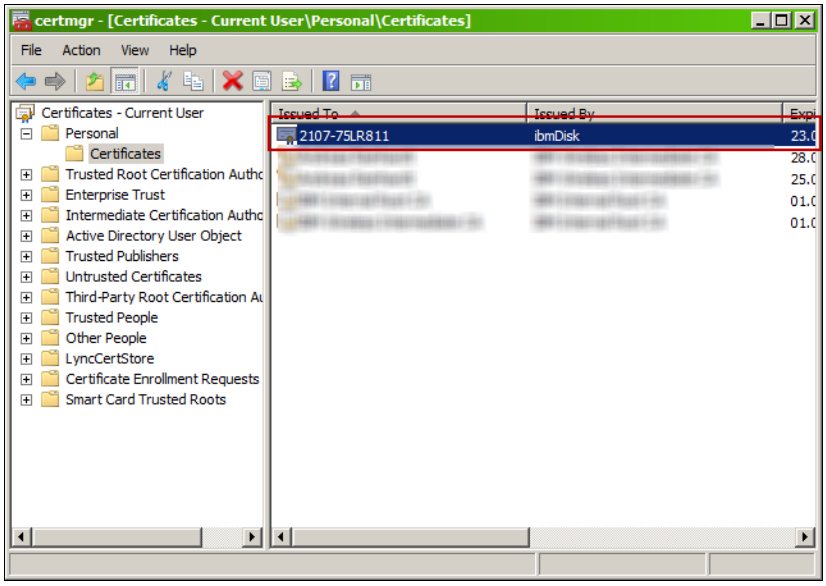


Figure 5-76 The imported certificate

d. Open the certificate, click the **Details** tab, and select **Subject**. Figure 5-77 shows the UID.

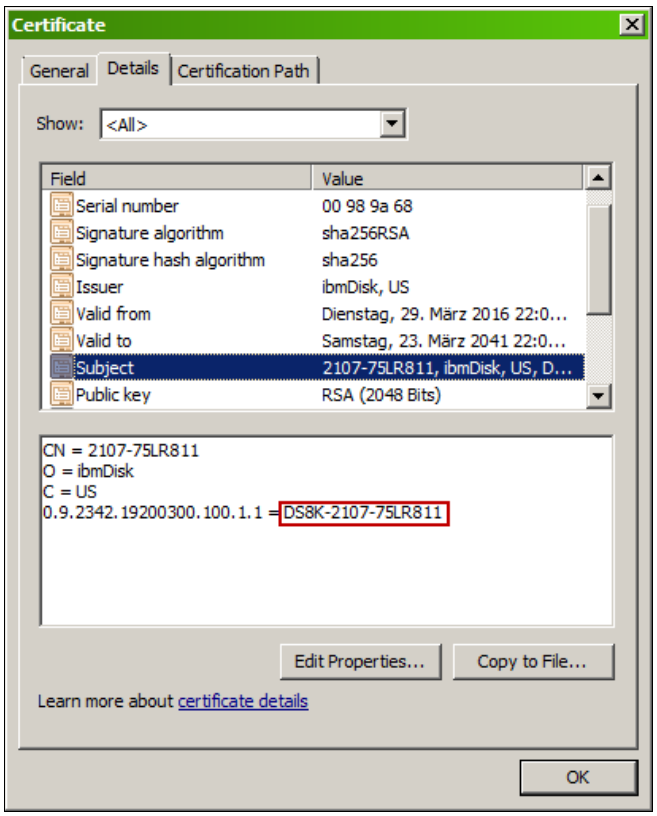


Figure 5-77 Show the UID

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

You can delete the Gen 2 certificate from the Windows keystore and Windows and UNIX HDDs 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 or later that is delivered from manufacturing has a Gen 2 certificate.

To update your Gen 1 certificate, see 5.10, “Migrating certificates” on page 199.

If you do not have access to the DS8000 IBM Knowledge Center, use the root certificates from the Example 5-23 (Gen 1) and Example 5-24 (Gen 2).

Make sure to copy everything, including `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.

Example 5-23 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 5-24 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 Gen 1 and Gen 2 root CA encryption certificates.

Configuration

Note: Illustrations in this section are shown courtesy of Thales DIS CPL US, Inc.

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 KS 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 KS 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 KS 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 KS 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. Select **Security** → **SSL Certificates** and create a certificate request, as shown in Figure 5-78.

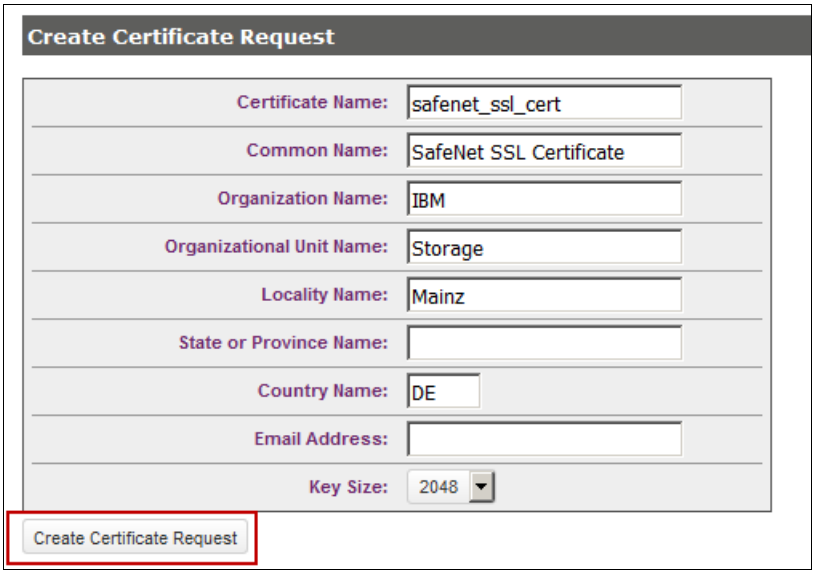


Figure 5-78 Create a certificate

3. After the SSL certificate is created, select it and click **Properties**, as shown in Figure 5-79.

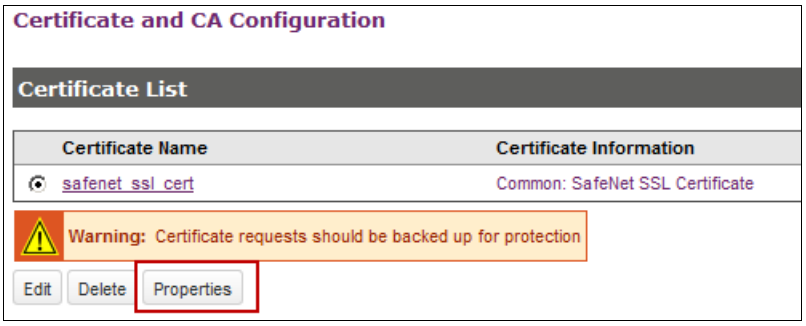


Figure 5-79 SSL certificate properties

4. Within the properties, select **Create Self-Signed Certificate**, as shown in Figure 5-80.

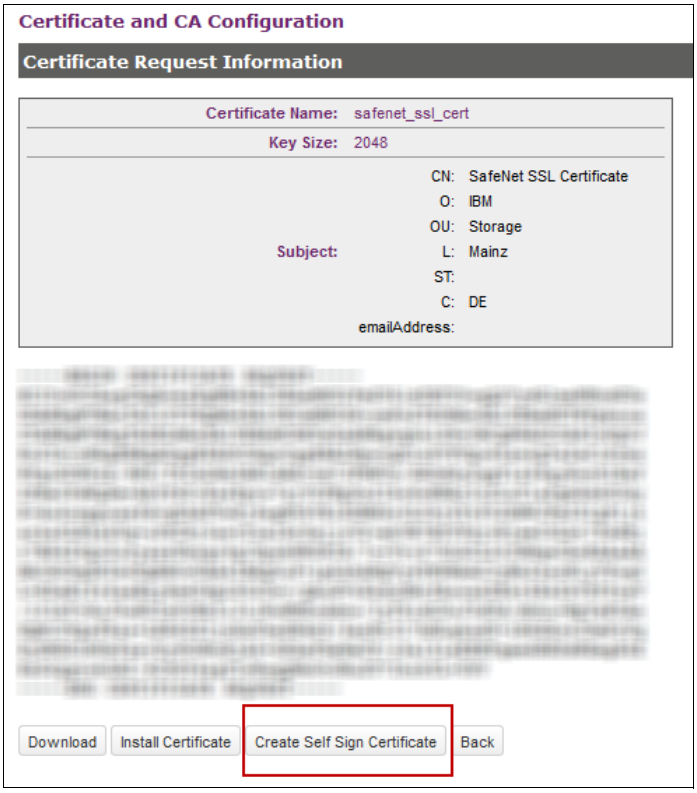


Figure 5-80 Self-sign the certificate

5. You can modify the certificate duration in days, as shown in Figure 5-81. Although you can use the system to specify a maximum of 7300 days (20 years), it is advised as a cryptographic best practice to use smaller durations, such as 365 or 730 days (1 or 2 years).

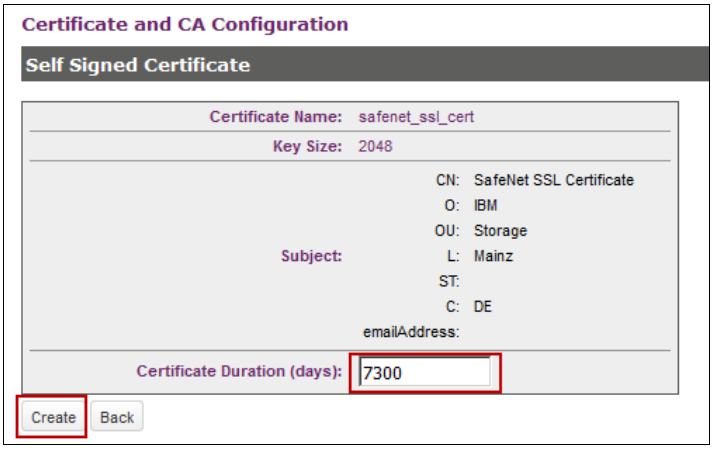


Figure 5-81 Maximum certificate duration

The SSL Certificate is now active, as shown in Figure 5-82.

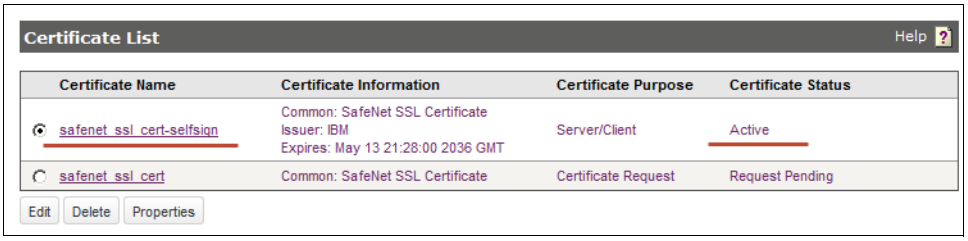


Figure 5-82 SSL Cert Ready

6. Select the self-signed SSL certificate again and select **Properties**. Then, click **Download**, as shown in Figure 5-83.

Figure 5-83 Download the Concurrent Code Load certificate

7. Save it to your local HDD and do not rename it, as shown in Figure 5-84.

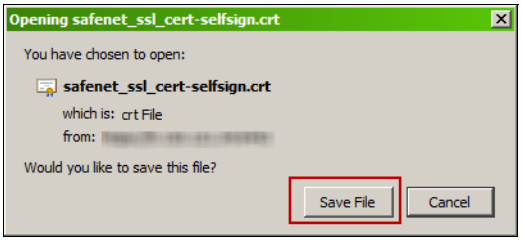


Figure 5-84 Save the SSL certificate to the hard disk drive

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

Installing the DS8000 Gen 2 root certificate from the DS8000 IBM Knowledge Center

This section shows how to install the DS8000 root certificate for the Gemalto SafeNet KS 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 KS key server, independently from which one you choose.

Complete the following steps:

1. Still logged in to the Gemalto SafeNet KS GUI, click **Security** → **Known CAs**, as shown in Figure 5-85, paste the certificate text, and click **Install**. The name should clearly identify the DS8000 Gen 2 root certificate.

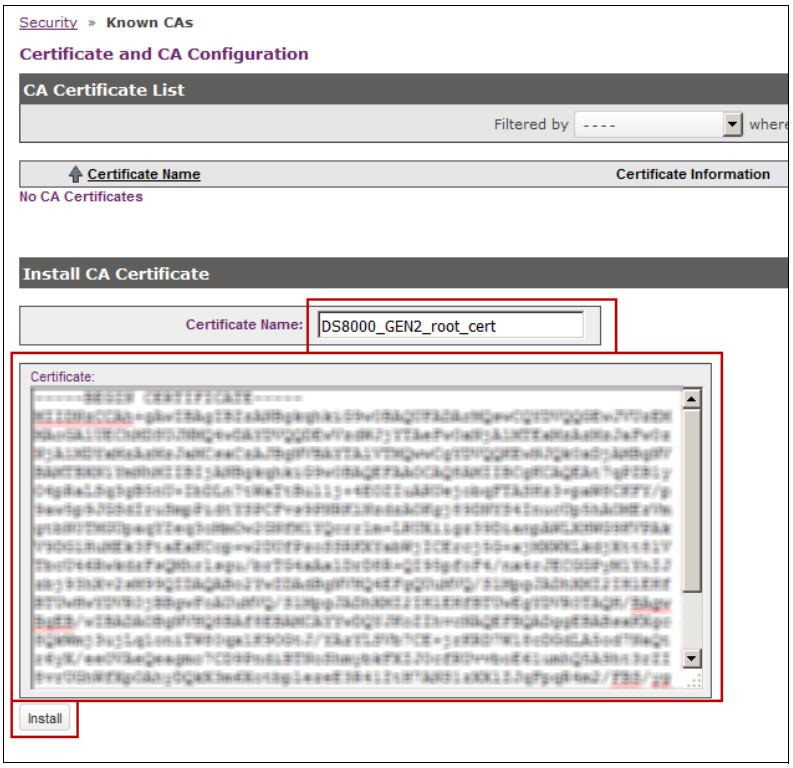


Figure 5-85 Install the DS8000 Gen 2 root certificate

2. The root certificate is now installed and active. As shown in Figure 5-86, the certificates must be added to a trusted CA list to be recognized by the KMIP server.

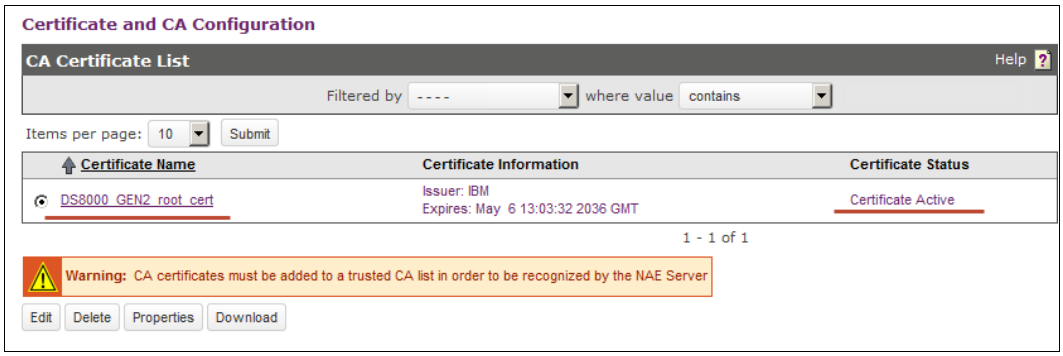


Figure 5-86 Root certificate is now installed

Using a multi-layer certificate chain of trust

In some environments, there might be a requirement to use certificates that are signed by multiple certificate authorities. This situation is known as a multi-layer certificate chain of trust.

The certificate chain of trust includes a leaf certificate, a set of intermediate CA certificates, and a root CA certificate.

For successful authentication between the DS8000 and the Gemalto SafeNet KS key server, the entire chain of trust must be presented to the DS8000. To ensure that condition, proceed as follows:

1. Configure the Gemalto SafeNet KS key server with the leaf certificate and the set of intermediate CA certificates.

   Refer to Figure 5-85 on page 127. In the entry box that is displayed, paste the certificate chain in PEM format, with the leaf certificate first and then the successive intermediate CA certificates in descending order.

2. When you configure the key manager on the DS8000, use the root CA certificate as the key server certificate.

Tip: You can test the chain of trust by using the following command:

```
openssl s_client -connect key_server_ip_address:key_server_port_number -showcerts
```

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 CA to the profile list.

Complete the following list:

1. Still logged in to the Gemalto SafeNet KS GUI, click **Security** → **Trusted CA Lists**, as shown in Figure 5-87, and click **Add**.

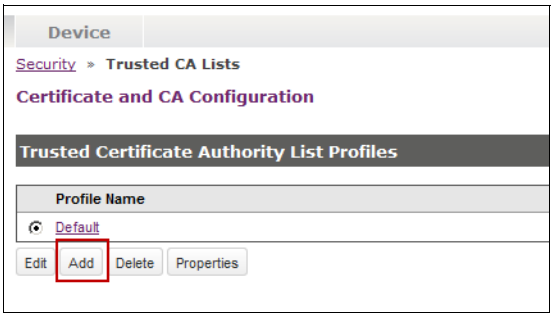


Figure 5-87 Add a profile

The name should clearly identify the DS8000 profile, as shown in Figure 5-88 on page 129.

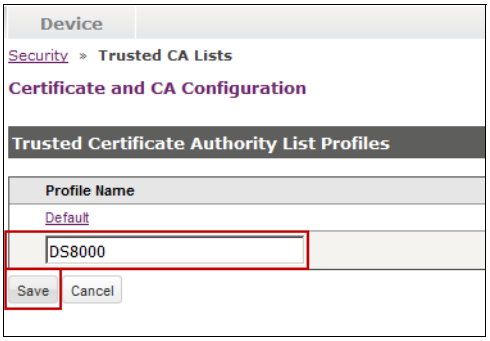


Figure 5-88 Name the profile

2. The new profile is now in the list of profiles, as shown in Figure 5-89. Select it and open the properties.

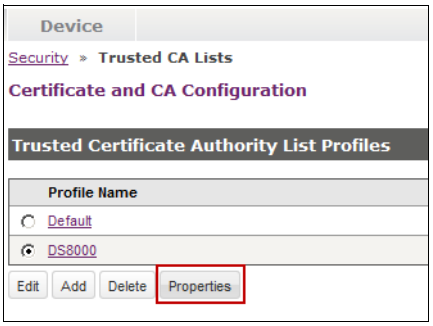


Figure 5-89 Profile properties

3. In the profile properties (see Figure 5-90), click **Edit**.

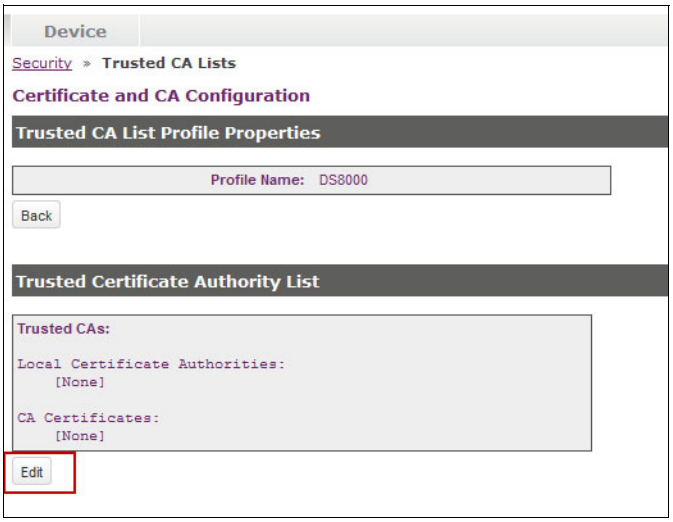


Figure 5-90 Edit the profile

4. Add the previously created DS8000 root CA to the list of trusted CAs and save it, as shown in Figure 5-91.

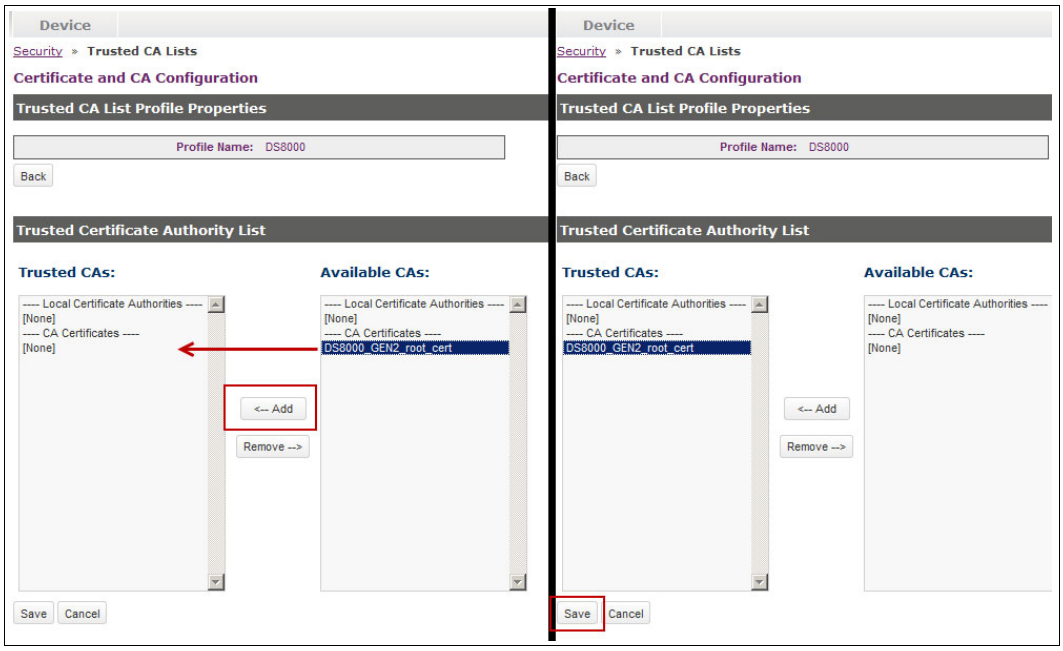


Figure 5-91 Add the DS8000 root CA to the list of trusted CAs

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 KS GUI, click **Device** → **Key Server** and click **Add** to create a protocol, as shown in Figure 5-92. One NAE-XML protocol is preconfigured, which can be ignored.

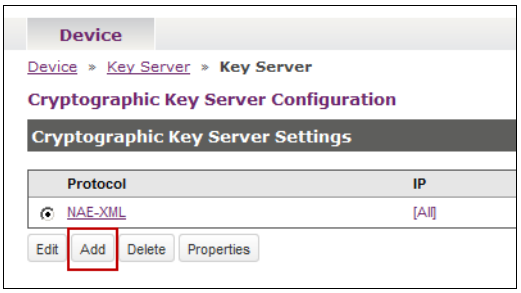


Figure 5-92 Add KMIP protocol

2. Figure 5-93 on page 131 shows the settings for the KMIP protocol that must be selected:
   - Protocol: KMIP.
   - IP: All.
   - Port: 5696. (This port 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 123.

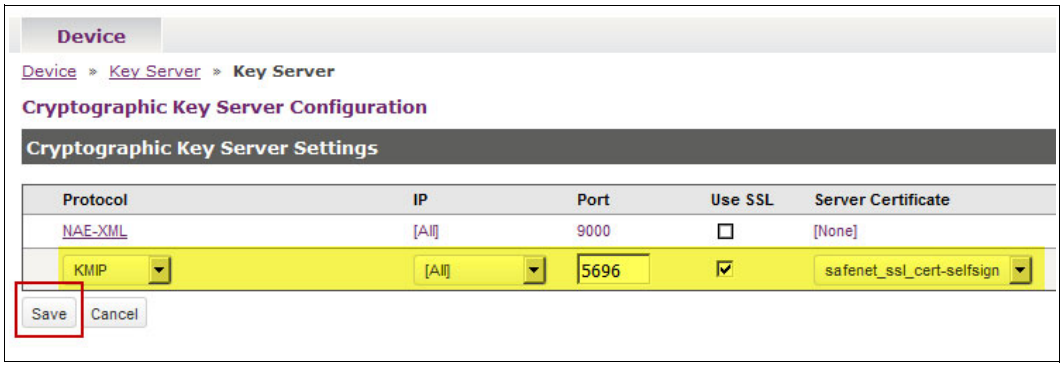


Figure 5-93 Define KMIP protocol settings

3. The KMIP protocol appears in the list, as shown in Figure 5-94. Select it and click **Properties**.

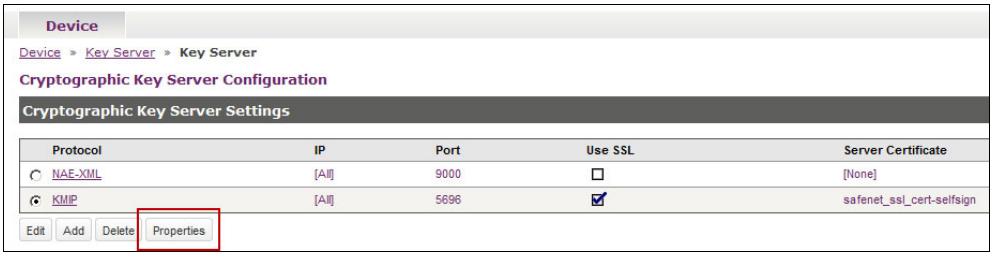


Figure 5-94 KMIP protocol created

4. In the KMIP Protocol properties, click **Edit** in the Authentication Settings area, as shown in Figure 5-95.

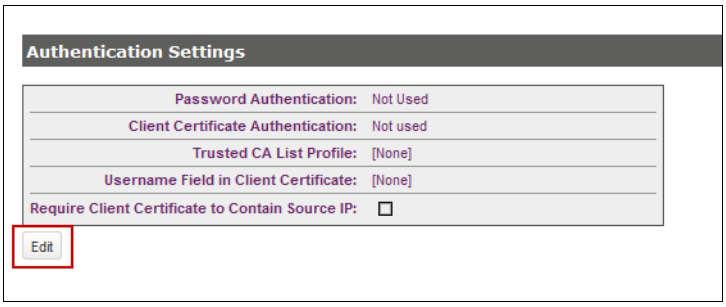


Figure 5-95 KMIP Authentication Settings

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 a 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 a 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 5-96. Click **Save** when you are ready.

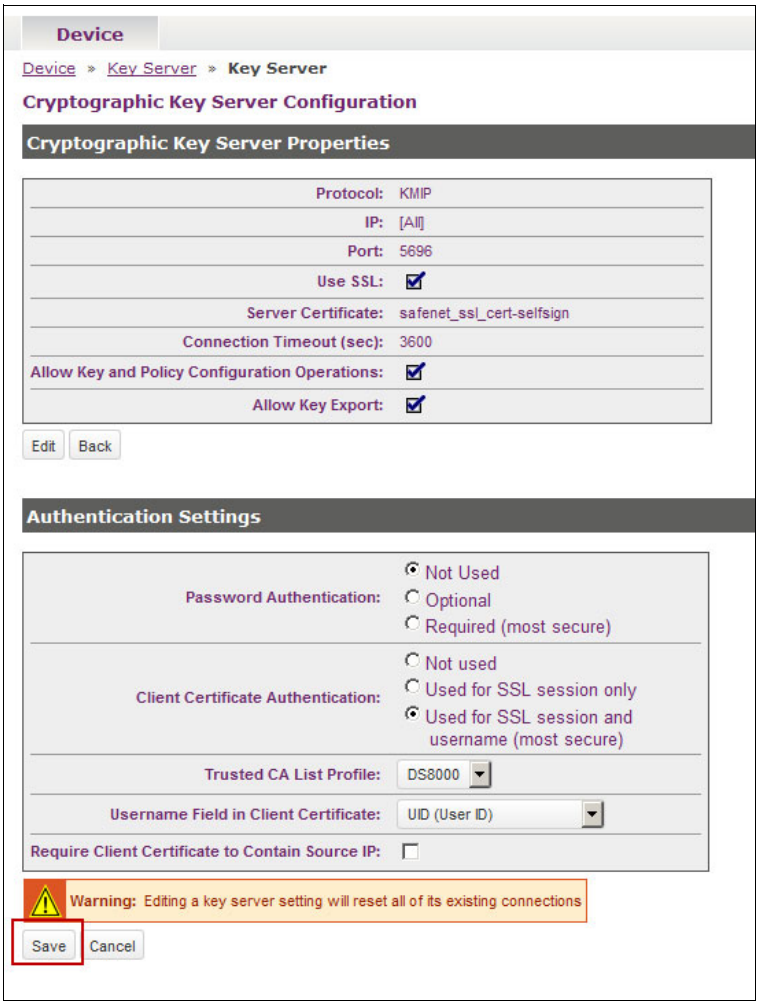


Figure 5-96 Configured KMIP profile

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 or later in your environment that is encrypted.

Complete the following steps:

1. Still logged in to the Gemalto SafeNet KS GUI, click **Security** → **Local Authentication** → **Local Users & Groups** and click **Add** to add a user, as shown in Figure 5-97.

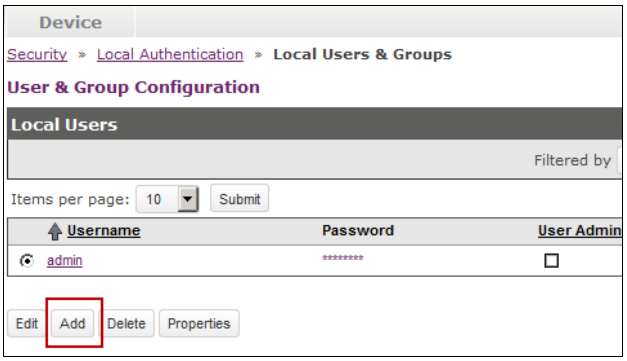


Figure 5-97 Add a DS8000 user

2. The UID equals the UID that you extracted from the DS8000 Gen 2 certificate in "Extracting the **UID** field from this certificate." on page 119. In Figure 5-98, the UID is DS8K-2107-75LR811. Add as many different UIDs as you require in your environment and click **Save**. Complete the following fields:
   - Password: The password 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.

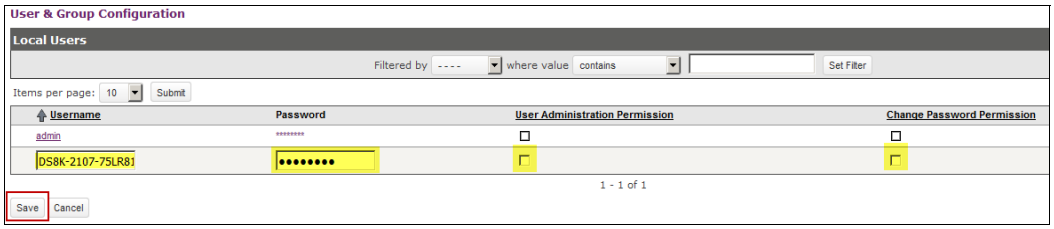


Figure 5-98 User details

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

5.4.3 Configuring Thales Vormetric Data Security Manager

The Thales Vormetric Data Security Manager (DSM) is another third-party external key manager that is supported for DS8000 DAR encryption and TCT encryption to cloud storage.

For more information, see Vormetric Data Security Manager.

Like IBM Security Key Lifecycle Manager, Thales Vormetric DSM provides a web browser-based GUI.

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

Prerequisites

Before starting the Thales Vormetric DSM Configuration, make sure to satisfy the following prerequisites:

- Make sure that you configure two or more independent key servers in a cluster, and the system clocks on the servers are synchronized.
- Configure the RK (see “Creating the recovery key” on page 161).
- Client Certificate Authentication for SSL Session and UID only: Export the Gen 2 or Gen 3 certificate to the client computer.

Note: DS8000 Release 8.1 and later features a Gen 2 certificate from manufacturing by default. DS8900F R9 features a Gen 2+ certificate that is active and a Gen 3 certificate that is not active.

Preparation steps

Note: Illustrations in this section are shown courtesy of Thales DIS CPL US, Inc.

Start the Thales Vormetric DSM Management console from a web browser and provide your login and password credentials. In our example in Figure 5-99, we used the `ds8000` UID, which is defined as a DSM administrator.

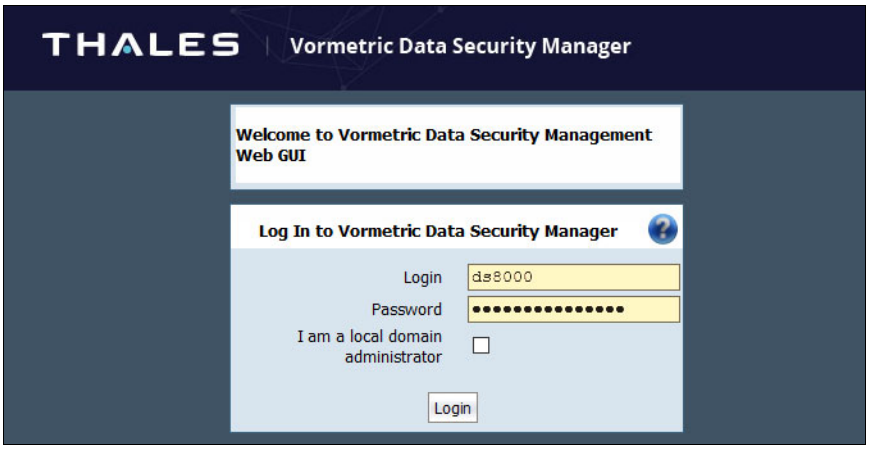


Figure 5-99 Log on to the Thales Vormetric Data Security Manager Management GUI

You must have a valid license to configure hosts and register agents with a DSM. To register and configure a DS8000 system, install a license file that contains KMIP agents. Licenses are provided by the Thales Vormetric DSM customer support and are uploaded to the DSM (see your DSM documentation on how to obtain a license file).

Complete the following steps:

1. Upload a license file:

   a. Log on to a DSM HA node as a System/All administrator (other administrator types can view the license in the Management Console, but they cannot upload a license file. In our example, we defined an administrator that is named ds8000 with the All type.

   b. Get a license file from Customer Support.

   c. Select **System** → **License** in the menu bar. The License window opens, as shown in Figure 5-100.

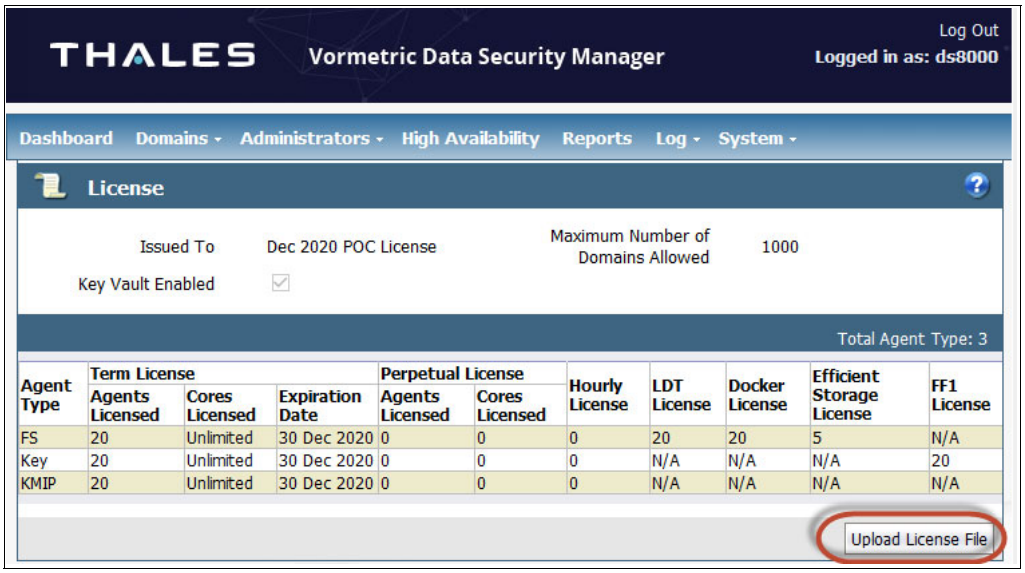


Figure 5-100 Thales Vormetric Data Security Manager: Install system licenses

   d. Click **Upload License File**. The Upload License File window opens.

      If you are in a domain, **Upload License File** is disabled. Click **Domain** → **Exit Domain** instead.

   e. In the **License File** dialog box, click **Browse** to find and select the license file, as shown in Figure 5-101.

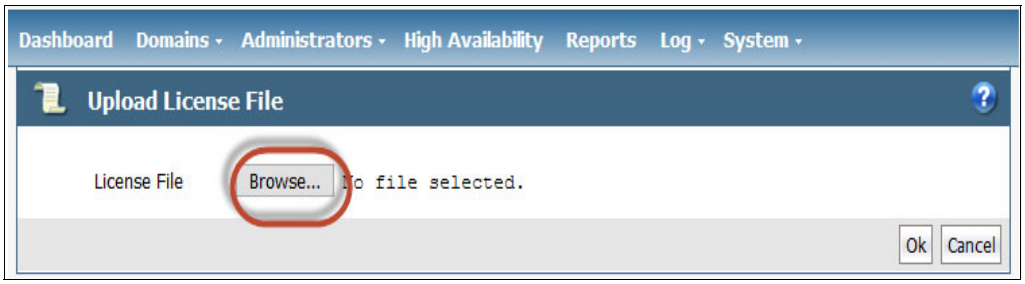


Figure 5-101 Browse and select the license file

   f. Click **OK**.

2. Create a domain and enable KMIP:

   a. Select **Domain** → **Manage Domains** from the menu bar. The Manage Domains window opens, as shown in Figure 5-102.

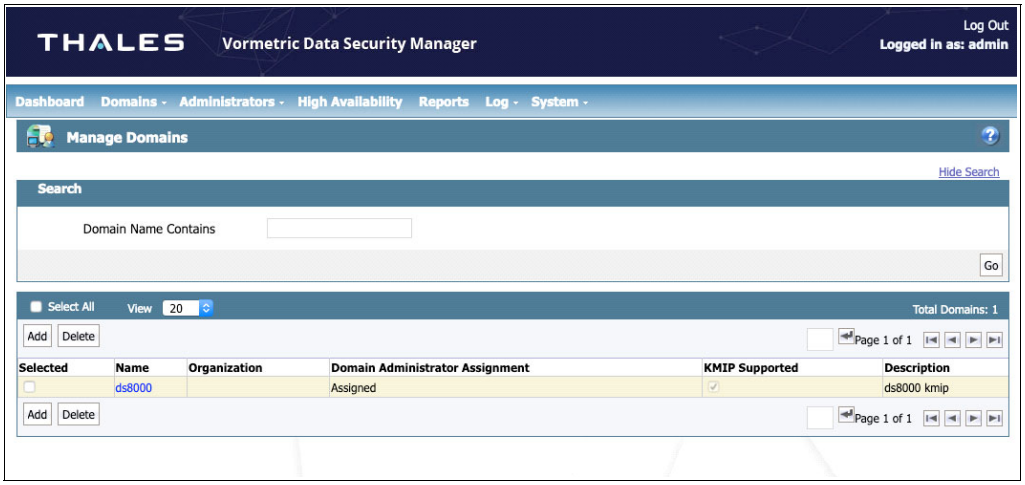


Figure 5-102 Manage and create domains

   b. Click **Add** to open the Add Domains window. In the **General** tab, complete the details for the domain. Select **Enable KMIP**. Click Apply to create the domain, as shown in Figure 5-103.

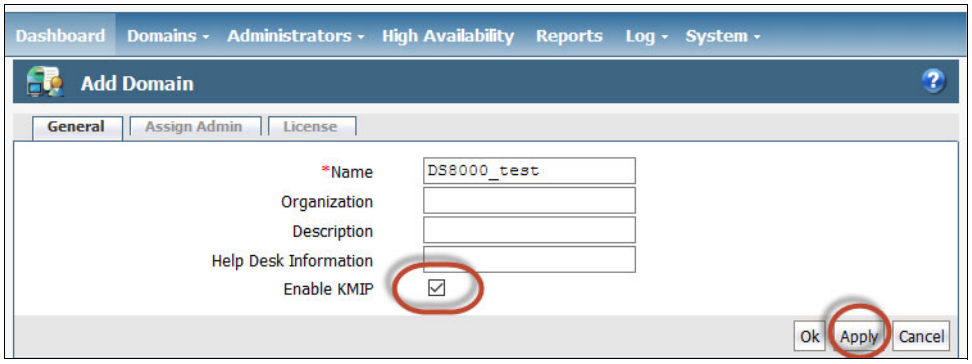


Figure 5-103 Add Domain window

   c. In the **Assign Admin** tab, select the administrator account that you created, and then click **Apply**, as shown in Figure 5-104.

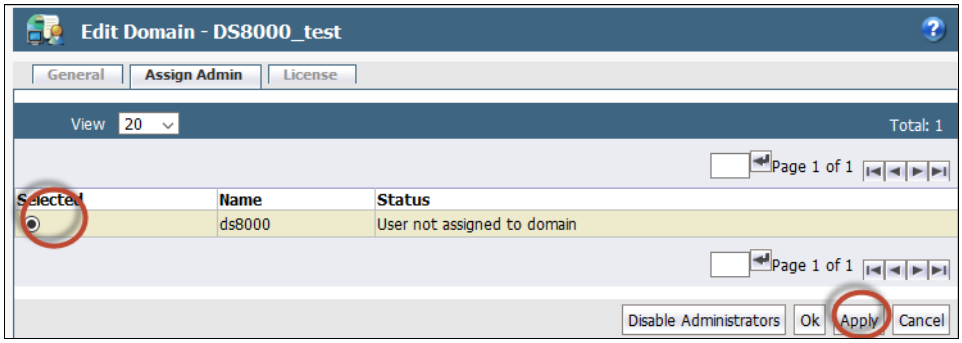


Figure 5-104 Assign an administrator to the new domain

d. In the **License** tab, complete the KMIP Agent information, and then click **OK** to return to the Manage Domain window.

Configuration steps

To configure a DS8000 storage system in Thales Vormetric DSM, log on to the DSM console as the Security administrator with Host role permissions, a Domain and Security administrator, or All administrator.

Complete the following steps:

1. Select **Domain** → **Switch Domains** from the menu bar to switch to the domain where you will add the host, as shown in Figure 5-105.

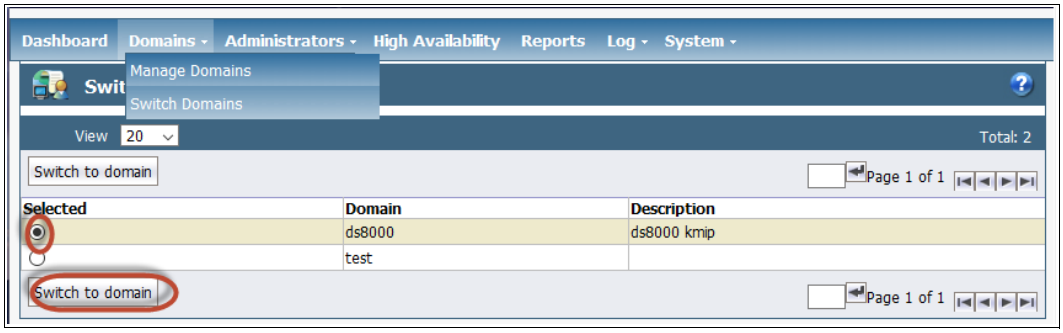


Figure 5-105 Switch Domain

Make your selection, and then click **Switch to domain**.

2. Select **Hosts** → **Hosts** in the menu bar. The Hosts window opens, as shown in Figure 5-106.

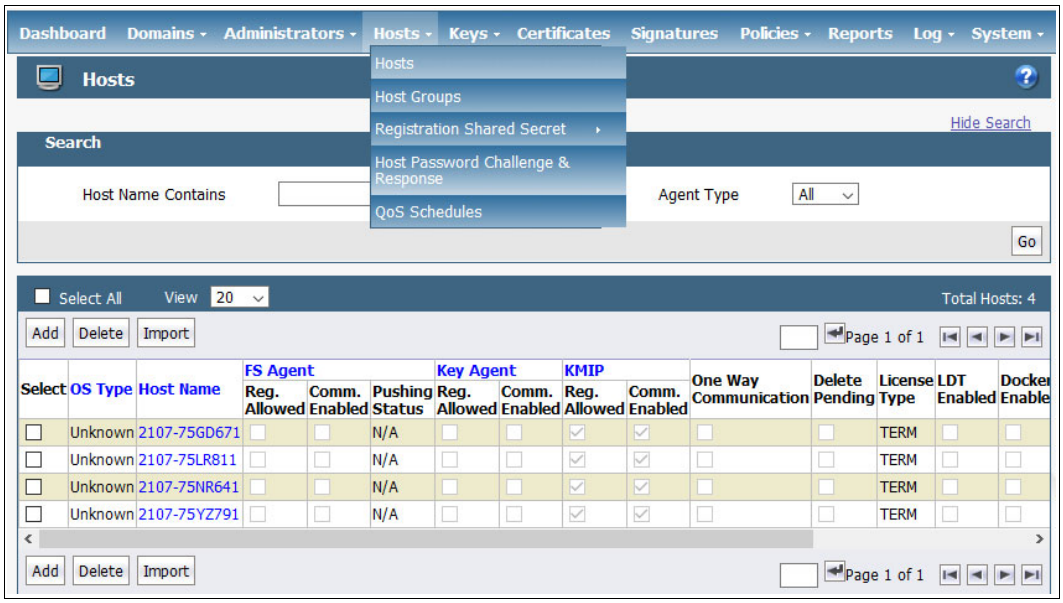


Figure 5-106 Hosts window

3. Click **Add** to add a host. The Add Host window opens, as shown in Figure 5-107.

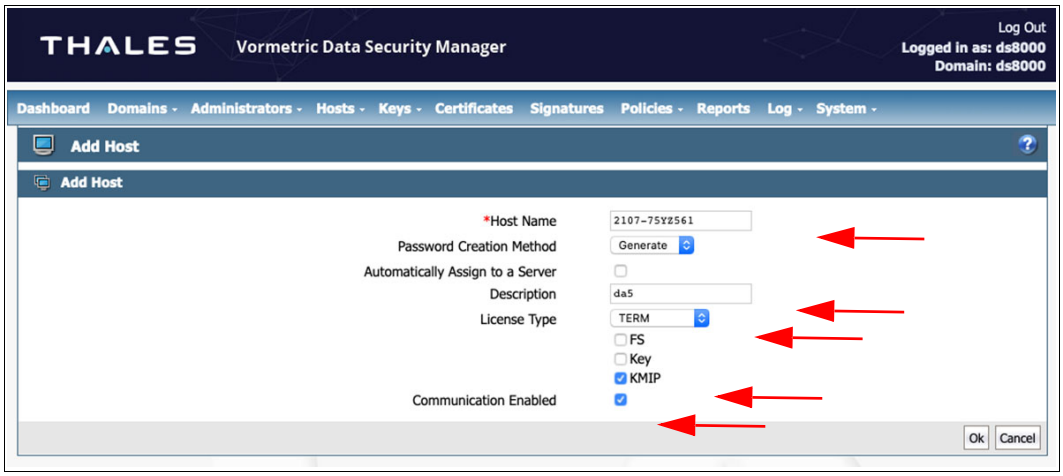


Figure 5-107 Add Host

In the Add Host window, complete the details for the host as follows:

– **Hostname**

  The hostname must match the CN field of the client certificate for the storage system. You can get the CN field by using the DS8000 DS Command-Line Interface (DS CLI) **showkeygrp** command, as shown in Example 5-25.

  Example 5-25 Retrieve the CN from the DS8000 certificate

```
dscli> showkeygrp  -certificate  1
Date/Time: December 16, 2020 6:41:16 PM MST IBM DS CLI Version: 7.9.20.181
DS: IBM.2107-75NR641
ID                    1
certificate           GEN2
version               3
serial number         00:fd:1b:32:f1:35:b3:7c:74
issuer                O=ibmDisk,C=US
not valid before      06/26/2019 20:45:19 MST
not valid after       08/11/2032 17:36:55 MST
subject               CN=2107-75NR641,O=ibmDisk,C=US,uid=DS8K-2107-75NR641
WWNN                  OTHERNAME=50:05:07:63:06:FF:C5:DD
size                  2048
..
..
```

– **Password Creation Method**: Select **Generate**.

– **License Type**: Select **TERM** and **KMIP**.

– Under **Agent Information**, make sure to select **Communication Enabled**.

4. Click **OK**. You are returned to the Hosts window.

5. You must now import the DS8000 public certificate. From the Host window, click the hostname for the storage system, which opens the Edit Host window, which is shown in Figure 5-108.

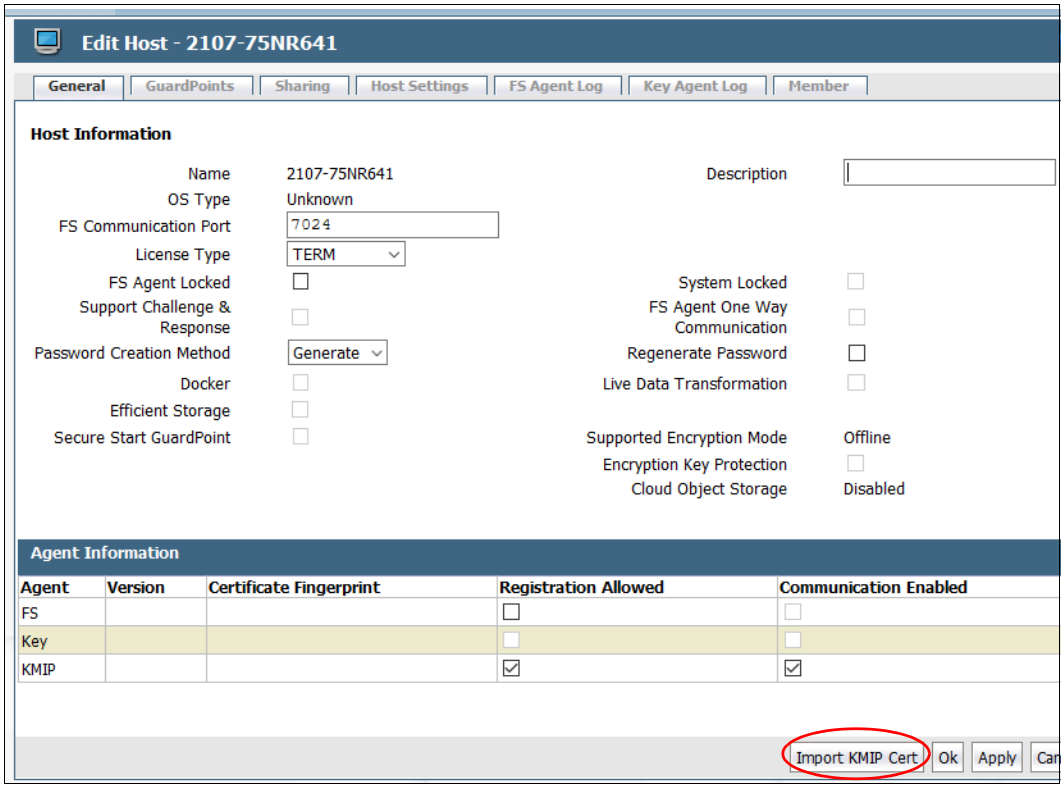


Figure 5-108 Edit Host window

6. Click **Import KMIP Cert** to import the DS8000 public certificate. The Import KMIP Client Certificate dialog box opens, as shown in Figure 5-109.

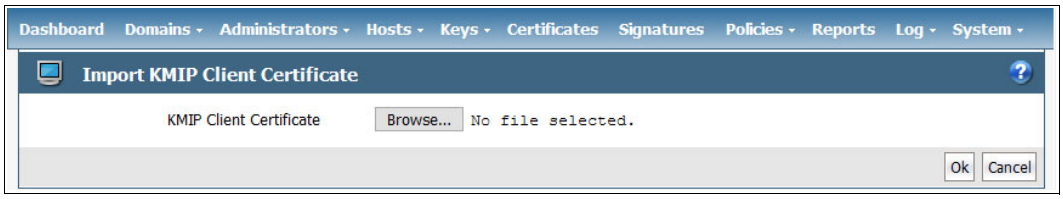


Figure 5-109 Browse for the DS8000 KMIP communication certificate

Click **Browse** to retrieve the file containing the DS8000 public communication certificate. You normally download the certificate as describe under “Prerequisites” on page 134. You also can retrieve it by running the **`showkeygrp -certificate 1`** command, as shown in Example 5-25 on page 138.

7. Click **OK**. You return to the Edit Host window, which now shows the Certificate Fingerprint, as shown in Figure 5-110.

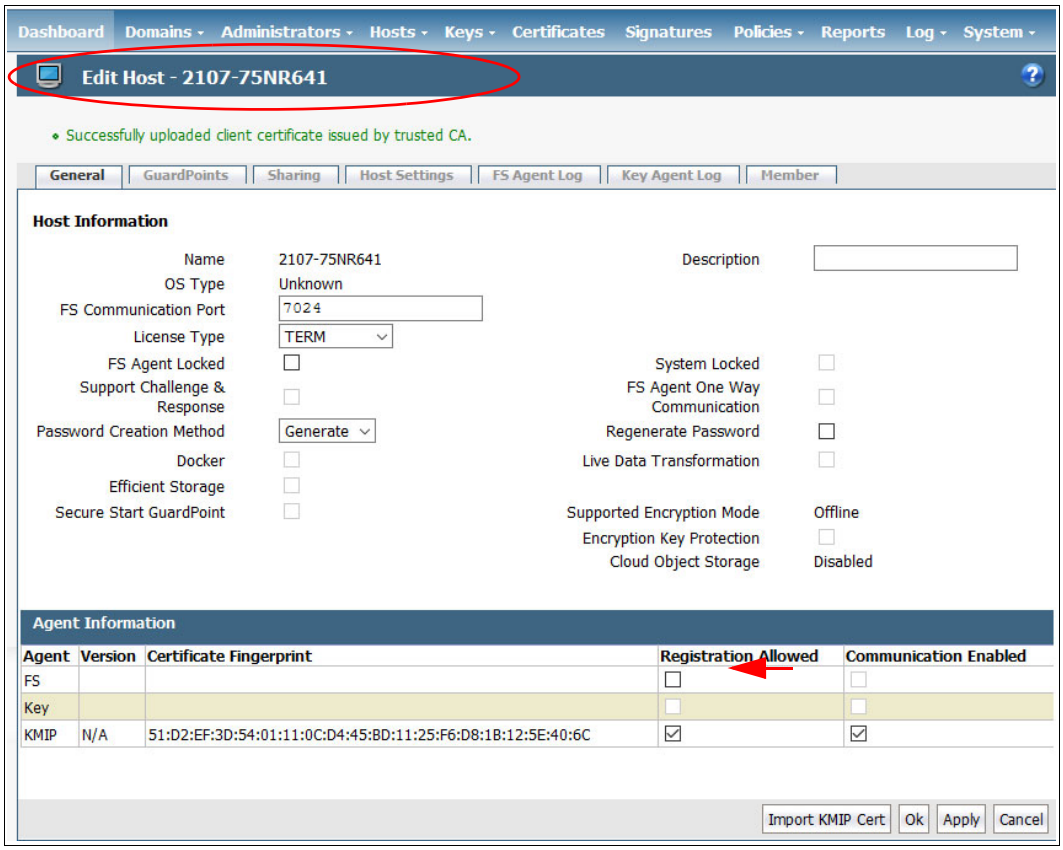


Figure 5-110 Edit Host

The configuration is now complete.

5.4.4 Configuring Thales CipherTrust Manager

Thales CipherTrust Manager is a third-party centralized key management platform like IBM Security Key Lifecycle Manager that is fully supported by DS8000 Release 9.1 and later. Thales CipherTrust Manager was formerly known as Next Generation KS.

Thales CipherTrust Manager is available as a hardware or virtual software appliance.

The examples that follow are from Version 2.0, which supports KMIP V1.1 and authentication by using a local user or LDAP/Active Directory authentication.

Like IBM Security Key Lifecycle Manager, Thales CipherTrust Manager provides a web browser-based GUI.

Thales CipherTrust Manager can manage up to 1,000,000 keys and 1,000 devices. It supports the HSM to store the master key.

For more information about Thales CipherTrust Manager, see CipherTrust Manager.

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

Preparation

When using Thales CipherTrust Manager KMIP Compatible Key Servers, KMIP must be configured with the necessary client certificate authentication policy. Two policies are supported by the DS8900F:

- Client Certificate Authentication that is used for an SSL session only
- Client Certificate Authentication for SSL Session and UID

  A UID is added to the Gen 2 and Gen 3 certificate in the DS8000 by manufacturing, which connects the DS8000 to the KMIP capable key server by using Client Certificate Authentication for SSL Session and UID. It is the more secure way.

Both methods are supported and covered in this paper.

Prerequisites

Before starting the Thales CipherTrust Manager Configuration, make sure to satisfy the following prerequisites:

- Make sure that you configured two independent key servers in a cluster.
- Configure the RK (see “Creating the recovery key” on page 161).
- Download the root certificate for the DS8000 to the client computer from IBM Knowledge Center for the DS8000, as described in 5.10, “Migrating certificates” on page 199.
- Client Certificate Authentication for SSL Session and UID only: Export the Gen 2 or Gen 3 certificate 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 “DS8000 Encryption Communication Certificate (Gen 2) export and usage” on page 119.

Note: DS8000 Release 8.1 and later features a Gen 2 certificate from manufacturing by default. DS8900F R9 features a Gen 2+ certificate that is active and a Gen 3 certificate that is not active.

Configuration

Note: Illustrations in this section are shown courtesy of Thales DIS CPL US, Inc.

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

1. Import and install the DS8000 root and intermediate CA certificates.
2. Create a registration token.
3. Configure the KMIP interface.
4. Obtain the KMIP public certificate to use as a trust anchor when creating the key server on the DS8000.
5. Create DS8000 users and add them to appropriate groups.
6. Restart the KMIP System service.

Importing and installing the DS8000 root and intermediate CA certificates

For each certificate, complete the following steps:

1. On the Thales CipherTrust Manager home page, click **Keys & Access Management** (Figure 5-111).

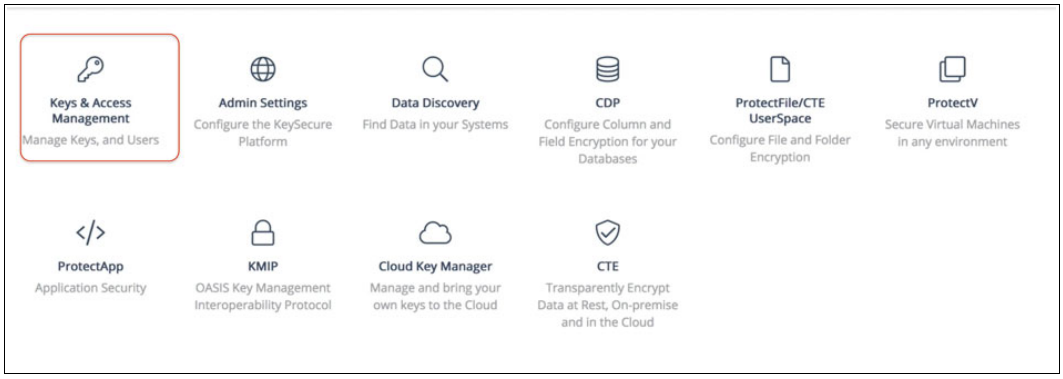


Figure 5-111 Main menu: Select Keys & Access Management

2. Add the DS8000 root and intermediate CA certificates as external CAs, as shown in Figure 5-112:
   a. To open the **Certificate Authorities** window, click **CA**.
   b. Expand **Upload External Certificate**.
   c. Paste the PEM formatted text from a DS8000 certificate into the text box and click **Upload**.
   d. Repeat for each DS8000 root and intermediate CA certificate.

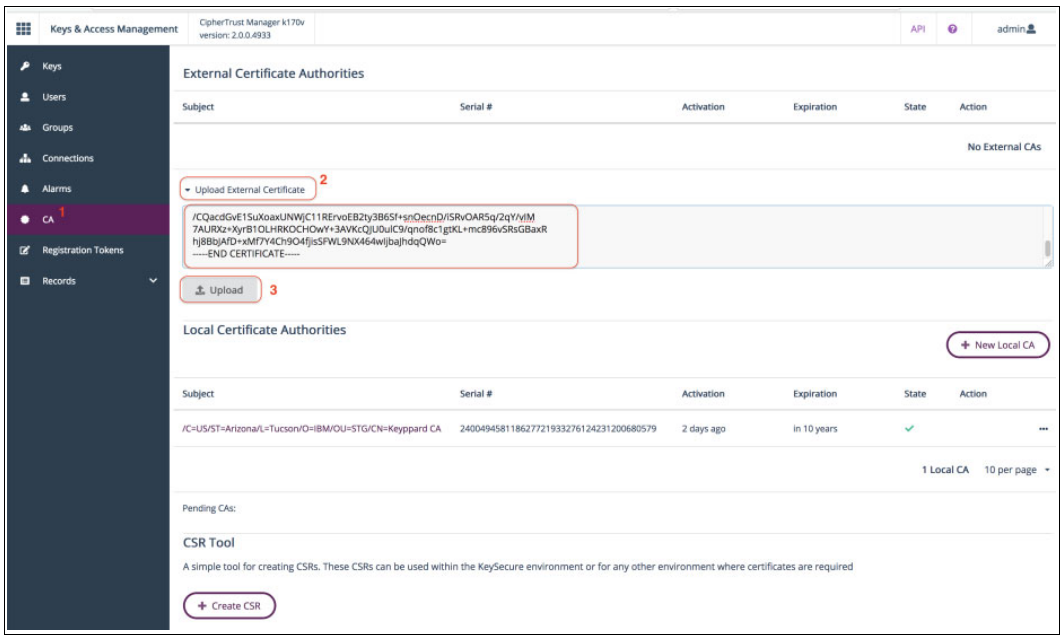


Figure 5-112 Upload the DS8000 certificate

Creating a registration token

Complete the following steps:

1. On the Thales CipherTrust Manager home page, click **KMIP** (Figure 5-113).

Figure 5-113 Main menu: Select KMIP

2. Select **Registration Token** and then **+New Registration Token** (Figure 5-114).

Figure 5-114 New Registration Token

3. Follow the wizard and copy the token when the wizard completes (Figure 5-115).

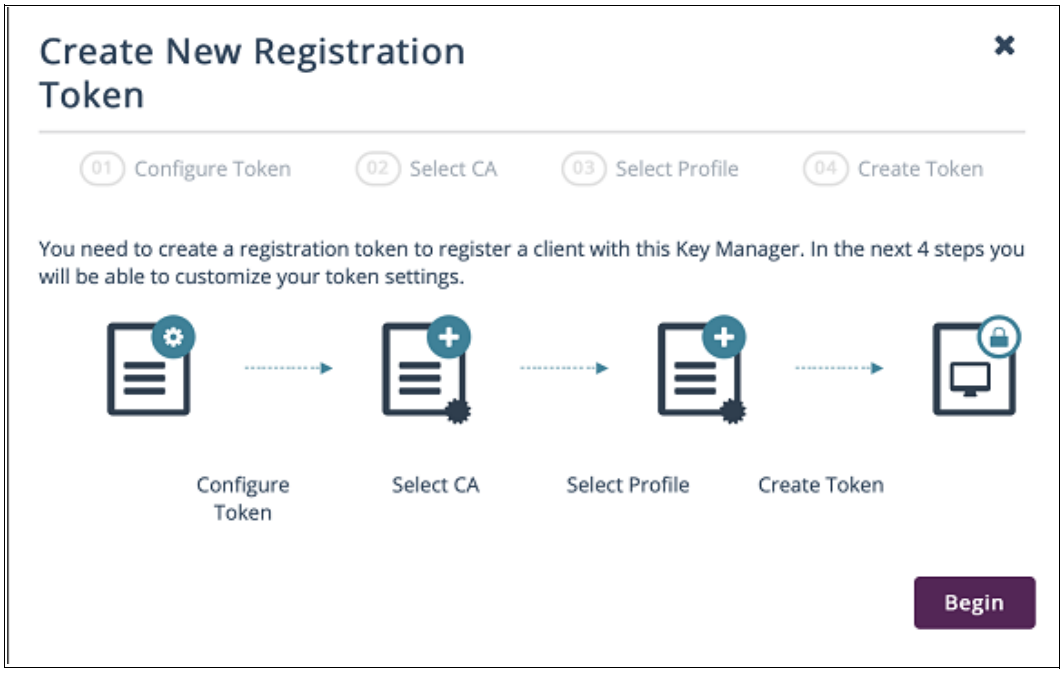


Figure 5-115 Create New Registration Token wizard

Configuring the KMIP interface

Complete the following steps:

1. Go to the KMIP Interface Configuration window:
   a. On the Thales CipherTrust Manager home page, click **Admin Settings** (Figure 5-116).

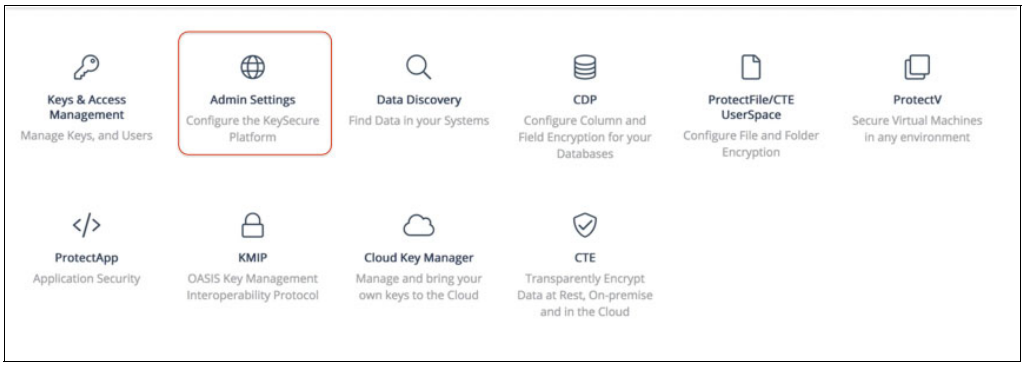


Figure 5-116 Main menu: Select Admin Settings

   b. Select **System** → **Interfaces** to open the Interface Configurations window.
   c. Click the ellipsis icon at the upper right of the **KMIP Interface Configuration** window and select **Edit** (Figure 5-117).

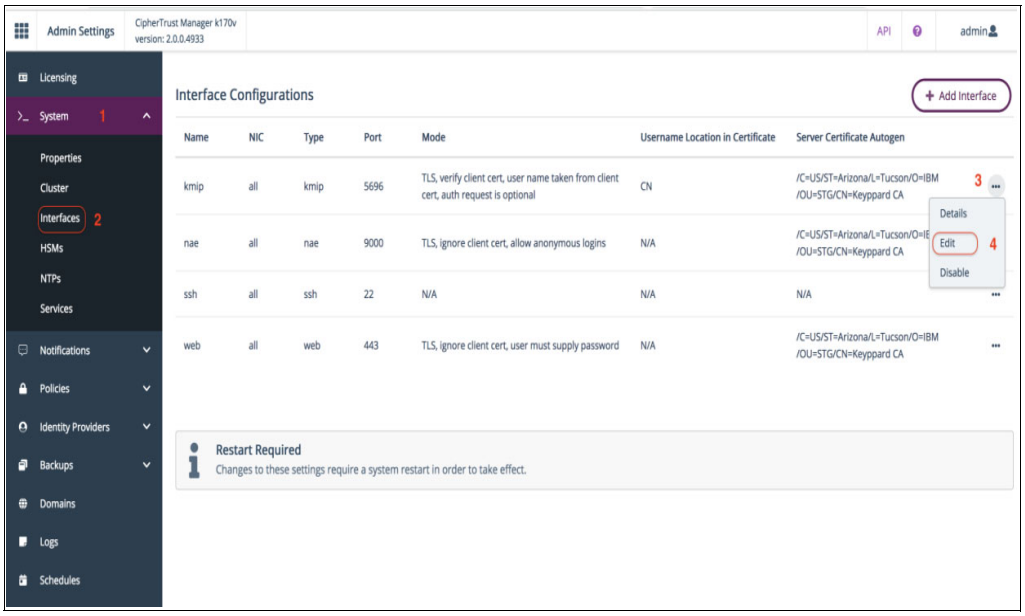


Figure 5-117 Edit the KMIP Interface

   d. Select a DS8000 CA certificate from the **External Trusted CAs** certificate list or click the **+** icon to upload a new one, as shown in Figure 5-118 on page 145. Repeat for each DS8000 root and intermediate CA certificate.

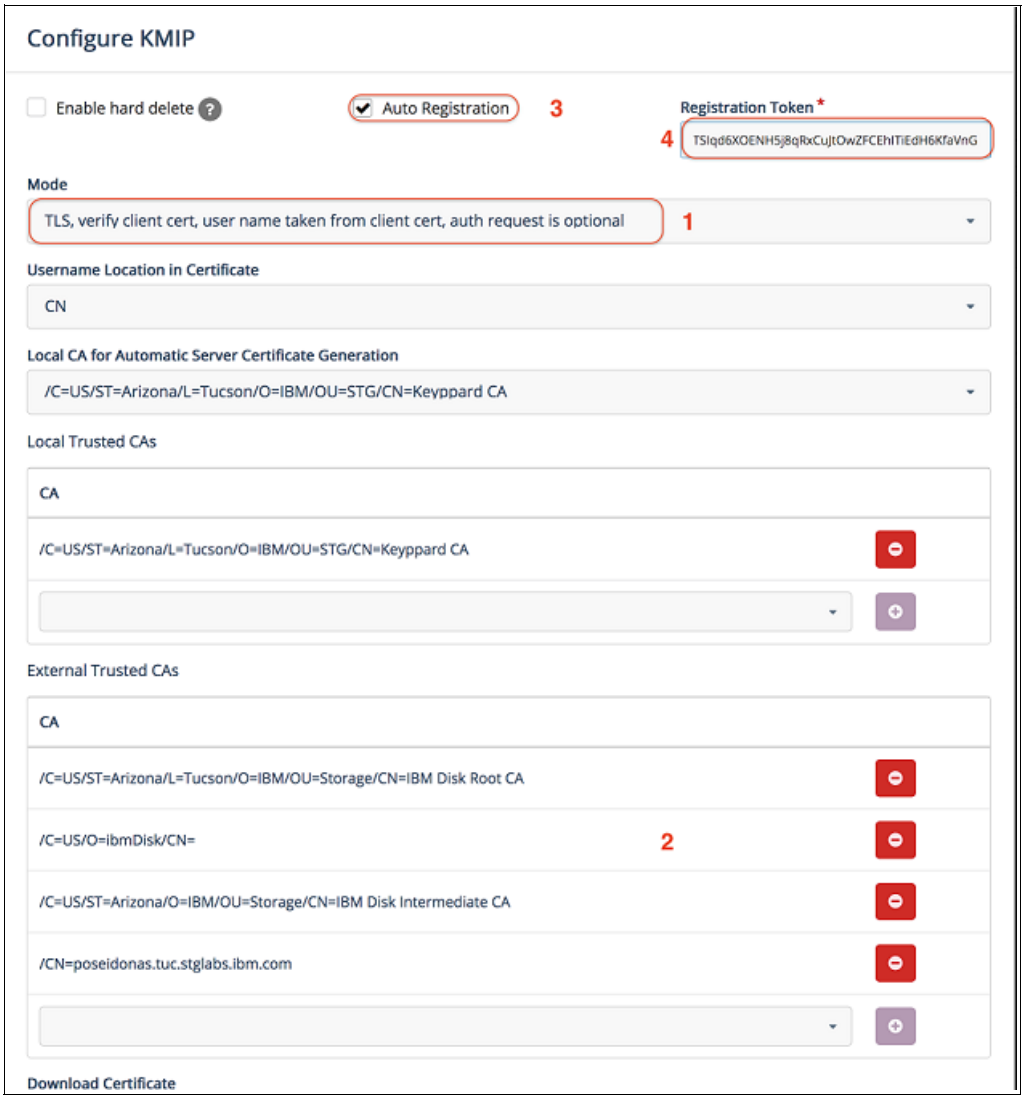


Figure 5-118 Add a CA Certificate to the KMIP interface

e. Enable **Auto Registration** and provide a registration token. The token can be created as described in “Creating a registration token” on page 143.

f. Set **Mode** to one of the following options:

   i. **TLS, verify client cert, username taken from client cert, auth request is optional** (**tls-cert-pw- opt**)

   > **Note:** The username must match the value that is created in “Creating DS8000 users and adding them to the appropriate groups”.

   ii. **TLS, verify client cert, allow anonymous logins** (**tls-pw-opt)** (recommended)

g. If **Mode** is set to “**TLS, verify client cert, username taken from client cert, auth request is optional”**, Set **Username Location in Certificate** to one of the following options:

   i. **CN**

   ii. **UID**

Obtaining the KMIP public certificate

Complete the following steps:

1. Go to the KMIP Interface Configuration window:
   a. In the Thales CipherTrust Manager home page, click **Admin Settings** (Figure 5-119).

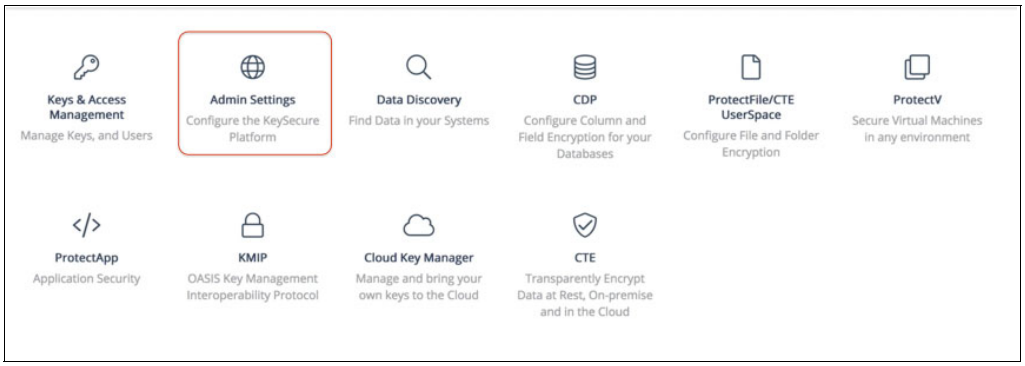


Figure 5-119 Main menu: Admin Setting selected

   b. Click the ellipsis icon at the upper right of the **KMIP Interface Configuration** window, and then select **Edit**.
   c. Click **Download Current Certificate**, as shown Figure 5-120 on page 147.

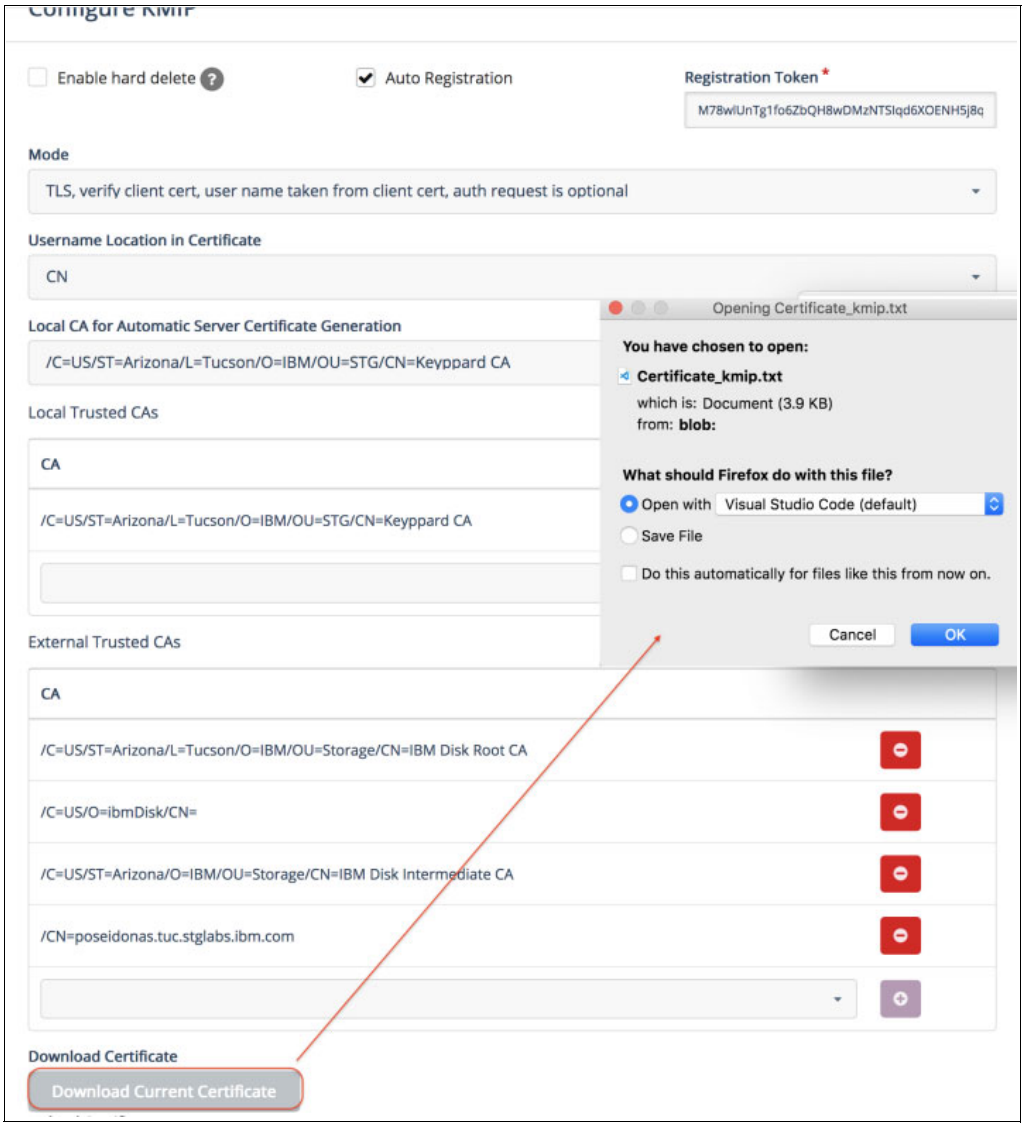


Figure 5-120 Download Current Certificate

d. Save the file.

This file is used later when creating the key server on the DS8000 system.

Creating DS8000 users and adding them to the appropriate groups

Complete the following steps:

1. Create a user:

   a. In the Thales CipherTrust Manager home page, click **Key and Access Management**, as shown in Figure 5-121.

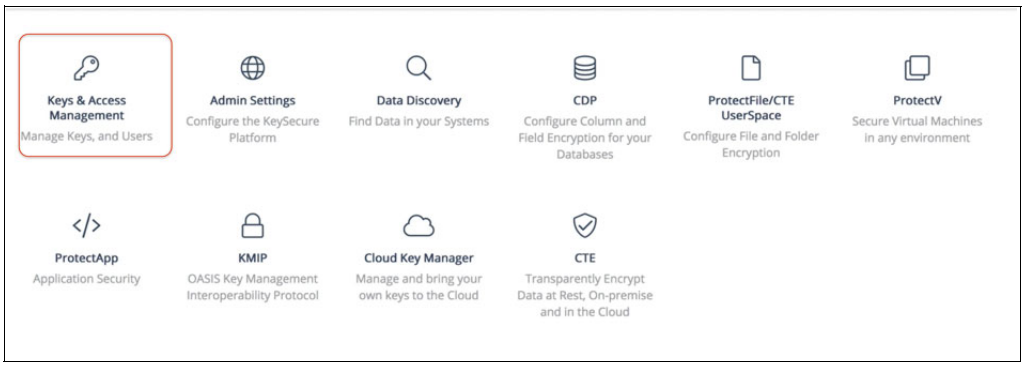


Figure 5-121 Main menu: Select Key & Access Management

   b. Click **Users**, and then click **Create New User**, as shown in Figure 5-122.

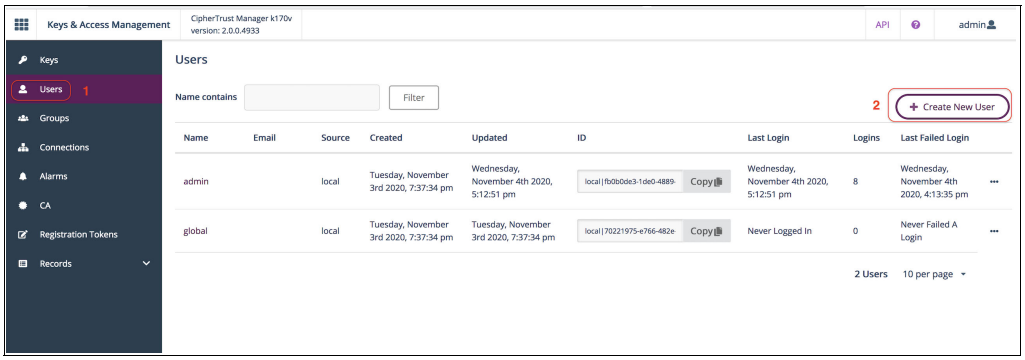


Figure 5-122 Create New User

   c. Complete the required fields, as shown in Figure 5-123 on page 149.

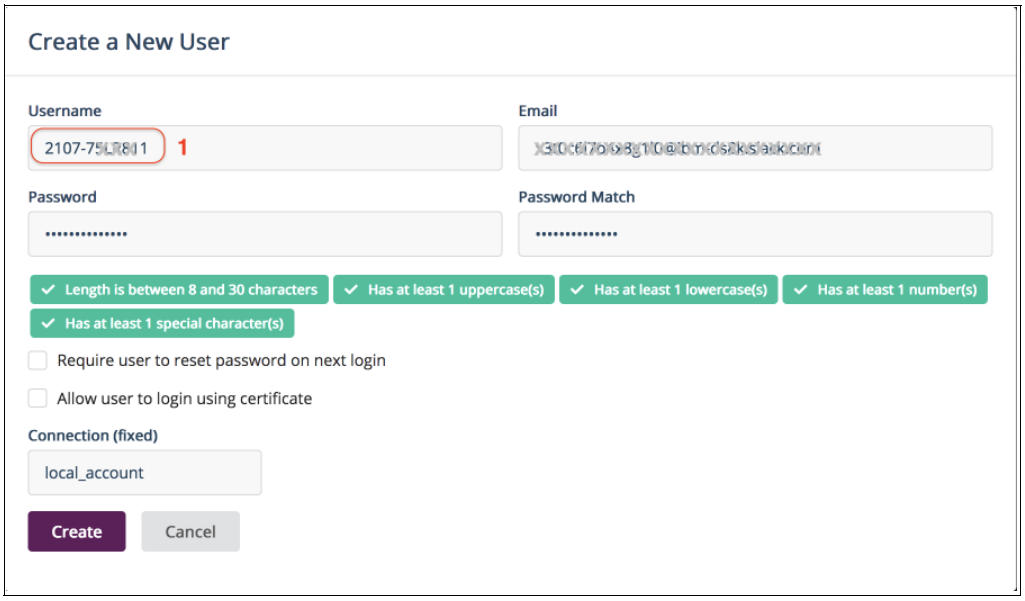


Figure 5-123 Create New User Fields

d. Verify the user details in the **Users** window.

2. Add the user to groups:

a. Click the username (Figure 5-124).

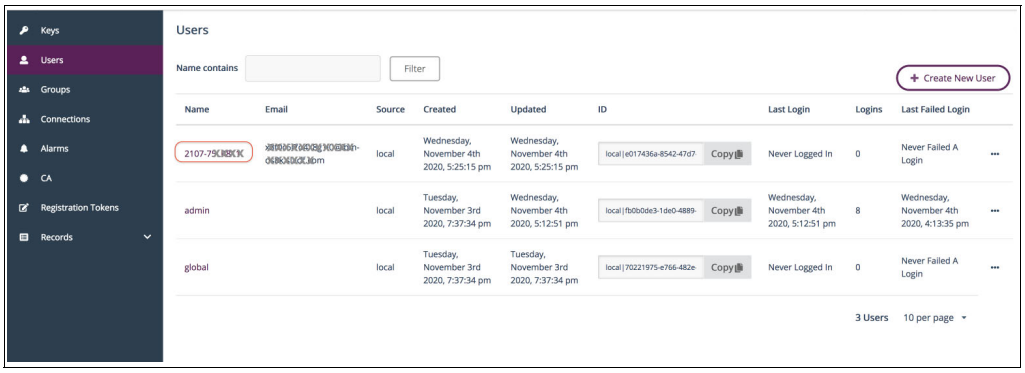


Figure 5-124 Select the username

b. Expand **Groups**, find the group, and click **Add** (Figure 5-125).

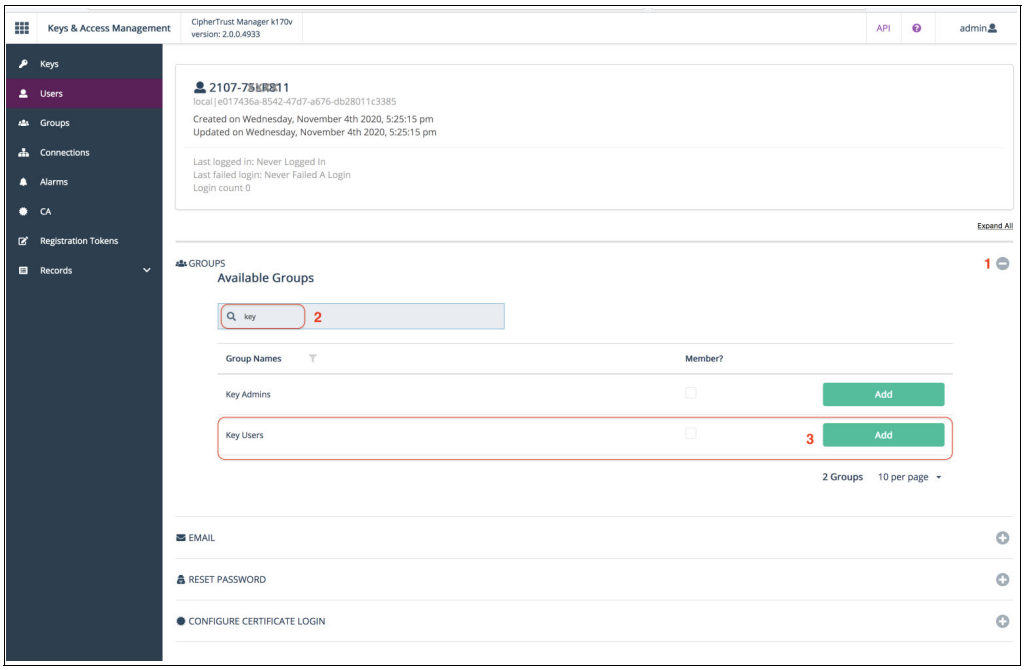


Figure 5-125 Add groups to user

> **Note:** The user must be added to a group with permissions to create keys and perform operations on keys that they own.

c. For DAR, you may use the default Key Users group.

d. For TCT, if there are multiple DS8000 systems that can retrieve an object, the user should also be a member of a user-defined group, and each key should have the group added for key sharing.

3. Complete the required fields.

> **Note:** If the KMIP interface mode requires **Username taken from client certificate**, the value of the username must match the value of the required field that is contained in the DS8000 client public certificate.

4. Verify the user details in the Users window.

Restarting the KMIP system service

Complete the following steps:

1. Select **System** → **Services**.
2. A window opens to confirm the KMIP restart. Click **Restart kmip** (Figure 5-126 on page 151).

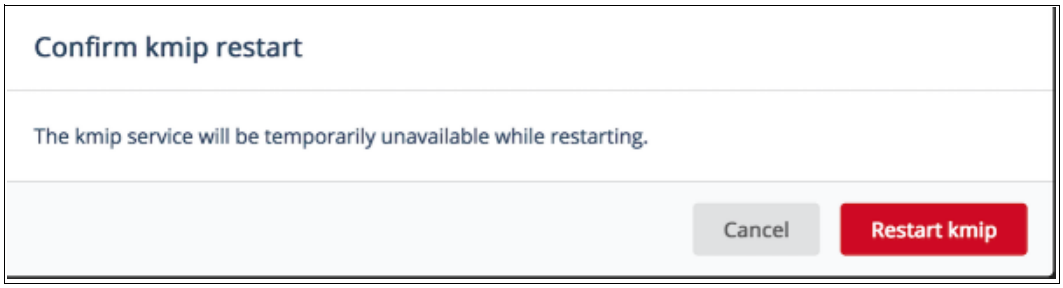


Figure 5-126 Restart KMIP

The Thales CipherTrust Manager server is now ready to serve keys to the DS8000 server.

5.5 Configuring data at rest

This section describes the following topics:

- Setting up IBM Security Key Lifecycle Manager Key management by using IBM Proprietary Protocol
- IBM Security Key Lifecycle Manager Key management setup by using KMIP
- DS8000 configuration for data at rest encryption in DS GUI
- DS8000 CLI configuration for data at rest encryption
- Various authentication mechanisms in IBM Security Key Lifecycle Manager

5.5.1 Setting up IBM Security Key Lifecycle Manager Key management by using IBM Proprietary Protocol

With the IBM Security Key Lifecycle Manager key server setup complete, you can now add the configuration details for DS8000 DAR encryption with IBM Proprietary Protocol or KMIP, as described in this section. For more information about TCT encryption, see 5.6, “Configuration for TCT encryption” on page 179. For more information about IBM Fibre Channel Endpoint Security, see 5.7, “Configuration for IBM Fibre Channel Endpoint Security” on page 188.

Configuring IBM Security Key Lifecycle Manager for a data at rest configuration with IBM Proprietary Protocol

Complete the following steps:

1. From the Welcome window, in the Key and Device Management pane, select **DS8000**. Then, from the **Go to** 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 5-127 shows these selections.

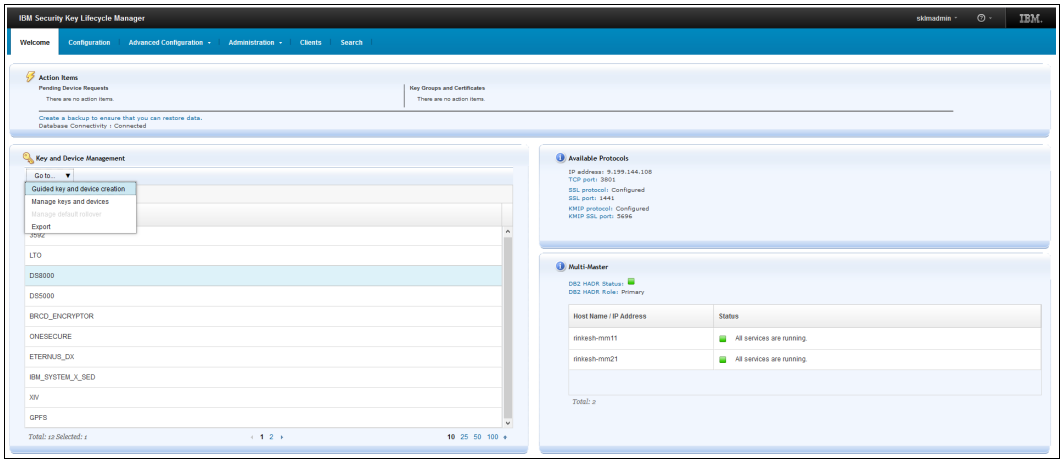


Figure 5-127 Guided key and device creation for DS8000 in IBM Security Key Lifecycle Manager

2. Click **Create** to start creating the certificate (Figure 5-128) that will be associated with the storage facility image.

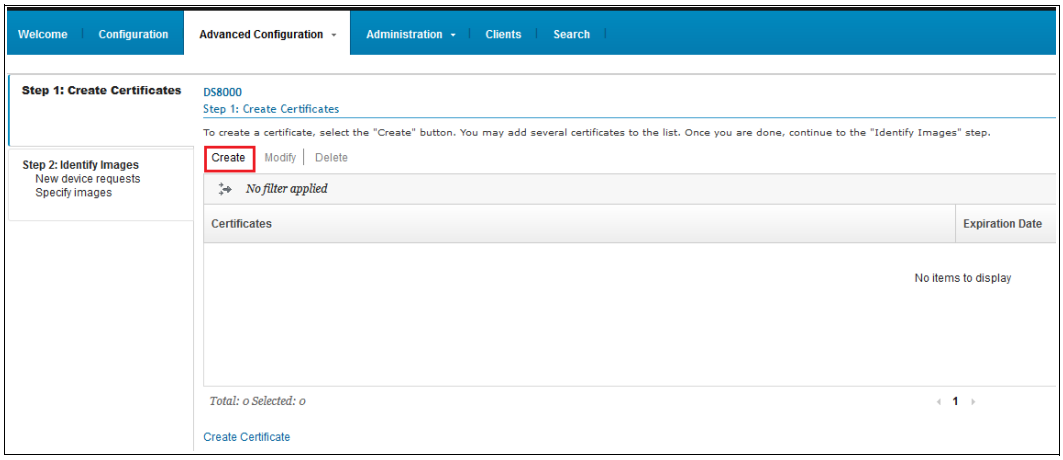


Figure 5-128 Create the DS8000 certificate

3. After completing the required fields (see Figure 5-129 on page 153), click **Create Certificate**.

> **Note:** Do not confuse this certificate with the one that was created in “Option 1: Creating a self-signed SSL/KMIP server certificate” on page 96 for network communication.

This certificate that is created here is used with the DS8000 storage image to encrypt the data key (DK). The maximum validity period of a certificate is 9000 days (more than 24 years).

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

To use a third-party signed SSL certificate, follow the same steps to create the request, and export and import the signed certificate, as described in “Option 2: Creating a certificate that is signed by a third-party provider” on page 97, but for the Device Certificate this time, and not for the SSL / KMIP Server Certificate.

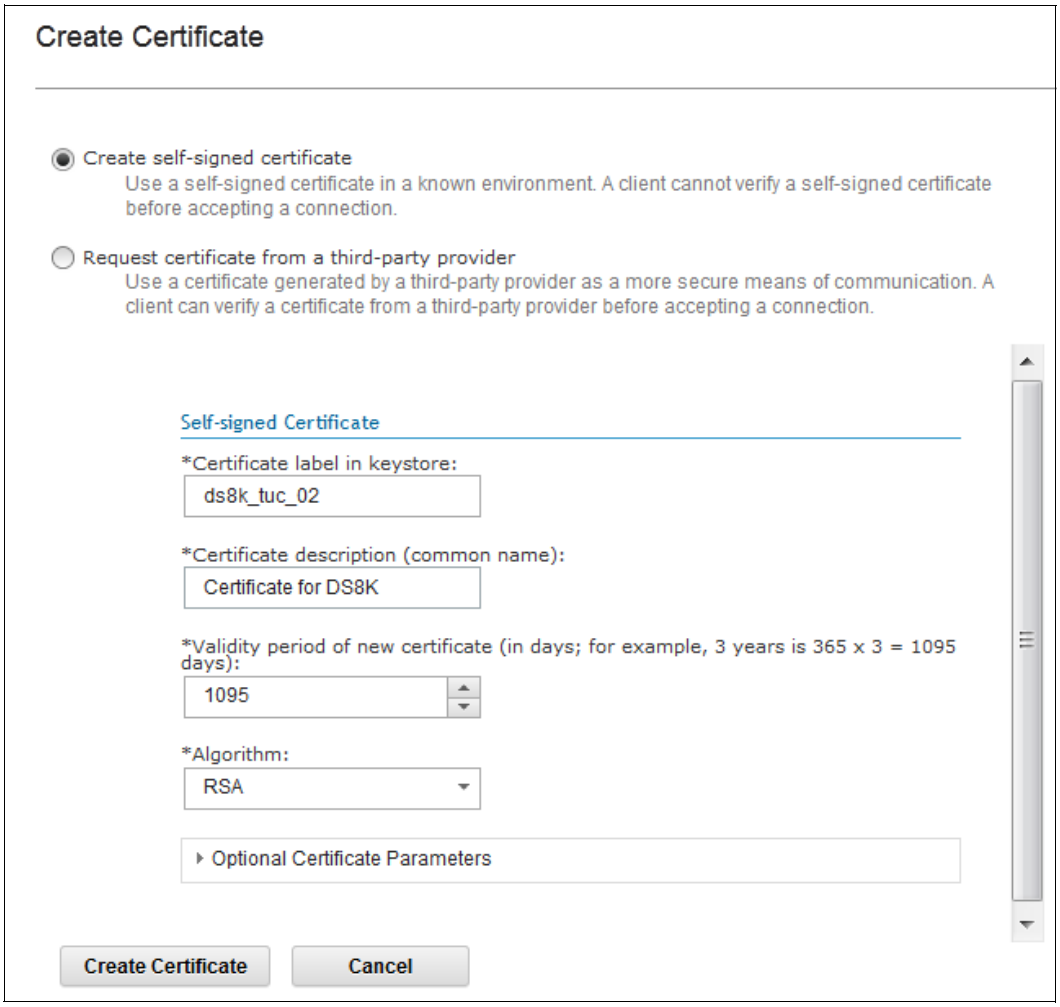


Figure 5-129 Create Certificate window to use for the storage facility image

After the certificate is created, a warning to create a backup displays (see Figure 5-130). 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 for DK encryption. You can wait until after all storage images are defined to create the backup. It takes about 2 minutes to create the backup, and there is no progress indicator.

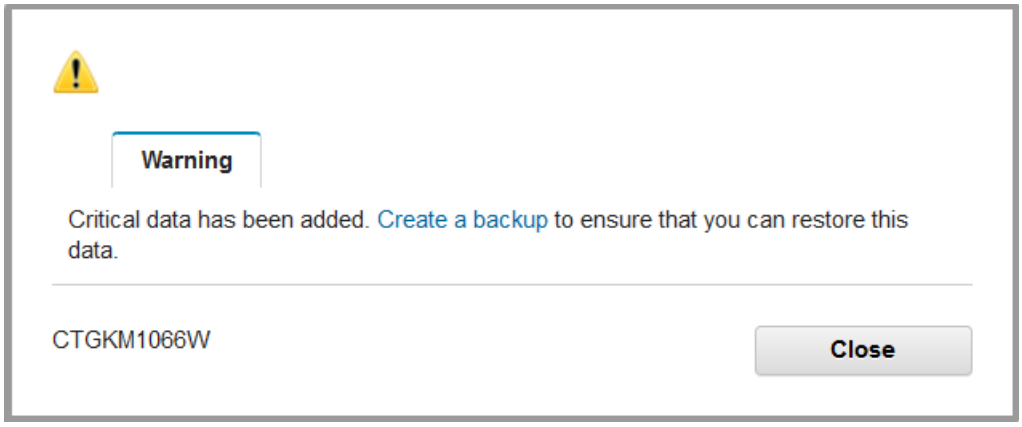


Figure 5-130 Warning to ensure the creation of a backup

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 5-131.

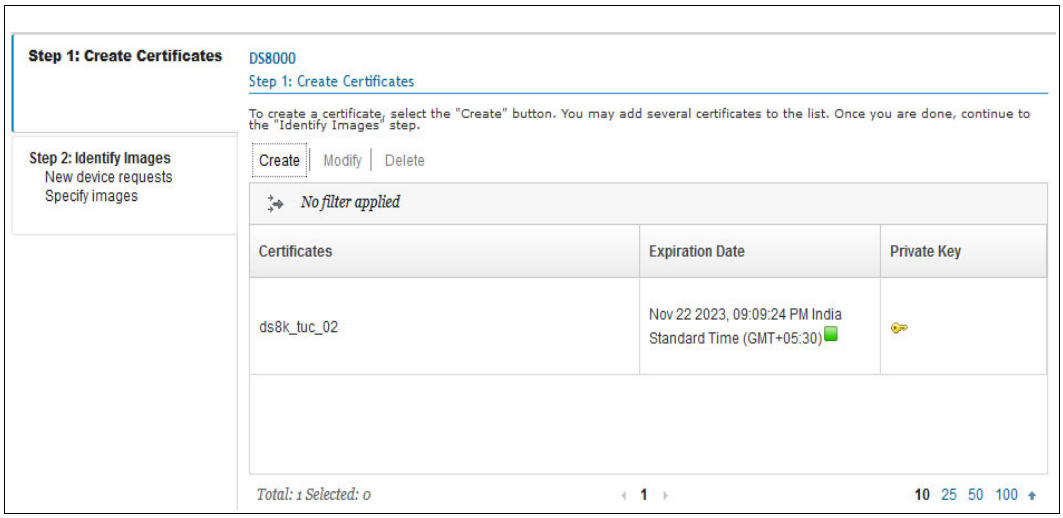


Figure 5-131 Create Certificate window for DS8000 Guided Path

6. In the Step 2: Identify Images window (Figure 5-132 on page 155), add the storage image that is to be managed by the key server.

   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.

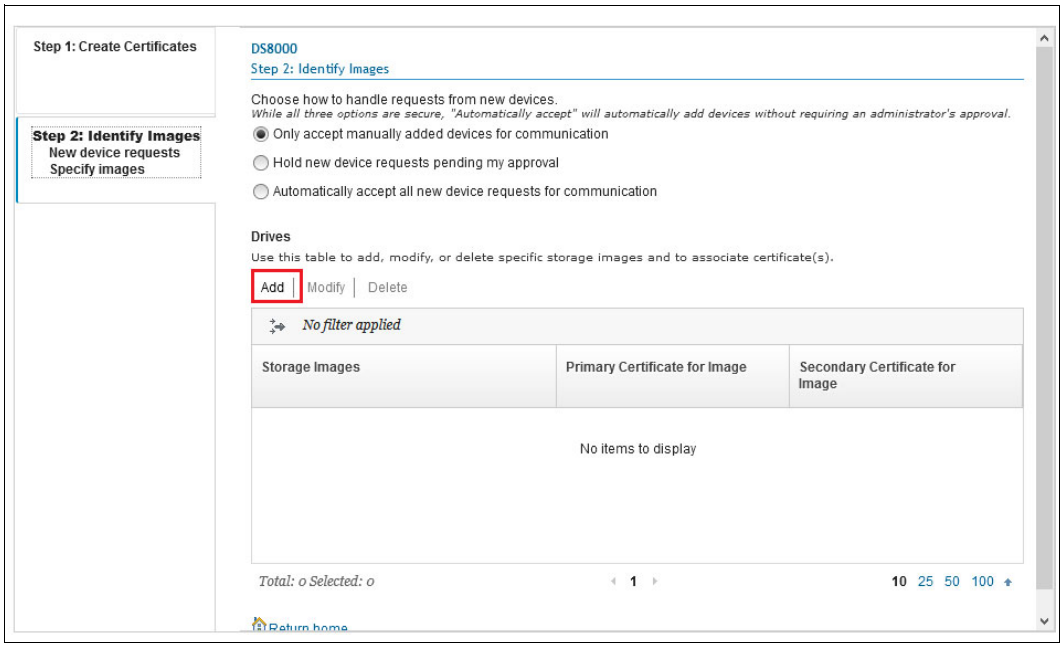


Figure 5-132 Add a storage image

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 5-133 shows the correct format of the information that is required. All DS8000 models use the 2107-75XXXX1 format for the serial number.

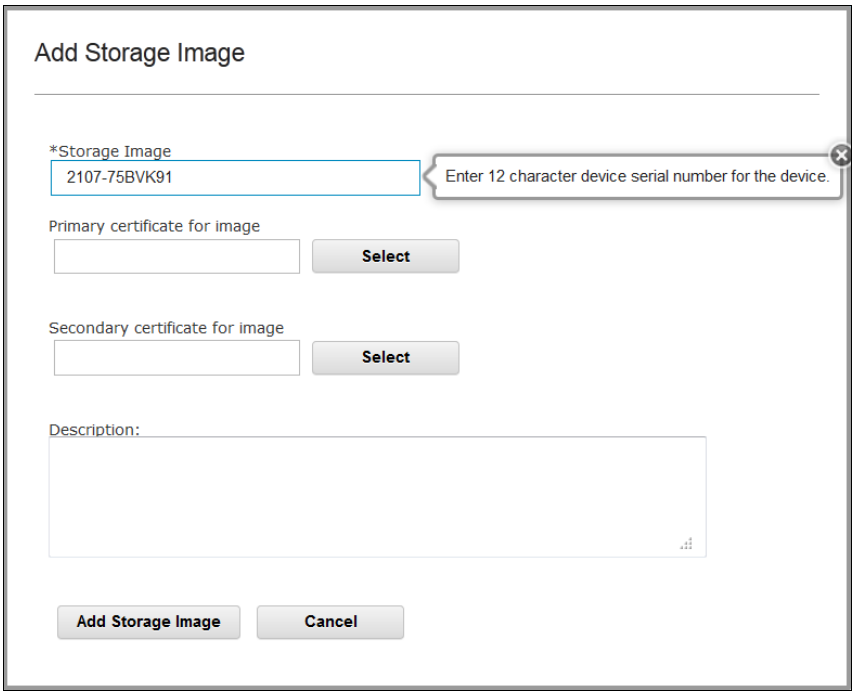


Figure 5-133 Add Storage Image window

In that window, if you click **Select** after the **Primary certificate for image** field, you see the window that is shown in Figure 5-134, in which all certificate names are listed.

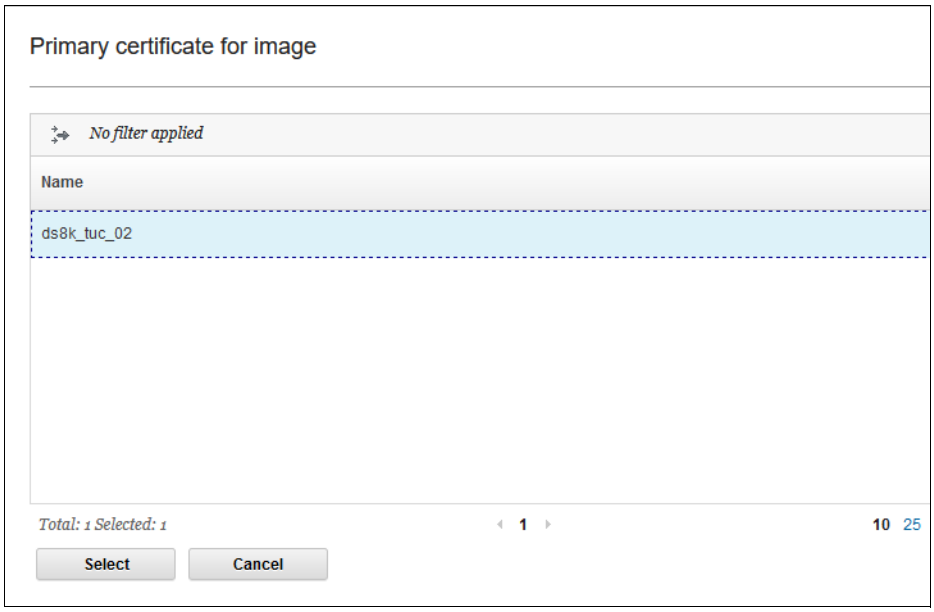


Figure 5-134 Selecting the primary certificate for image

8. Select the certificate (**ds8k_tuc_02** in this example).

   After selecting the primary certificate for image, you return to the Add Storage Image window that is shown in Figure 5-135. Click **Add Storage Image** to complete the task.

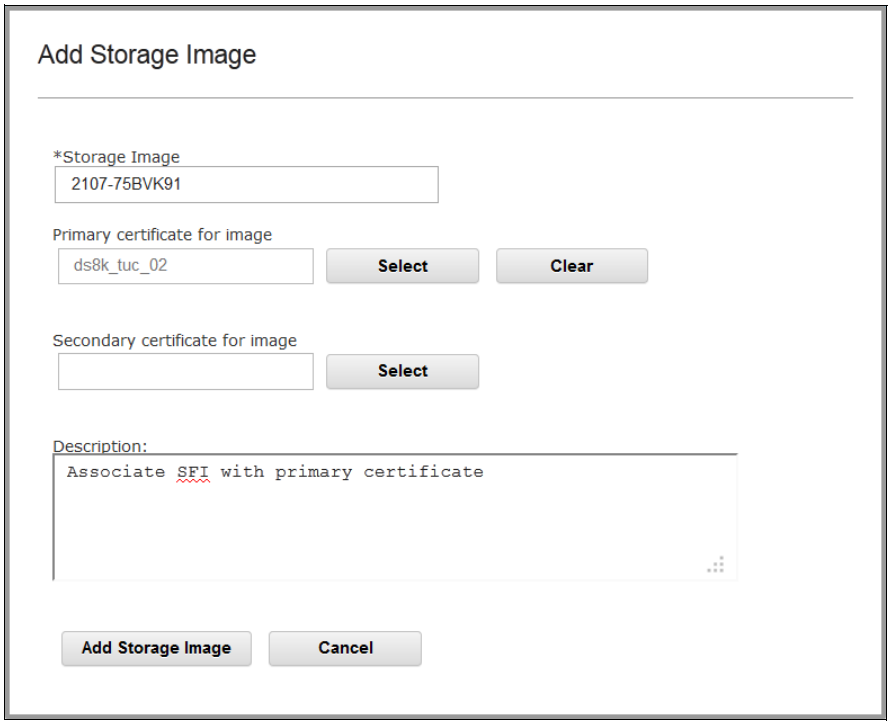


Figure 5-135 Add Storage Image window

The task to add a storage image is complete, as shown in Figure 5-136.

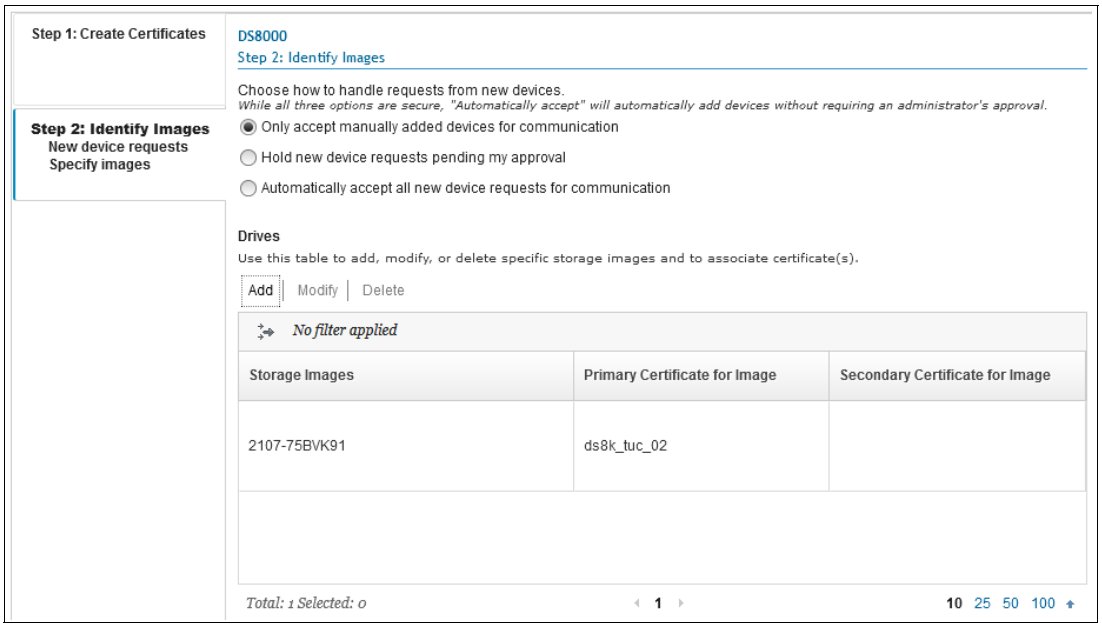


Figure 5-136 DS8000 machine defined

Verifying the DAR configuration with IBM Proprietary Protocol

Complete the following steps:

1. Verify that the storage image and certificate information are correct. From the Welcome window, select **DS8000** and under Key and Device Management, select **Manage keys and devices**, as shown in Figure 5-137. It is a good idea to verify all changes when adding storage images.

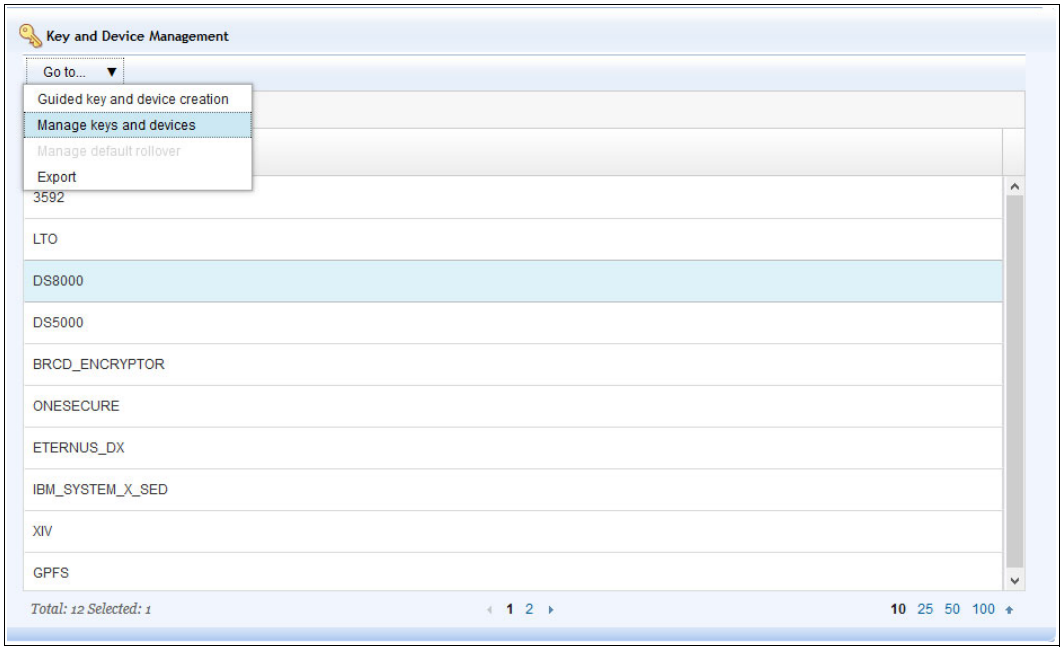


Figure 5-137 Manage keys and devices menu

2. Check all the keys and storage images to verify that the information is correct, as shown in Figure 5-138.

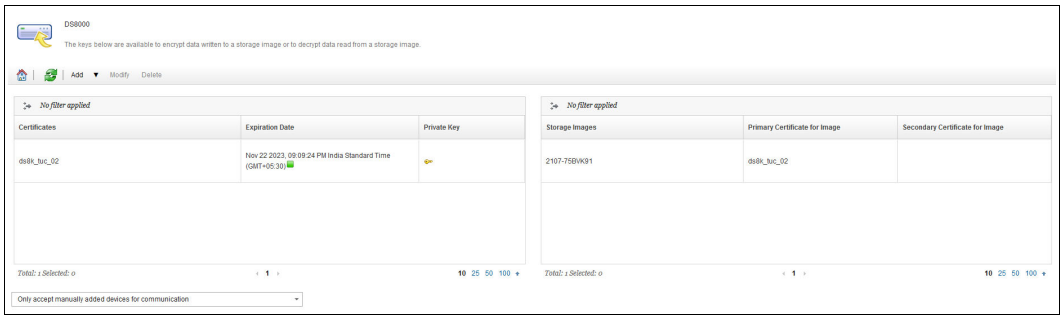


Figure 5-138 Check the keys and storage images

3. If remote replication is enabled, go to the **Replication configuration** menu and manually replicate, as described in step 4 on page 110.

   If no backup was created, create one now, as described in “Backup and restore” on page 101.

   In Multi-Master environments, the certificate and devices are automatically synchronized. No other actions are necessary.

5.5.2 IBM Security Key Lifecycle Manager Key management setup by using KMIP

If you are using KMIP instead of IBM Proprietary Protocol, no other setup is required in the IBM Security Key Lifecycle Manager server. Directly proceed with 5.5.3, “DS8000 configuration for data at rest encryption in DS GUI” on page 158.

5.5.3 DS8000 configuration for data at rest encryption in DS GUI

This section explains how to configure DAR encryption on the DS8000 by using the DS GUI and the DS CLI. The high-level configuration sequence includes the following steps (DS8000 encryption is part of the Base Function license and no longer requires a license key):

1. Create more *storage* and *security* administrator users.
2. Create the RK.
3. Save the RK in a secure location.
4. Enable encryption:
   a. If you use IBM Security Key Lifecycle Manager:
      i. Define the key labels (if IBM Proprietary Protocol is used).
      ii. Configure the key server connection to the DS8000.
      iii. Authorize the RK.
   b. If you use Thales Vormetric DSM or Thales CipherTrust Manager:
      i. Configure the key server connection to the DS8000.
      ii. Authorize the RK.
5. Configure and administer encrypted arrays, ranks, and storage pools.

For more information about enabling NIST SP 800-131a-compliant encryption certificates and TLS V1.2 communication, see 5.9, “NIST SP 800-131a requirements for key servers” on page 194.

Creating more Storage and Security Administrators by using the DS GUI

The DS8000 includes an internal authentication and authorization service that is called the *basic authentication service*. This service also provides local user management. With the DS8000, you 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`
- `ibm_engineering`
- `ibm_service`

With the introduction of the encryption RK on DS8000, a *dual control* security process is required to prevent unauthorized use of the RK. This dual control process requires two separate user accounts to process most recovery commands. If these accounts are owned by two separate people, the RK 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 should assign these roles to two individuals.

To define new UIDs or modify the default usernames, 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 then, select **Users**. as shown Figure 5-139.

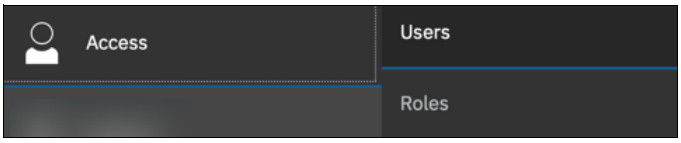


Figure 5-139 Go to the User Administration window

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. Because these users are owned by different individuals, the admin and secadmin passwords must not be stored in one place.

The list of users that are filtered for admin is shown in Figure 5-140. The full unfiltered list includes all of the users on the system. The default users are `admin` and `secadmin`.

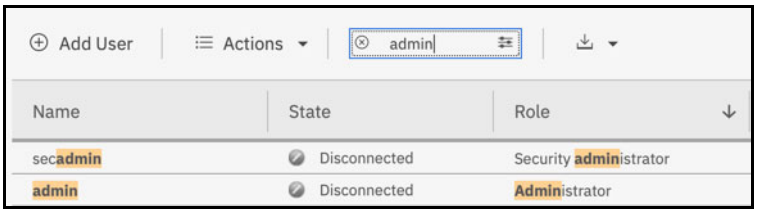


Figure 5-140 User Administration window

3. Click the **Add User** option to create users (see Figure 5-141).

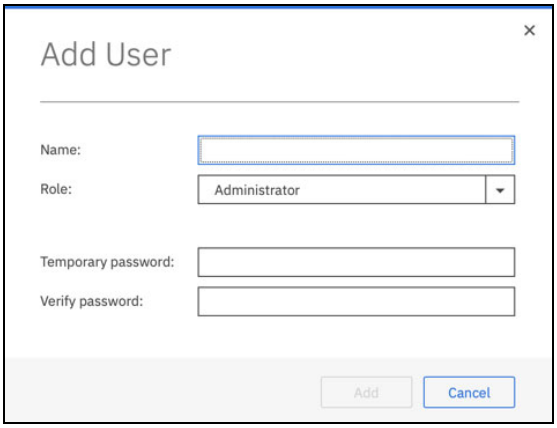


Figure 5-141 Add User

A user who has the Security Administrator role 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 is required to create 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 5-142 on page 161. The security administrator role is required for creating the RK.

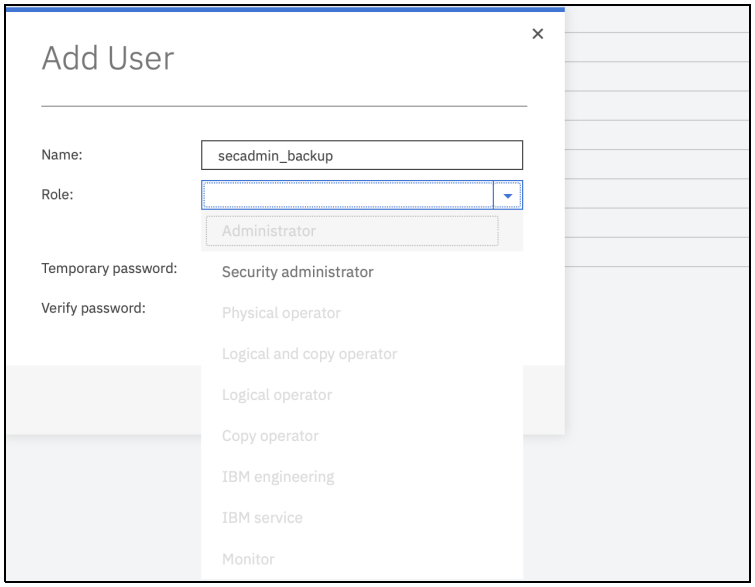


Figure 5-142 Add a user by using the Security administrator role

4. After you enter the new username, Security Administrator role, and temporary password and verification, click **Add** to complete this task.

Creating the recovery key

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

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

Without the RK, the DS8000 data becomes unrecoverable if access to the key servers is permanently lost (typically if the key servers are corrupted and not recoverable).

Note: Creating an RK applies only for DAR encryption. It is not supported by TCT encryption or IBM Fibre Channel Endpoint Security.

The decision about whether to create the RK must be made at this stage. Creation enables the RK automatically. If you decide to proceed without creating the RK, 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 RK during initial configuration and later decide to disable it.

To create an RK, 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 that is shown in Figure 5-143 opens.

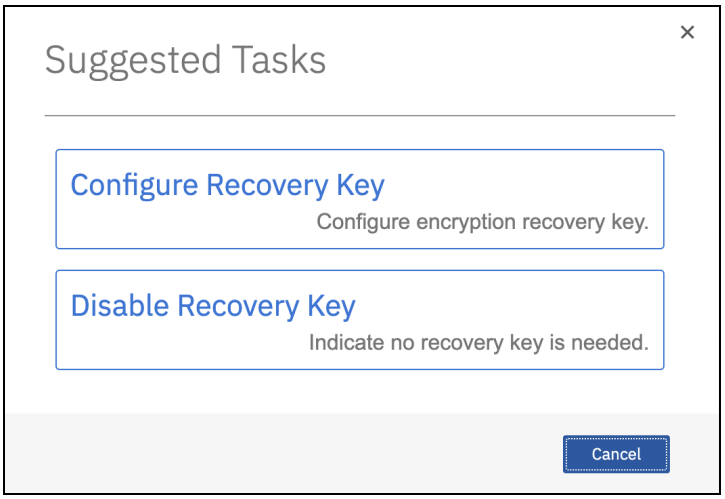


Figure 5-143 Configure Recovery Key

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 RK. 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, as shown in Figure 5-144.

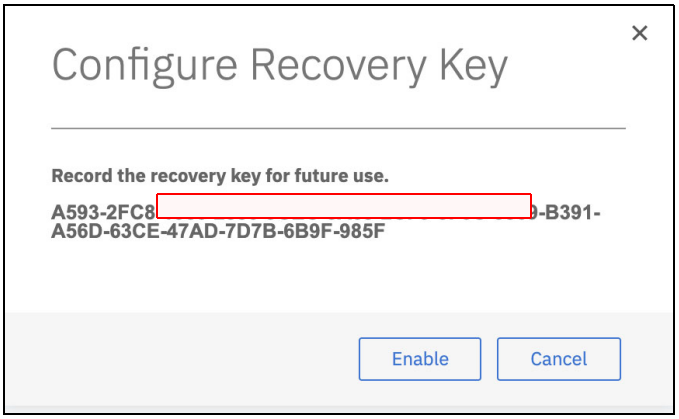


Figure 5-144 Configure Recovery Key (or disable recovery key) window

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

3. To ensure that the RK was recorded correctly, enter it into the Verify Recovery Key field (see Figure 5-145) and click **Verify**. This information also can be pasted into the field.

Figure 5-145 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 RK. At this stage, you can log off as the Security Administrator user, log back in with the Storage Administrator access and continue as described in the next section.

DS8000 enabling data at rest encryption

The data that is encrypted within the DS8000 is partitioned in one *encryption key group*. The encryption key group that contains encrypted data is enabled to access data through one DK that is obtained from a key server. After DAR encryption is enabled, all data on the system is encrypted.

> **Note:** As of this writing, the DS8000 supports only one encryption key group for DAR encryption. It must be configured for IBM Proprietary Protocol or KMIP. The IBM Proprietary Protocol is supported by IBM Security Key Lifecycle Manager key servers only. Separate encryption groups are used for TCT and IBM Fibre Channel Endpoint Security encryption.

Enabling DAR encryption with the IBM Security Key Lifecycle Manager by using KMIP requires a Multi-Master configuration of the key server. The key servers do not require any other configuration to serve keys to the DS8000.

KMIP is supported starting with DS8000 Release 8.5 and IBM Security Key Lifecycle Manager V3.0.0.2.

To configure the DS8000, complete the following steps:

1. After the RK is created, log on to the DS8000 GUI as a user with Administrator role to enable the encryption and authorize the previously generated RK. From the DS8000 GUI Welcome window, click **Settings** and then, **Security**, as shown in Figure 5-146.

Figure 5-146 Go to the Encryption window

2. The encryption wizard is shown in Figure 5-147. This wizard is started only when you enable the encryption for the first time. Click **Enable Encryption** to continue.

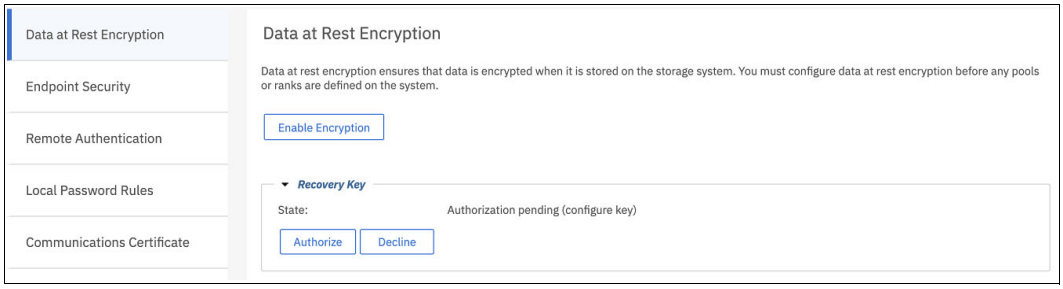


Figure 5-147 Encryption wizard: Enable Encryption

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 must be configured and online and connected to the DS8000 (see Figure 5-148 on page 165). Click **Next** to continue.

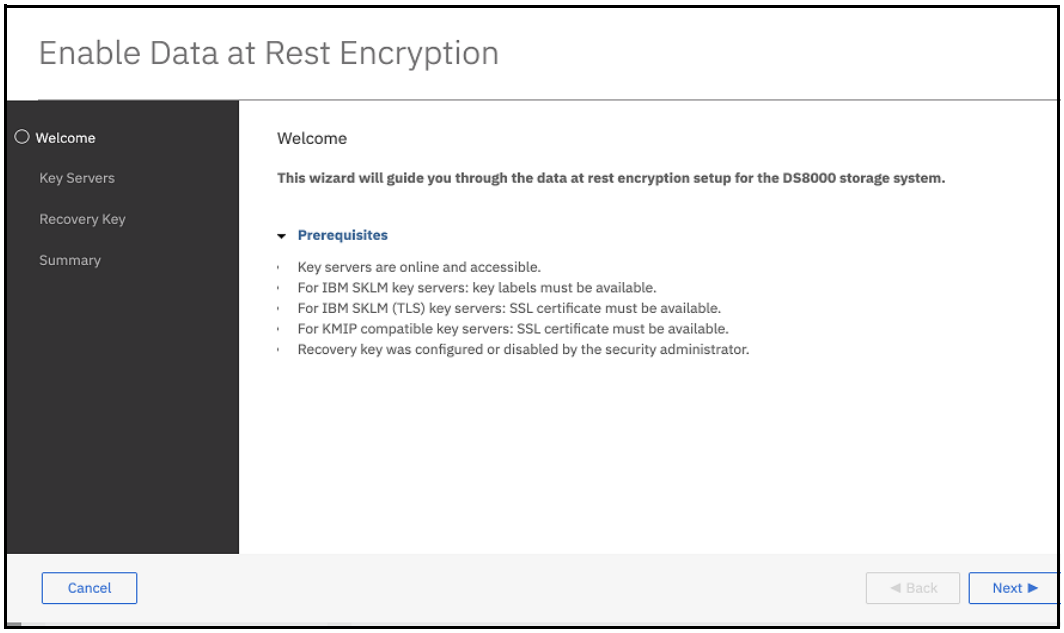


Figure 5-148 Encryption wizard: Welcome window

3. Select the Key Server Type depending on the setup:
   - Select **IBM SKLM (TLS)** to use TLS port 441 to communicate with IBM Security Key Lifecycle Manager if IBM Proprietary Protocol is used.
   - If KMIP is used, select **KMIP Compatible (TLS)**, as shown in Figure 5-149. It is the default selection.

Figure 5-149 Type selection

Note: If you select IBM Security Key Lifecycle Manager as Key Server Type, the default port number for the key servers will be set to 3801. Port 3801 is the default IBM Proprietary Protocol TCP port for IBM Security Key Lifecycle Manager; it is not secured by TLS and therefore not recommended.

However, if you choose that option at your own risk instead of TLS, then you must NOT provide key server certificates as part of key server configuration.

If you select Key Server Type **IBM SKLM (TLS)**, the key servers' port will be set to 441 and you will be required to provide key server certificates, even if you change the key server port number to the IBM Proprietary Protocol TCP port of the IBM Security Key Lifecycle Manager, which defaults to 3801, as shown in Figure 5-150.

If your key server port number is the IBM Proprietary Protocol TCP port of the IBM Security Key Lifecycle Manager and you configured a certificate when creating the key server object, key server communication will fail because the IBM Security Key Lifecycle Manager is communicating through TCP and the DS8000 is communicating via TLS. This situation results in errors during encryption enablement.

Figure 5-150 shows the default ports as displayed in the IBM Security Key Lifecycle Manager GUI Welcome window.

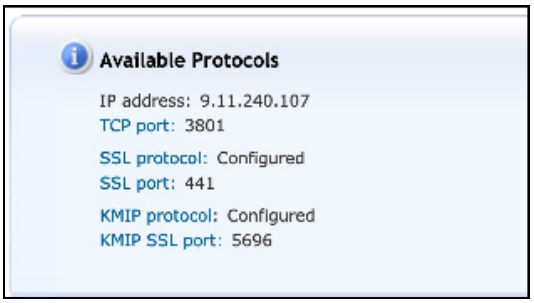


Figure 5-150 Default ports in IBM Security Key Lifecycle Manager

4. 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 5-151, two key servers are defined. You can add up to four or remove key servers by clicking the + or - sign next to each key server field. Specify the host address (IP address or full qualified hostname of the key server). Click **Next**.

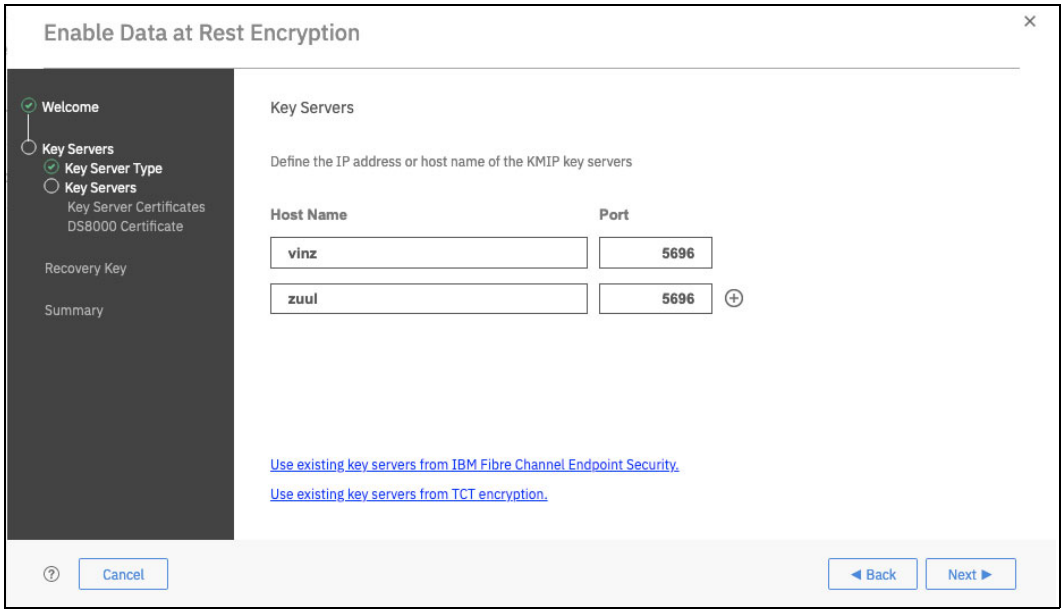


Figure 5-151 Encryption wizard: Define key servers

5. Each key server connection is tested. The message that is shown in Figure 5-152 is displayed if all the key servers that you defined 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.

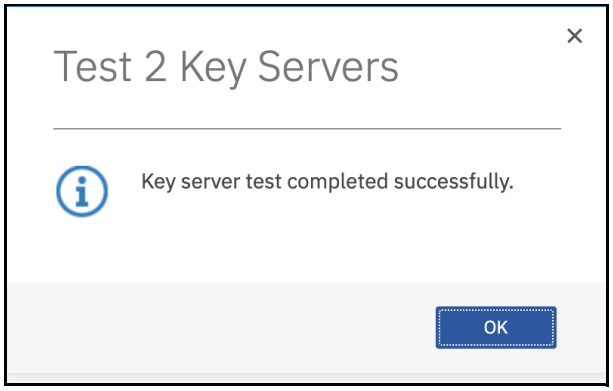


Figure 5-152 Encryption wizard: Test Key Servers

6. If the setup is using IBM Proprietary Protocol, continue with step 8. If KMIP is used, continue with step 9.

7. Define the key label for the DK 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 the KMIP protocol is used. This key label must match the one that is defined as described in 5.5.1, “Setting up IBM Security Key Lifecycle Manager Key management by using IBM Proprietary Protocol” on page 151. In this example, this key label is named `ds8900_cert`.

   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 an IBM System z® server (z/OS) and the other on the open systems platform because of the different keystore type that is used on IBM Z servers. In addition, the IBM Security Key Lifecycle Manager on IBM Z does not support TCT encryption.

   In Figure 5-153, 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. You can continue by using the encryption wizard, although 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.

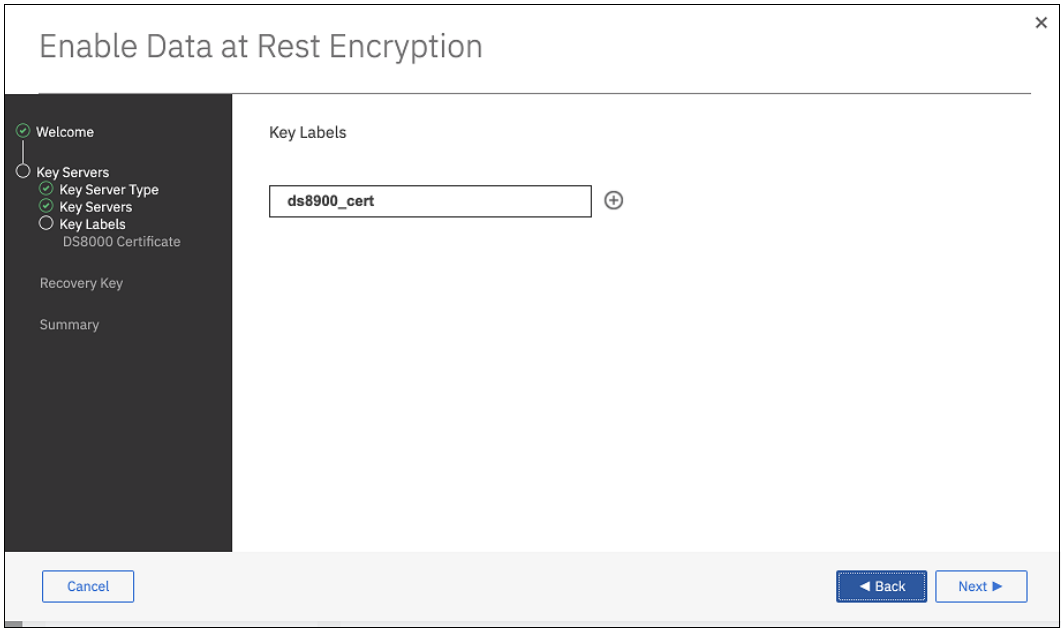


Figure 5-153 Encryption wizard: Define the key label

8. Transfer the SSL certificates from the key servers to the DS8000 server, as shown in Figure 5-154 and Figure 5-155.

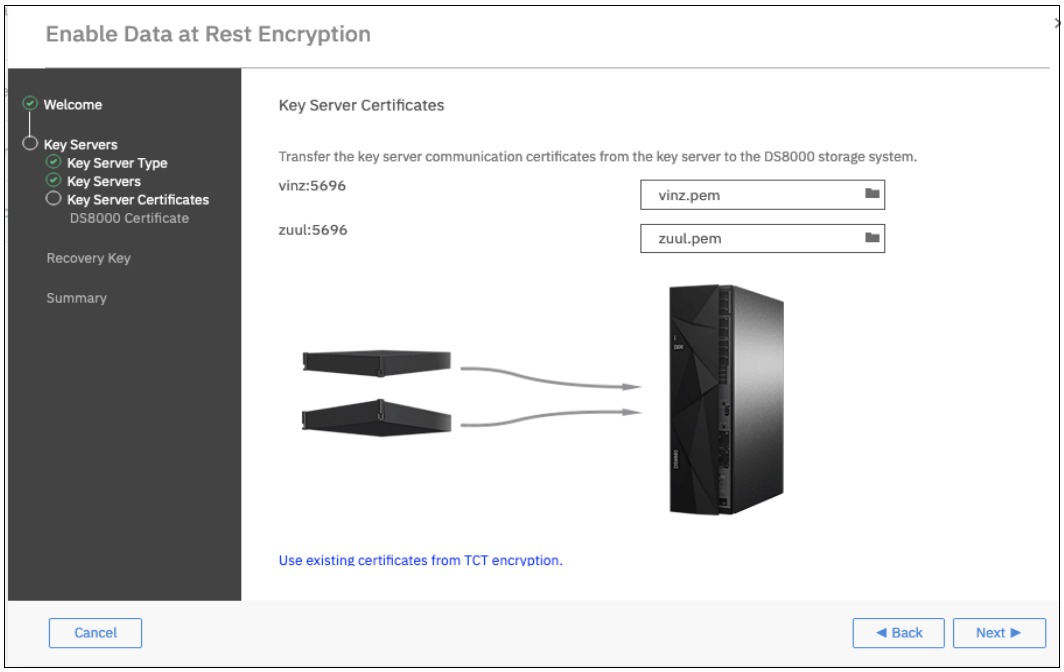


Figure 5-154 Transfer the SSL certificates to a DS8000 server

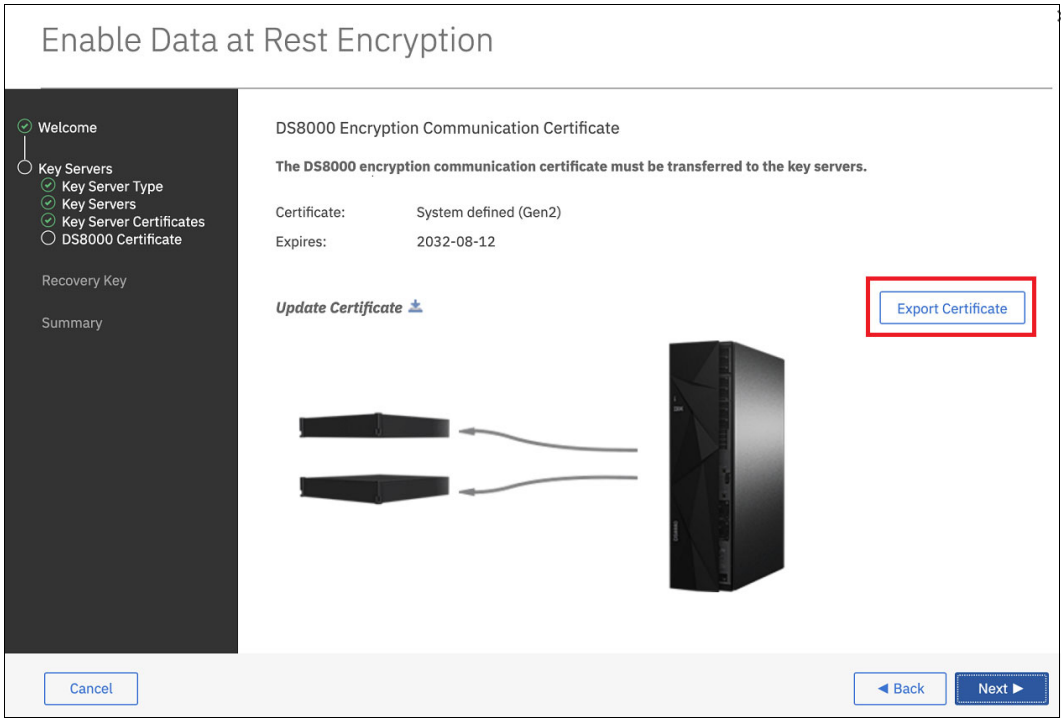


Figure 5-155 Transfer the certificate step 2

Exporting the SSL certificates is described in “Exporting the SSL/KMIP server certificate” on page 100 for IBM Security Key Lifecycle Manager, “Creating a self-signed SSL server certificate” on page 123 got Thales Vormetric DSM, and “Obtaining the KMIP public certificate” on page 146 for Thales CipherTrust Manager.

9. Authorize the pending request for the RK from the Security Administrator (if not already done). Click **Authorize**, as shown in the Figure 5-156.

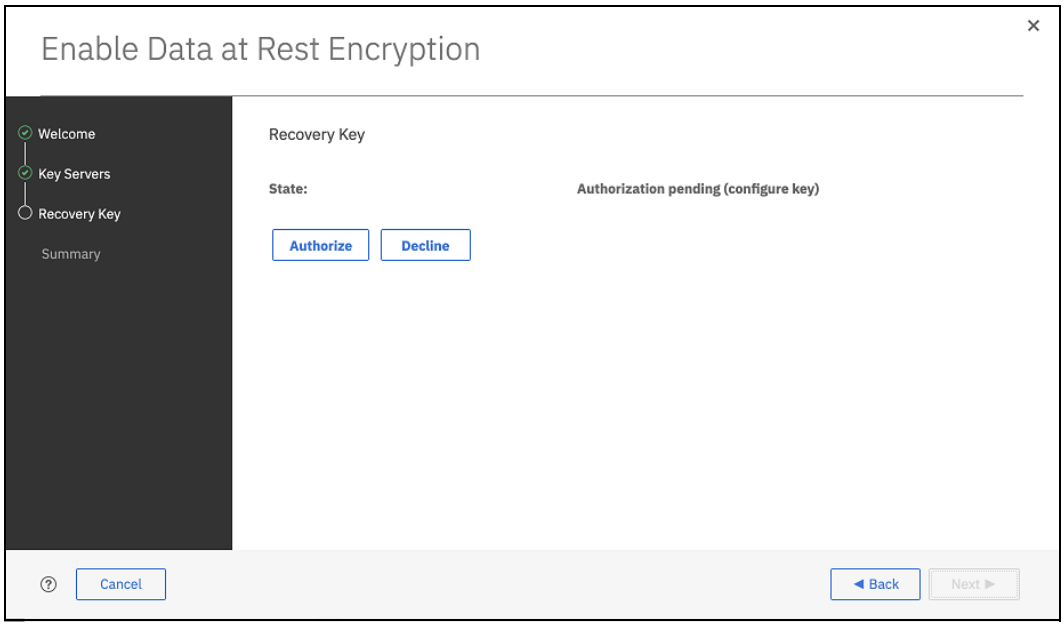


Figure 5-156 Encryption wizard: Authorize the recovery key

10. The confirmation message window opens (see Figure 5-157). Click **Yes** to continue.

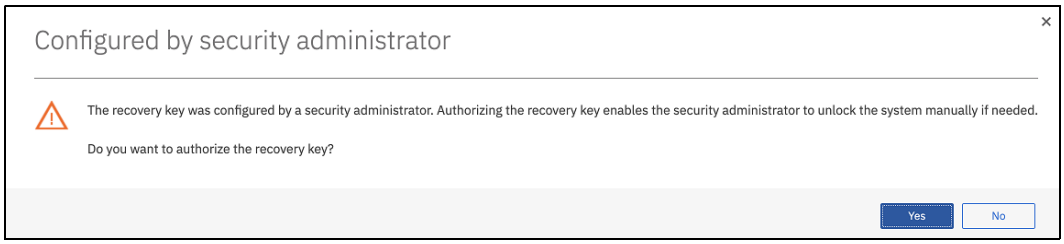


Figure 5-157 Encryption wizard: Confirm recovery key authorization

The RK state changes to Configured when the RK authorization is confirmed (see Figure 5-158). If the RK is disabled, the state shows as disabled.

Figure 5-158 Encryption wizard: Recovery key configured

11. Figure 5-159 shows a summary of the configuration. Click **Finish** to start all of the tasks that are required to enable the encryption.

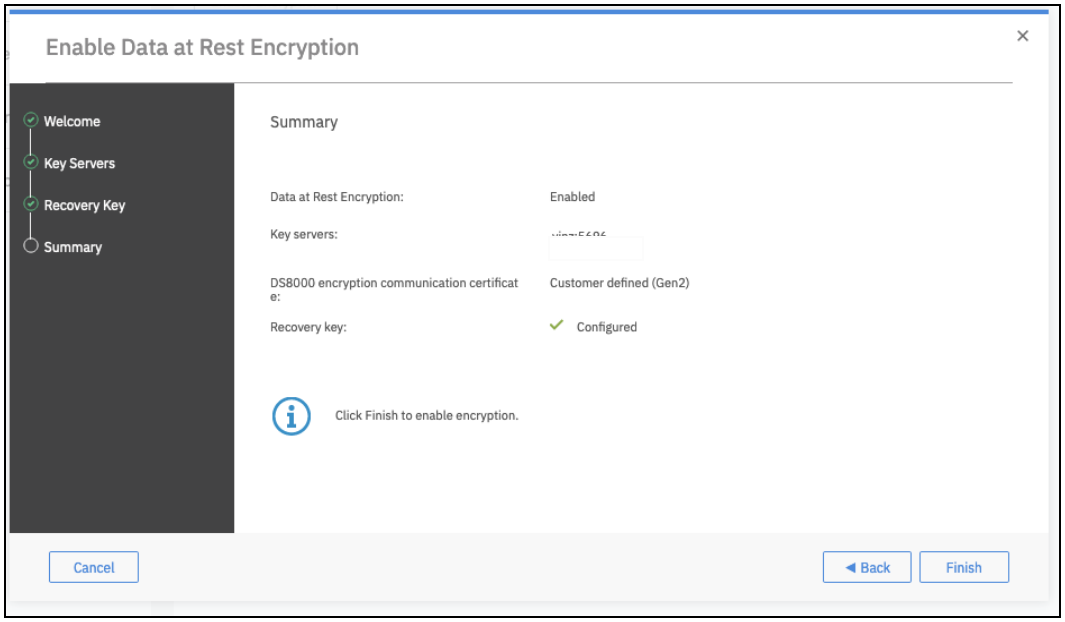


Figure 5-159 Encryption wizard: Summary

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**.

In the Encryption window (see Figure 5-160), the encryption state is Enabled and the encryption key is Accessible. By expanding each section, you get more information. In this example, one key label exists and in this case, two key servers that are accessible and online.

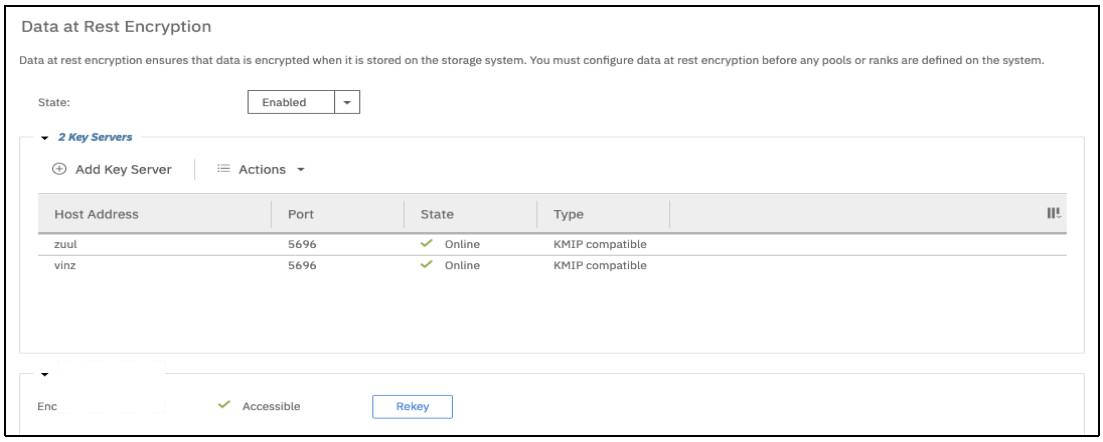


Figure 5-160 Encryption enabled and accessible

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. When 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 DK and RK. For more information, see 6.1, “Rekeying the data key for data at rest encryption” on page 214 and 6.2, “Recovery key use and maintenance” on page 221.

5.5.4 DS8000 CLI configuration for data at rest encryption

You can configure DAR on the DS8000 server by using the DS CLI. The high-level configuration sequence includes the following steps:

1. Enable the RK (for DAR encryption only).
2. Configure the Key Manager server connection to the DS8000.
3. Configure the encryption key group.
4. Apply the activation key.

Configure and administer the encrypted storage pools and assign arrays.For more information about enabling NIST SP 800-131a compliant encryption certificates and TLS 1.2 communication, see 5.9, “NIST SP 800-131a requirements for key servers” on page 194.

Details: For more information about the CLI and commands, see *IBM DS8000 Series Command-Line Interface User's Guide*, SC27-9562.

Enabling the recovery key for data at rest encryption

Note: Creating and managing an RK applies for DAR encryption only.

As described in “Creating the recovery key” on page 161, the risk of a deadlock situation can be substantially minimized by maintaining redundant (dual-platform) IBM Security Key Lifecycle Manager servers for DAR encryption, but it cannot be eliminated.

The RK feature for DAR encryption provides a way to get out of a deadlock state.

You can enable or disable RK management. This choice must be made before you configure any encryption key group.

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

1. The CLI command that is used to configure the RK must be entered by a user with Security Administrator (secadmin) authority. Run the **mkreckey** command, as shown in Example 5-26.

   Example 5-26 Configure the recovery key with the mkreckey command

   ```
   dscli> mkreckey –dev IBM.2107-75xxxx1 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 RK text from the terminal and save it in a file, which can be used for printing. However, this approach 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 RK and storing the paper in a safe place.

2. The secadmin runs the **managereckey -action verify** command to ensure that the written key is correct, as shown in Example 5-27.

   Example 5-27 Verify the recovery key

   ```
   dscli> managereckey –dev IBM.2107-75xxxx1 –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 RK is now in the `Authorization Pending` status.

3. After the RK is verified, the Storage Administrator authorizes usage of the RK 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 5-28.

   Example 5-28 Authorize the recovery key

   ```
   dscli> managereckey –dev IBM.2107-75xxxx1 –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 RK state by running the **lskeygrp** command, as shown in Example 5-29.

Example 5-29 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`, which indicates that a new RK was requested, verified, and authorized. `State` is still unconfigured because the encryption key group is not configured yet.

Configuring the key server connection

The DS8000 supports up to four Key Manager Server connections per encryption key group. However, only one encryption key group (encryption key group 1) is available for DAR encryption.

An intermixing between different sorts of key servers connections is not allowed. KMIP and IBM Proprietary Protocol protocols also cannot be intermixed for DAR encryption.

Having an IBM Proprietary Protocol DAR key group and KMIP for the key group for TCT encryption or IBM Fibre Channel Endpoint Security in parallel is allowed when you use IBM Security Key Lifecycle Manager for the key servers.

The following suggestions apply to configurations per encryption key group:

- 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 configuration describes how to connect the key servers by using these port assignments:

- SSL/TLS port 441 for IBM Security Key Lifecycle Manager
- KMIP port 5696 for IBM Security Key Lifecycle Manager
- Port 5697 for Thales Vormetric DSM
- Port 5697 for Thales CipherTrust Manager

If you must connect the IBM Security Key Lifecycle Manager by using SSL/TLS v1.2 for NIST SP 800-131a compliance, see 5.9, “NIST SP 800-131a requirements for key servers” on page 194. Then, see “Configuring IBM Security Key Lifecycle Manager servers for data at rest encryption” on page 175.

Note: When running **mkkeymgr** to configure IBM Proprietary Protocol key managers, be aware that port 3801 is the default IBM Proprietary Protocol TCP port for IBM Security Key Lifecycle Manager.

IBM Proprietary Protocol communication on the IBM Security Key Lifecycle Manager IBM Proprietary Protocol TCP port is not secured by TLS and it is not recommended because it is not as secure. However, if you choose that option at your own risk instead of TLS, then you must NOT provide key server certificates as part of key server configuration. In your **mkkeymgr** parameters, if you select the port that is being used for IBM Proprietary Protocol TCP communication on IBM Security Key Lifecycle Manager (default is 3801), then you must not provide a certificate location.

If your key server port number is the IBM Proprietary Protocol TCP port of the IBM Security Key Lifecycle Manager and you configured a certificate when creating the key server object, key server communication fails because the IBM Security Key Lifecycle Manager is communicating through TCP and the DS8000 is communicating through TLS. This situation results in errors during encryption enablement.

Figure 5-161 shows the default ports as displayed in the IBM Security Key Lifecycle Manager GUI Welcome window.

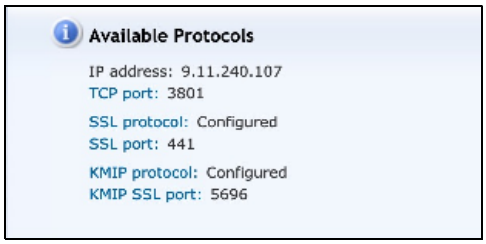


Figure 5-161 Default ports in IBM Security Key Lifecycle Manager

Configuring IBM Security Key Lifecycle Manager servers for data at rest encryption

To configure the IBM Security Key Lifecycle Manager server connection, use the **mkkeymgr**, **mkkeygrp**, and **lskeymgr** commands.

With Release 8.5 and later, your configuration commands must also specify the encryption type **dar** and the encryption key group number. (The defaults are **IPP** protocol, type **DAR** (DAR), and encryption key group **1**.)

Complete the following steps:

1. Run the **mkkeymgr** command at the **dscli** command prompt, with the parameters and variables for IBM Proprietary Protocol and DAR that are shown in Example 5-30.

   Example 5-30 Create one or more key servers with IBM Proprietary Protocol

```
dscli> mkkeymgr -addr 9.123.456.10 -serverport 441 -cert
/home/source/keba_ssl_kmip_2.cer -keyprotocol IPP -type dar -keygrp 1 1
-------------------------------------------------------
CMUC00354I mkkeymgr: The key server 1 has been created.
```

Commands for KMIP are shown in Example 5-31.

Example 5-31 Create one or more key servers with KMIP

```
dscli> mkkeymgr -addr 9.123.456.11 -serverport 5696 -cert
/home/source/keba_ssl_kmip_2.cer -keyprotocol KMIP -type dar -keygrp 1 1
-------------------------------------------------------
CMUC00354I mkkeymgr: The key server 1 has been created.
```

Repeat the **mkkeymgr** command for key servers 2 - 4, if required.

2. 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 with the **-l** parameter, as shown in Example 5-32.

Example 5-32 Verify the IBM Security Key Lifecycle Manager servers

```
dscli> lskeymgr
Date/Time: October 1, 2019 5:07:38 PM CEST IBM DS CLI Version: 7.9.0.509 DS: -
ID  state   status keyprotocol addr         port type keygrp
============================================================
  1 active  normal IPP         9.123.456.10 441  DAR  1
```

3. Create the Key group with IBM Proprietary Protocol (see Example 5-33) or for KMIP (see Example 5-34).

Example 5-33 Create Key Group with IBM Proprietary Protocol

```
mkkeygrp -keyprotocol ipp -label ds8900 -type dar 1
```

Example 5-34 Create Key Group with KMIP

```
mkkeygrp -keyprotocol kmip -type dar 1
```

A DAR encryption key group contains a set of extent pools, each of which has a set of associated ranks and volumes. The remote mirror and copy functions can migrate data within or between encryption key groups.

You can now start creating the extent pools.

You do not need to specify or enable any other parameters to start using the encrypted DS8000 disks. When you select arrays, the encryption status for each array is displayed.

5.5.5 Various authentication mechanisms in IBM Security Key Lifecycle Manager

IBM Security Key Lifecycle Manager have different levels of authentication for incoming requests from IBM Proprietary Protocol and KMIP devices.

Certificate authentication for TLS 1.2 communication

The first level of authentication is done with the validation of certificates. The DS8000 server communicates with IBM Security Key Lifecycle Manager over TLS 1.2. IBM Security Key Lifecycle Manager has the root and intermediate certificates (Gen 2 and Gen 3) of DS8000 certificates in IBM Security Key Lifecycle Manager truststore (Figure 5-162). So, if no DS8000 server is using default certificates, then the certificates are automatically trusted by IBM Security Key Lifecycle Manager.

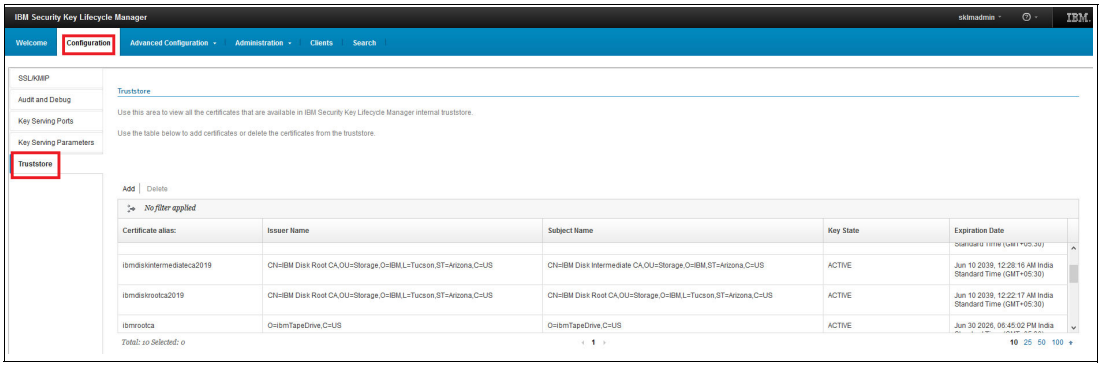


Figure 5-162 IBM Security Key Lifecycle Manager truststore

If the DS8000 server presents the Chain of Trust and the root certificate of that certificate chain is trusted in IBM Security Key Lifecycle Manager, then whole chain is automatically trusted. You do not have to import and trust all the intermediate levels and endpoint certificate in IBM Security Key Lifecycle Manager. IBM Security Key Lifecycle Manager servers do not present the Chain of Trust to the DS8000 devices, so you must ensure that the IBM Security Key Lifecycle Manager server certificate or certificate-signing IBM Security Key Lifecycle Manager server certificate is trusted at the DS8000 endpoint.

Note: The certificate that is described here is the network communication certificate, which should not be confused with the certificate that is created for the DAR or TCT encryption.

Device authentication for IBM Proprietary Protocol

A DS8000 server can communicate with IBM Security Key Lifecycle Manager by using IBM Proprietary Protocol over TLS communication channels. In IBM Security Key Lifecycle Manager, you can configure policy to have step-up authentication.

There are three policy options (Figure 5-163) for new DS8000 devices.

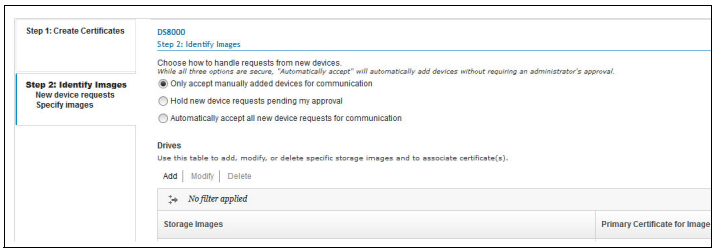


Figure 5-163 IBM Security Key Lifecycle Manager device policies

Here are the descriptions of the policy options:

1. **Only accept manually added devices for communication**

   This option is the default option, and it is the most secure option. When this option is configured, requests from new DS8000 devices are rejected automatically, so an administrator must manually configure the devices. This process is described in 5.5.1, “Setting up IBM Security Key Lifecycle Manager Key management by using IBM Proprietary Protocol” on page 151.

2. **Hold new device requests pending my approval**

   This option is a secure option where requests from new DS8000 devices are added to a pending list in IBM Security Key Lifecycle Manager. An administrator of IBM Security Key Lifecycle Manager can approve or reject these requests from the IBM Security Key Lifecycle Manager UI.

   To approve or reject a pending device, complete the following steps:

   a. Log in to the IBM Security Key Lifecycle Manager UI. You see the **Pending device** link in the Action Items section of Welcome page (Figure 5-164).

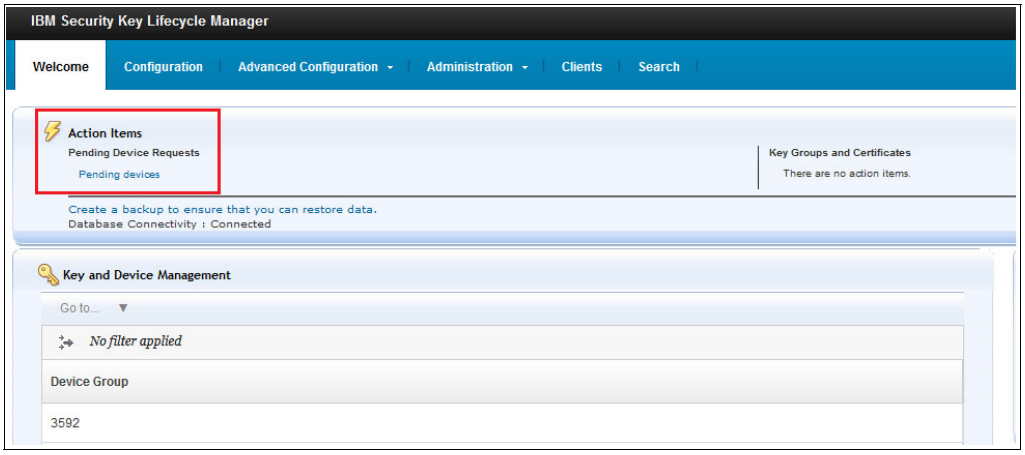


Figure 5-164 Pending Devices link on the Welcome page

   b. Click **Pending devices** to see list of pending devices. Select any device and click **Accept** to approve the pending device (Figure 5-165 on page 179).

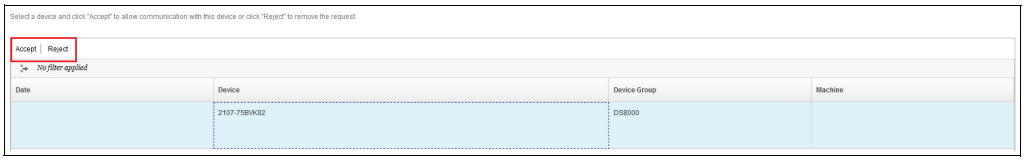


Figure 5-165 Pending devices

c. Click **Accept** to approve the device (Figure 5-166).

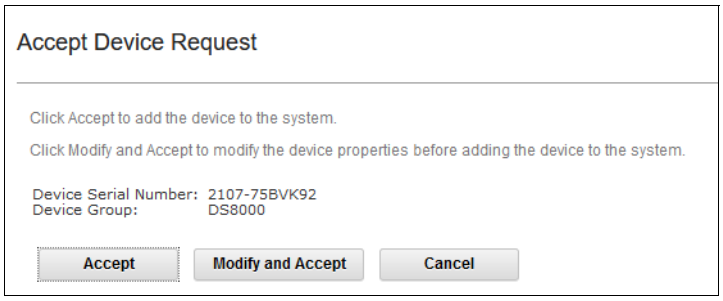


Figure 5-166 Accepting pending devices

3. **Automatically accept all new device requests for communication**

   This option is the least secure option. When this option is configured, a new device request is approved automatically.

Certificates in Transparent Cloud Tiering

Even when certificates are trusted in IBM Security Key Lifecycle Manager truststore, they will not be served keys until they are added to the `DS8000_TCT` device group. After certificates are added to the `DS8000_TCT` device group, any of the endpoints that are associated with those certificates can access any keys that are created in the `DS8000_TCT` device group. For more information, see 5.6, “Configuration for TCT encryption” on page 179.

Worldwide node name authentication in IBM Fibre Channel Endpoint Security

IBM Security Key Lifecycle Manager composes the names of the peer-to-peer device groups from the worldwide node names (WWNNs) of the peers (endpoint devices) in the group. The device group contains the security credentials in the form of a WWNN in the Subject Alternative Name of the certificates that the endpoints use to communicate with IBM Security Key Lifecycle Manager. For more information, see 3.6, “DS8000 endpoint encryption key management using KMIP” on page 56.

5.6 Configuration for TCT encryption

IBM Security Lifecycle Manager V3.0.0.2 or later can encrypt data objects that are transferred to the cloud by TCT and by using the KMIP.

To use TCT encryption, you must set up the IBM Security Key Lifecycle Manager with a DS8000 device. After the device is configured, you can enable TCT encryption by using a DS CLI command at any time. You run this command independently from DAR encryption and without affecting access to the data.

Note: Enablement of TCT encryption in the DS8000 GUI is not supported at the time of this writing.

TCT encryption includes the following prerequisites:

- The DS8880 must be upgraded to Release 8.5 or later
- IBM Security Lifecycle Manager must be upgraded or reinstalled to Version 3.0.0.2 or later
- IBM Security Lifecycle Manager must run in a Multi-Master environment

Both Thales Vormetric DSM and Thales CipherTrust Manager are supported by TCT encryption.

5.6.1 Setting up TCT encryption

The following process is used to set up TCT encryption:

1. Export the certificate from the DS8000.
2. Transfer certificate to the key server.
3. Import the DS8000 certificate into the key server TCT device group.
4. Configure DS8000 TCT by using CLI.

Exporting the DS8000 Certificate

Most often when you set up and use TCT encryption, it is not necessary to have DAR encryption enabled. However, the Gen 2 or Gen 3 certificate can be exported through the DS GUI only when DAR encryption is enabled, or while you are using the DAR encryption activation wizard. Consider the following points:

- If DAR encryption is enabled, log in to the DS8000 GUI and export the DS8000 Encryption Communication Certificate by using one of the following options:
  - Use DS CLI, as shown in Example 5-35.

Example 5-35 Export certificate with CLI

```
dscli> managekeygrp -action exportcert -certType GEN3 -loc
C:\Users\xxxxx\ds8k_75xxxx1_gen3_cert.pem 1
Date/Time: September 30, 2019 5:23:36 PM CEST IBM DS CLI Version: 7.9.0.491 DS:
IBM.2107-75xxxx1
CMUC00490I managekeygrp: The certificate for key group 1 has been exported.
```

– If DAR encryption is enabled, select **Settings** → **Security** → **Data at rest encryption** and click **Export Certificate**, as shown in Figure 5-167.

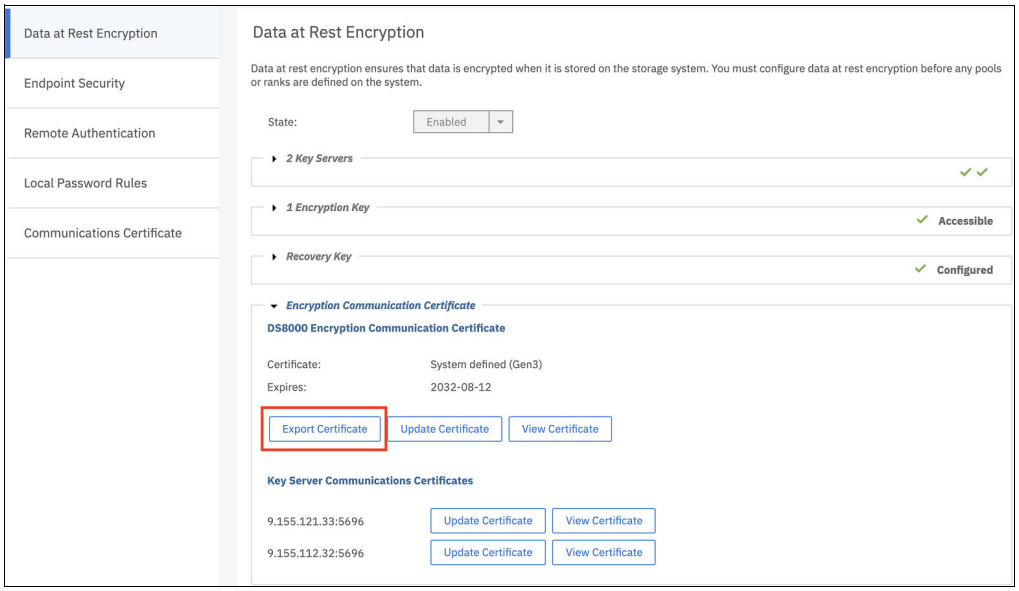


Figure 5-167 Export certificate

► If DAR encryption is not enabled and no logical configuration exists, log in to the DS8000 GUI and select **Settings** → **Security** → **Data at rest encryption**. Then, follow the wizard after you select **Enable**", as described in "DS8000 enabling data at rest encryption" on page 163 to enable DAR encryption. Continue until you reach the DS8000 Certificate step as shown in Figure 5-168. Export it by clicking the **Export Certificate**.

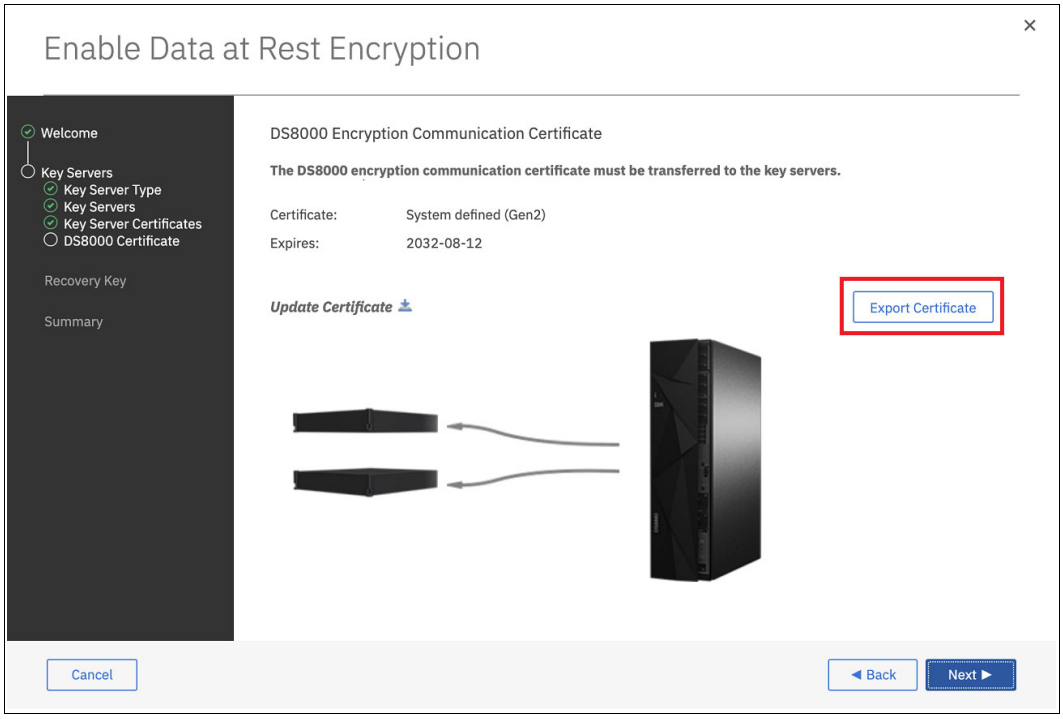


Figure 5-168 Export Certificate

- If DAR encryption is not enabled and a logical configuration exists, you can export the certificate by using the CLI, as shown in Example 5-35 on page 180. Even if no key manager and keygroup were configured, you can export the certificate.

Transferring the certificate to the key server

Then, you must transfer the file to the primary key manager server. Example 5-36 shows running the **scp** command to transfer the certificate file and shows the target directory `/opt/IBM/WebSphere/AppServer/products/sklm/data` on IBM Security Key Lifecycle Manager where the file needs to reside.

Example 5-36 Certificate transfer to IBM Security Key Lifecycle Manager

```
[root@sklma source]# scp ds8k_75xxxx1_gen3_cert.pem
root@0.0.0.1:/opt/IBM/WebSphere/AppServer/products/sklm/data/

ds8k_75BRX70_gen2_cert.pem 100% 1270     1.2KB/s   00:00
[root@sklma source]#
```

DS8000 TCT encryption certificate (Gen 2) import (IBM Security Key Lifecycle Manager)

The exported certificate can now be imported to the IBM Security Key Lifecycle Manager Primary Master server. Complete the following steps to finish the IBM Security Key Lifecycle Manager configuration:

1. Log in to the IBM Security Key Lifecycle Manager GUI and locate the device group “DS8000_TCT” in “Key and Device Management” on the left side, as shown in Figure 5-169.

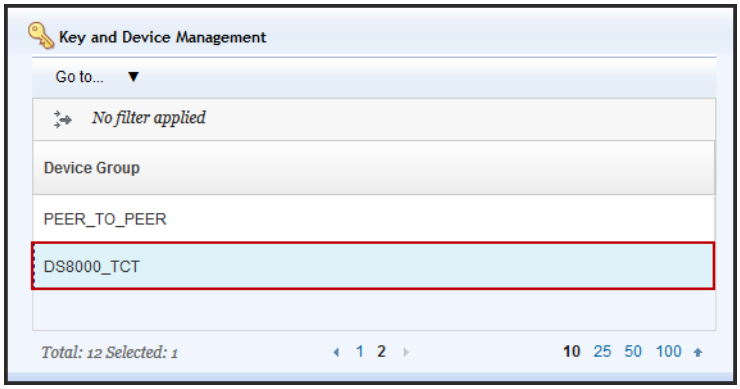


Figure 5-169 “DS8000_TCT” in “Key and Device Management”

2. Right-click the group **DS8000_TCT** and select **Manage keys and devices**, as shown in Figure 5-170.

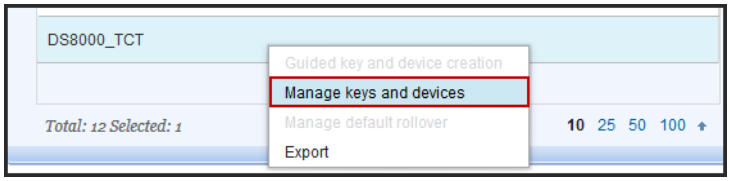


Figure 5-170 Manage keys and devices

3. The window that is shown in Figure 5-171 allows you to add or delete a certificate and the associated node name. You can also modify the node name that is associated with a certificate. Click **Add**" and select **Certificate** to import the DS8000 Encryption Communication Certificate (Gen 2).

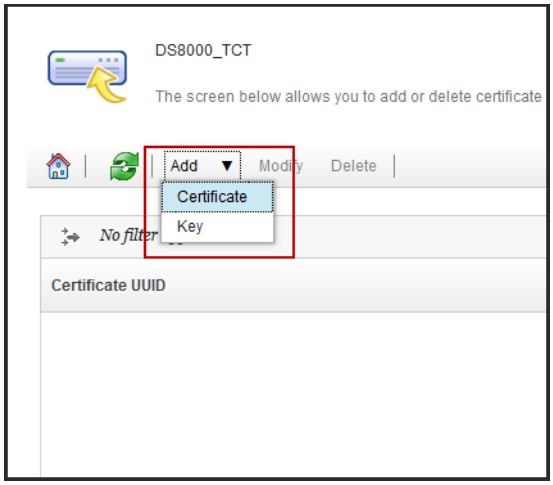


Figure 5-171 Add Certificate

4. In the Add Certificate window, enter a unique certificate alias name. Then, browse for the certificate file that was transferred in "Exporting the DS8000 Certificate" on page 180, as shown in Figure 5-172.

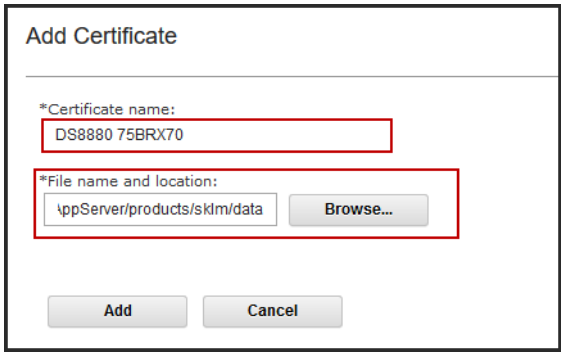


Figure 5-172 Certificate alias

5. If you did not use the default directory `/opt/IBM/WebSphere/AppServer/products/sklm/data`, browse and select the directory where the certificate file that was transferred in an earlier step.

Then, highlight the certificate file, and click **Select**, as shown in Figure 5-173.

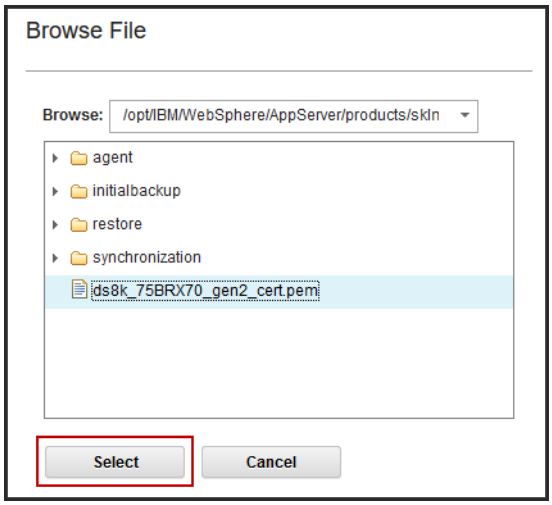


Figure 5-173 Browse file

6. Select **Add** in the Add Certificate window, as shown in Figure 5-174.

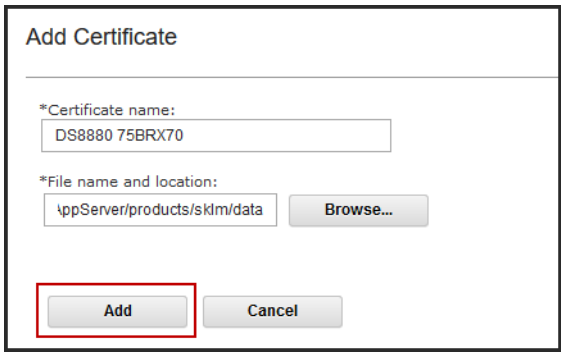


Figure 5-174 Add Certificate

7. A warning to create a backup is displayed. Click **Close** (see Figure 5-175) and create a backup now or later. For more information, see “Creating a backup” on page 102.

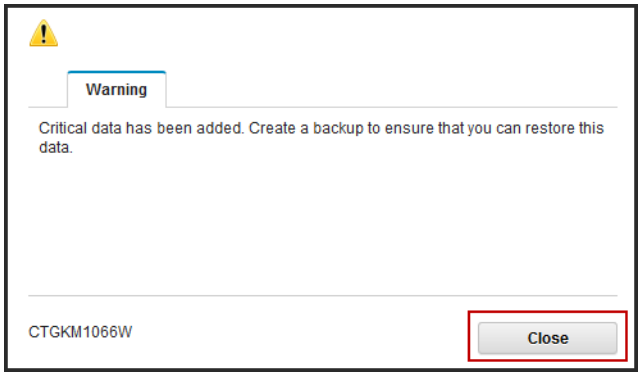


Figure 5-175 Warning - take a backup

8. Confirm that the certificate was imported successfully and appears in the list, as shown in Figure 5-176.

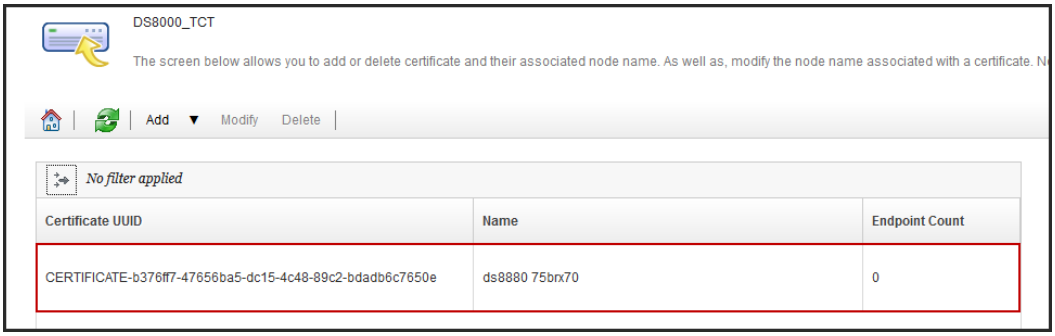


Figure 5-176 Certificate list

The key servers are now configured and synchronized. For Multi-Master setup, the synchronization is done automatically. To create the IBM Security Key Lifecycle Manager server connection for TCT, proceed to “DS8000 CLI configuration for TCT”.

DS8000 CLI configuration for TCT

To configure the key servers with KMIP for TCT encryption, run the **lskeymgr**, **mkkeymgr**, and **managekeymgr** commands, as shown in the following steps:

1. To view the list of registered key servers, if any exist, run the **lskeymgr** command (Example 5-37).

Example 5-37 List the key managers

```
dscli> lskeymgr
Date/Time: October 15, 2019 8:12:26 AM CEST IBM DS CLI Version: 7.9.0.512 DS: -
ID state   status keyprotocol addr         port type keygrp
==========================================================
 1 active  normal KMIP        9.155.121.33 5696 DAR  1
 2 active  normal KMIP        9.155.112.32 5696 DAR  1
```

The preceding example shows two key managers that are configured in the DS8000 for DAR encryption by using the KMIP. If those servers are prepared to serve keys for both types DAR encryption and TCT encryption), the connection can be modified by using the **managekeymgr** command in Step 1. If the key server cannot be used for TCT encryption, see “Creating a TCT key server” on page 186 to create key servers.

Adding a TCT key group to a KMIP key server

Complete the following steps:

1. Run the **managekeymgr** command to modify the key servers and add encryption key group 2 for TCT, as shown in Example 5-38.

Example 5-38 Manage existing key managers

```
dscli> managekeymgr -action addgrp -keygrp 2 -type tct 1
CMUC00563I managekeymgr: The key group was added to the key manager successfully.

dscli> managekeymgr -action addgrp -keygrp 2 -type tct 2
CMUC00563I managekeymgr: The key group was added to the key manager successfully.
```

2. Verify the extra connections by using the **lskeymgr** command, as shown in Example 5-39. Both key servers show now `type = DAR,TCT` and `keygrp = 1,2`.

Example 5-39 List key managers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type    keygrp
==========================================================
  1 active  normal KMIP        0.000.000.01 5696 DAR,TCT 1,2
  2 active  normal KMIP        0.000.000.02 5696 DAR,TCT 1,2
```

3. Configure encryption key groups for TCT encryption, as shown in Example 5-40.

Example 5-40 Create encryption key group

```
dscli> mkkeygrp -keyprotocol kmip -type tct 2
CMUC00358I mkkeygrp: The key server key group 2 has been created.
```

4. List the encryption key groups, as shown in Example 5-41.

Example 5-41 List encryption key group

```
dscli> lskeygrp
ID  state      reckeystate reckeydate datakeydate keyprotocol type name
=====================================================================
1   accessible configured  10/01/2019 10/02/2019  KMIP        DAR  DAR_1
2   accessible disabled    -          10/02/2019  KMIP        TCT  TCT_2
```

A TCT encryption key group contains a set of cloud server connections.

You can now configure the cloud server connection for TCT with the encryption parameters. For more information, see *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381.

Creating a TCT key server

Complete the following steps:

1. Run the **lskeymgr** command (see Example 5-42) to view the list of registered key servers, if any.

Example 5-42 List key managers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type keygrp
=========================================================
  1 active  normal IPP         0.000.000.01 441  DAR  1
  2 active  normal IPP         0.000.000.02 441  DAR  1
```

The preceding example shows two key managers that are configured in the DS8000 for DAR encryption by using the IBM Proprietary Protocol.

Key managers that are configured with IBM Proprietary Protocol cannot be used for TCT encryption. The encryption type **tct** is not supported by IBM Proprietary Protocol and TCT encryption requires the KMIP protocol. You also cannot have IBM Proprietary Protocol and KMIP protocols on the same key server.

Therefore, key servers must be created.

To create key servers, use the **mkkeymgr** command, as shown in Example 5-43.

You must specify the following values:

- Server certificate (**-cert**) that was exported in “Exporting the SSL/KMIP server certificate” on page 100
- Protocol (**-keyprotocol**) **KMIP**
- Servers IP address or hostname (**-addr**)
- Encryption type (**-type**) **tct**
- Encryption key group (**keygrp**) **2**
- New key server ID

Example 5-43 Creating key servers

```
dscli> mkkeymgr -cert
/opt/IBM/WebSphere/AppServer/products/sklm/data/ssl_kmip_server_cert.cer
-keyprotocol KMIP -addr 0.000.000.10 -type tct -keygrp 2 10
CMUC00354I mkkeymgr: The key server 10 has been created.

dscli> mkkeymgr -cert
/opt/IBM/WebSphere/AppServer/products/sklm/data/ssl_kmip_server_cert.cer
-keyprotocol KMIP -addr 0.000.000.11 -type tct -keygrp 2 11
CMUC00354I mkkeymgr: The key server 11 has been created.
```

Note: The **-cert** parameter specifies the location of the certificate file that was exported earlier. This certificate is used as a trust anchor to authenticate the certificate of the specified key server when a TLS security protocol is used. If the parameter is not specified, 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 you use KMIP.

2. Verify the new connections by using the **lskeymgr** command, as shown in Example 5-44. The key servers 1 and 2 that are use IBM Proprietary Protocol by way of port 441 still show **type = DAR** and **keygrp = 1**. However, the new key servers 10 and 11 that use KMIP by way of port 5696 show **type = TCT** and **keygrp = 2**.

Example 5-44 List all key servers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type keygrp
==========================================================
  1 active  normal IPP         0.000.000.01 441  DAR  1
  2 active  normal IPP         0.000.000.02 441  DAR  1
 10 active  normal KMIP        0.000.000.10 5696 TCT  2
 11 active  normal KMIP        0.000.000.11 5696 TCT  2
```

3. Configure encryption key groups for TCT encryption, as shown in Example 5-45.

Example 5-45 Create encryption key group

```
dscli> mkkeygrp -keyprotocol kmip -type tct 2
CMUC00358I mkkeygrp: The key server key group 2 has been created.
```

4. List encryption key groups, as shown in Example 5-46.

Example 5-46 List encryption key group

```
dscli> lskeygrp
ID  state      reckeystate reckeydate datakeydate keyprotocol type name
=====================================================================
1   accessible configured  10/01/2019 10/02/2019  DAR         DAR  DAR_1
2   accessible disabled    -          10/02/2019  KMIP        TCT  TCT_2
```

A TCT encryption key group contains a set of cloud server connections.

You can now configure the cloud server connection for TCT with the encryption parameters. For more information, see *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381.

5.7 Configuration for IBM Fibre Channel Endpoint Security

This section provides a brief overview of the IBM Fibre Channel Endpoint Security encryption setup on the target DS8000 (DS8900F).

Tip: For more information about the setup that is required on the Z Central Processor Complex (CPC) (initiator), see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

5.7.1 DS8000 GUI configuration for IBM Fibre Channel Endpoint Security

Enabling IBM Fibre Channel Endpoint Security with the IBM Security Key Lifecycle Manager by using KMIP requires a Multi-Master configuration of the key server. The key servers do not require any more configuration to serve keys to the DS8900F, assuming that you exported the DS8000 certificates to the key servers.

To configure the DS8000, complete the following steps:

1. Log on to the DS8000 GUI as a user with Administrator role to enable the IBM Fibre Channel Endpoint Security. From the DS8000 GUI Welcome window, click **Settings** and then, **Security**, as shown in Figure 5-177.

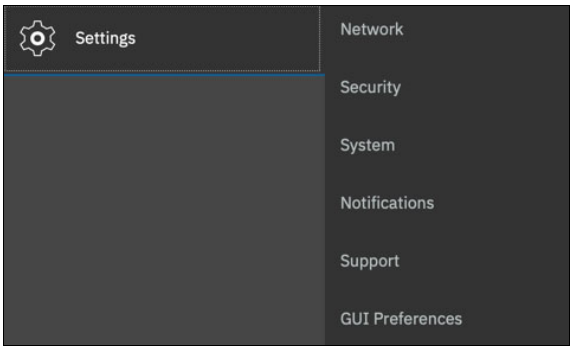


Figure 5-177 Go to the Encryption window

2. The encryption wizard is shown in Figure 5-178. This wizard is started only when you enable the IBM Fibre Channel Endpoint Security for the first time. Click **Configure Endpoint Security** to continue.

Figure 5-178 Encryption wizard: Configure Endpoint Security

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; that is, connected to the DS8000 (see Figure 5-179). Click **Next** to continue.

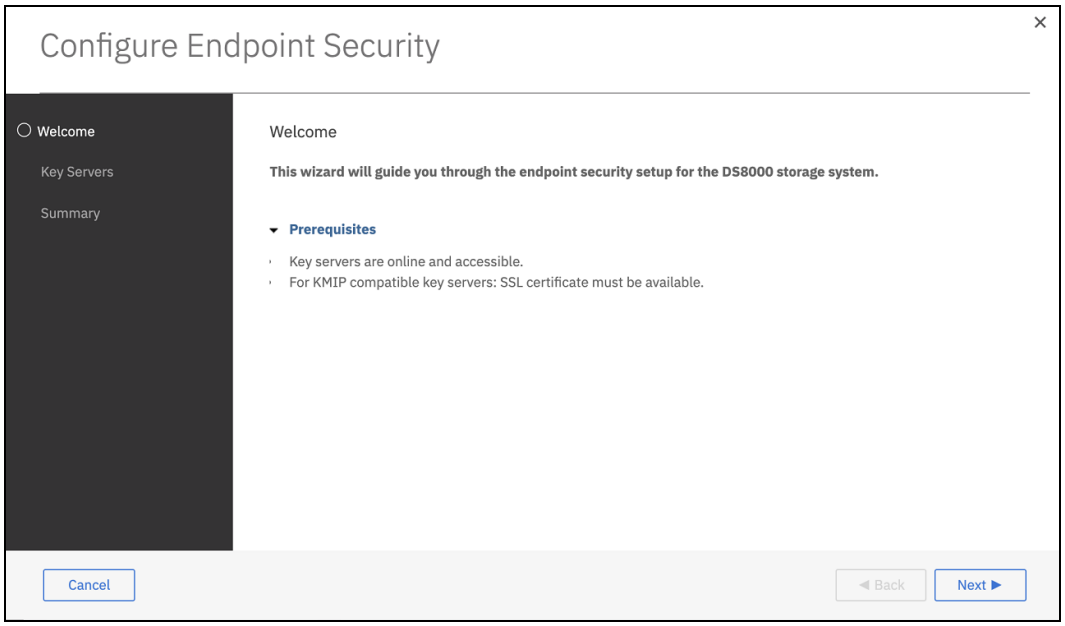


Figure 5-179 Encryption wizard: Welcome window

4. Configure the key servers. The DS8000 supports up to four key servers. In Figure 5-180, two key servers are defined. You can add up to four or remove key servers by clicking the + or - sign next to each key server field. Specify the host address (IP address or fully qualified hostname of the key server). Click **Next**.

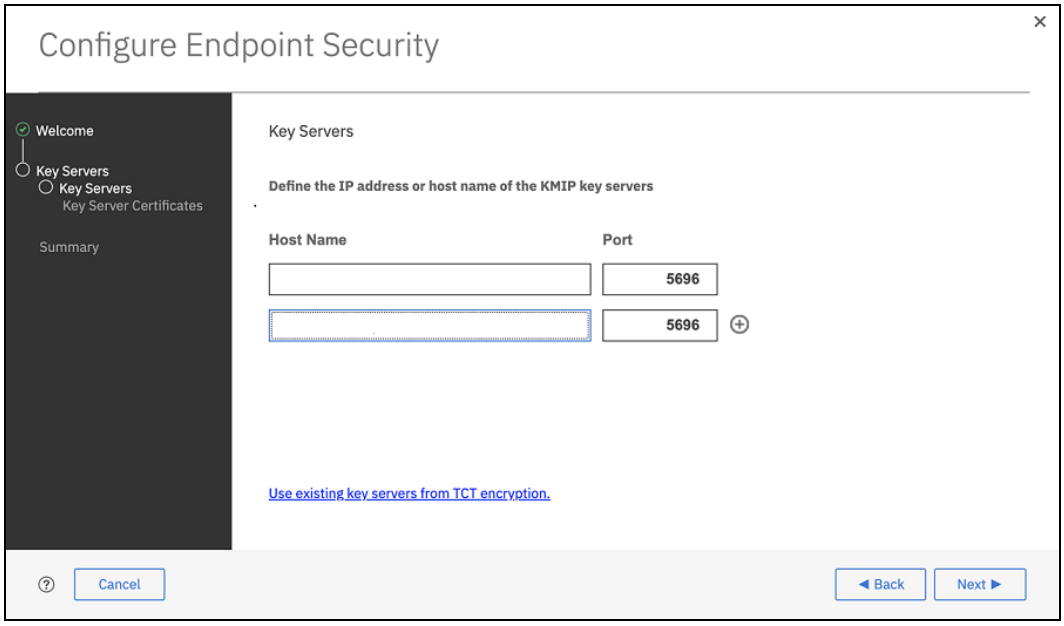


Figure 5-180 Encryption wizard: Define key servers

5. Each key server connection is tested. The message that is shown in Figure 5-181 is displayed if all the key servers that you defined 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 5696.

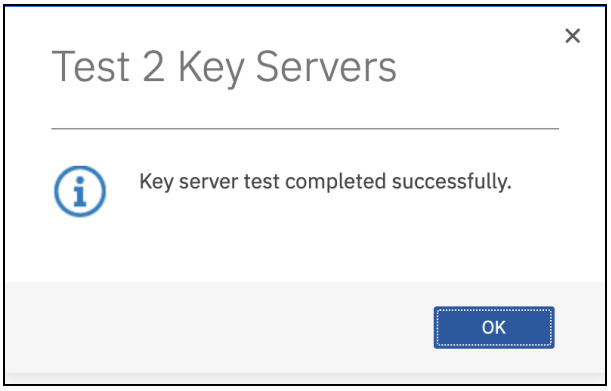


Figure 5-181 Encryption wizard: Test Key Servers

6. Transfer the SSL certificates from the key servers to the DS8000, as shown in Figure 5-182. Exporting the SSL certificates is described in “Exporting the SSL/KMIP server certificate” on page 100 for IBM Security Key Lifecycle Manager. If DAR or TCT encryption is set up, select **Use existing certificates from TCT encryption** at the bottom of the window.

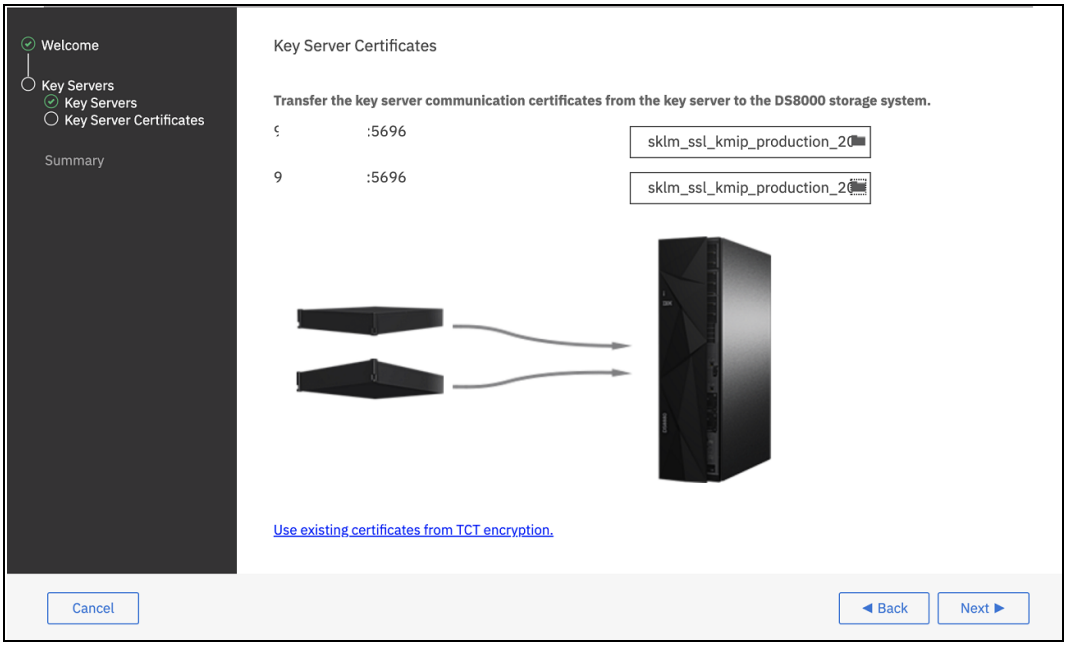


Figure 5-182 Transfer SSL certificates to DS8000

7. 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**.

In the Encryption window (see Figure 5-183), the encryption state is Enabled and the encryption key is Accessible. By expanding each section, you get more information.

The overall process to enable encryption on a DS8000 by using the GUI interface is simple. It takes approximately 5 minutes to complete these steps.

Now, configure the IBM Z. For more information, see *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455.

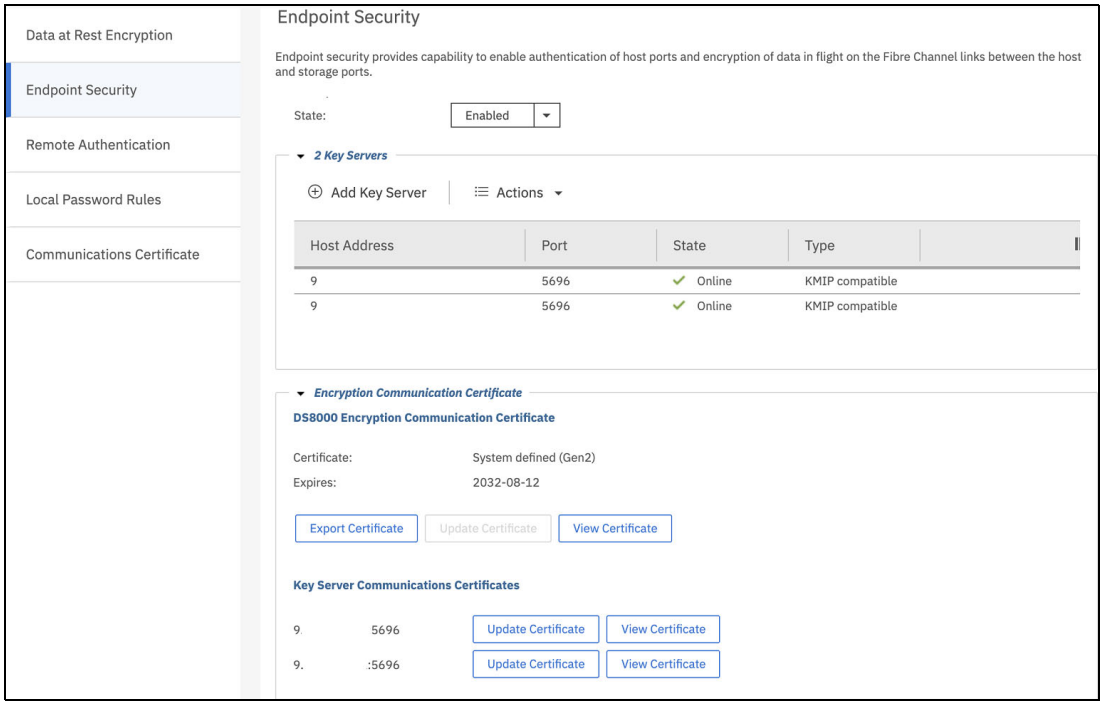


Figure 5-183 Encryption that is enabled and accessible

5.7.2 DS8000 CLI configuration for IBM Fibre Channel Endpoint Security

If you have a KMIP key group configured for DAR or TCT, you can add a key group for IBM Fibre Channel Endpoint Security for the configured key servers. Complete the following steps:

1. Check for current key managers, as shown in Example 5-47.

Example 5-47 List key managers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type keygrp
============================================================
10 active  normal KMIP        0.000.000.01 5696 DAR  1
11 active  normal KMIP        0.000.000.02 5696 DAR  1
```

2. Add Endpoint key group 2 to each key manager, which is identified by ID 10 and 11, as shown in Example 5-48.

Example 5-48 Add key group 2

```
dscli> managekeymgr -action addgrp -keygrp 2 -type endpoint 10
CMUC00563I managekeymgr: The key group was added to the key manager
successfully.
dscli> managekeymgr -action addgrp -keygrp 2 -type endpoint 11
```

```
CMUC00563I managekeymgr: The key group was added to the key manager
successfully.
```

3. Create key group 2, as shown in Example 5-49.

Example 5-49 Create IBM Fibre Channel Endpoint Security key group

```
dscli> mkkeygrp -keyprotocol kmip -type endpoint 2
CMUC00358I mkkeygrp: The key server key group 2 has been created.
```

4. Verify the key managers and see that Endpoint was added, as shown in Example 5-50.

Example 5-50 Verify key managers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type         keygrp
=================================================================
 10 active  normal KMIP        0.000.000.01 5696 DAR,ENDPOINT 1,2
 11 active  normal KMIP        0.000.000.02 5696 DAR,ENDPOINT 1,2
```

If no KMIP key manager exists, create a key manager and key group by completing the following steps:

1. Create the key managers, as shown in Example 5-51.

Example 5-51 Create endpoint key managers

```
dscli> mkkeymgr -cert C:\Users\xxxx\sklm_ssl_kmip_production_2019.cer
-keyprotocol KMIP -addr 0.000.000.01 -type endpoint -keygrp 2 10
CMUC00354I mkkeymgr: The key server 10 has been created.
dscli> mkkeymgr -cert C:\Users\xxxx\sklm_ssl_kmip_production_2019.cer
-keyprotocol KMIP -addr 0.000.000.02 -type endpoint -keygrp 2 11
CMUC00354I mkkeymgr: The key server 11 has been created.
```

2. Create the key group, as shown in Example 5-52.

Example 5-52 Create key group

```
dscli> mkkeygrp -keyprotocol kmip -type endpoint 2
CMUC00358I mkkeygrp: The key server key group 2 has been created.
```

3. Verify the key managers, as shown in Example 5-53.

Example 5-53 Verify key managers

```
dscli> lskeymgr
ID  state   status keyprotocol addr         port type     keygrp
=================================================================
 10 active  normal KMIP        0.000.000.01 5696 ENDPOINT  2
 11 active  normal KMIP        0.000.000.02 5696 ENDPOINT  2
```

5.8 Data at rest encryption and Copy Services functions

Copy Services operations are not affected by encrypting drives. The encryption applies only to DAR, 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 IBM 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 re-encrypted at *write*. The encryption is not intrusive in terms of performance because it is all done by the drives.

5.9 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 key group or RK is configured on the DS8000, the HMC initiates connections to the key servers to request key services and periodically verifies that any active DKs are valid on all configured key servers.

Encryption key servers use a secure connection with the HMC with IBM Proprietary Protocol. However, TLS 1.2 can also be enabled. IBM Security Key Lifecycle Manager V2.6 and later 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 DK 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 IBM Proprietary Protocol. The protocol uses a digital certificate to authenticate the key client with the key server, and it has data security for the DKs 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.

5.9.1 Configuring IBM Security Key Lifecycle Manager to use TLS 1.2

IBM DS8000 R8.1 and later do not support Gen 1 certificates and comes with TLS enabled by default. However, the IBM Security Key Lifecycle Manager V3.0 key servers and DS8000 up to Release 8.0 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=TLS1.2
   ```

   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 V3.0, 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 5-184.

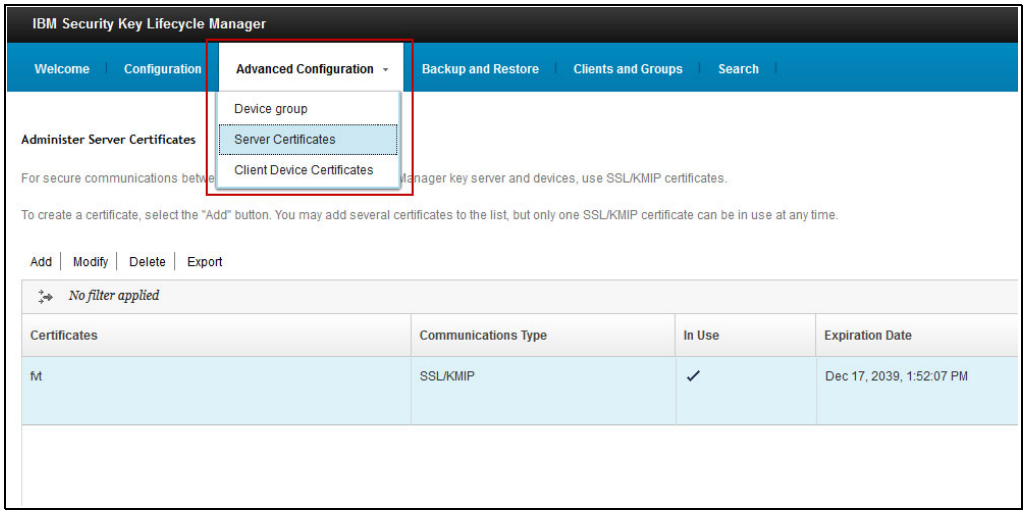


Figure 5-184 Server certificates

b. Right-click the certificate and select **Export**, as shown in Figure 5-185.

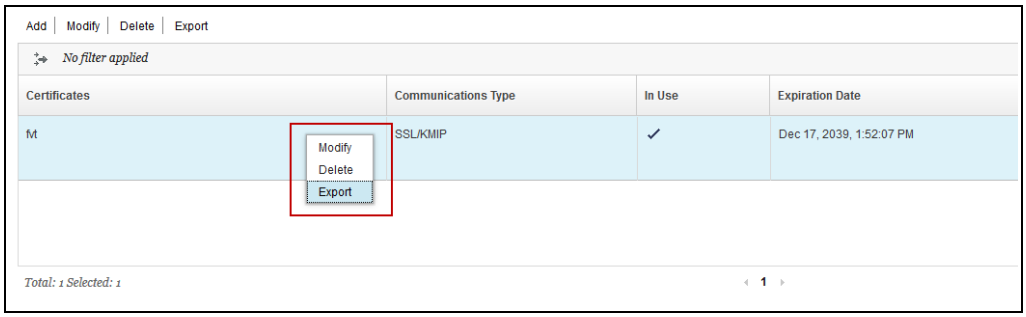


Figure 5-185 Select Export

c. Rename the certificate, if required, and export it to the file location that you want, as shown in Figure 5-186.

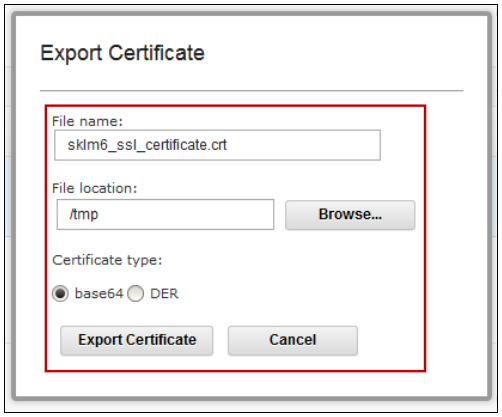


Figure 5-186 Rename and export

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 5-54.

Example 5-54 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 5-54 on page 196 shows the default username and password for the IBM Security Key Lifecycle Manager. Change them to the username 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 DS CLI with Storage Administrator authority. If the key servers already are defined to the DS8000, continue with this step. If no key servers are defined, skip to Step 6 on page 198 to install the new certificate that was exported from the IBM Security Key Lifecycle Manager in step 4.

Example 5-55 shows how to remove one of the key servers. Perform the tasks in this step 5 and step 6 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 5-55 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 5-56 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 197, 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 197. 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 197. You must customize these commands to match your environment. This example is from an IBM configuration that was created for providing example output.

Example 5-56 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 TCP 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.

5.10 Migrating certificates

Note: A Gen 2 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. Gen 2 and Gen 3 certificates are delivered with all DS8900F; although the Gen 2 is active, the Gen 3 is dormant.

5.10.1 Migrating from a Gen 1 to a Gen 2 certificate for encryption

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 5-57 shows how to verify which certificate (Gen 1 or Gen 2) that the DS8000 is using.

   Example 5-57 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 5-58, to change from the Gen 1 to Gen 2 certificate for data encryption on the DS8000 with Release 7.2 microcode, Licensed Managed Code (LMC) 7.7.20.xx. No previous models of the DS8000 series support the Gen 2 certificate.

   Example 5-58 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 key group 1 has been
   updated

   If you have a 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 5-59).

Example 5-59 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.

5.10.2 Migrating from a Gen 2 to a Gen 3 certificate for encryption

Important: To use the Gen 3 certificate on the DS8000, you must enable DAR. IBM Security Key Lifecycle Manager is the only supported key manager.

To migrate from Gen 2 to Gen 3 certificate, complete the following steps:

1. Verify the version of the certificate that is being used for data encryption. Example 5-60 shows how to verify which certificate (Gen 2 or Gen 3) that the DS8000 is using.

Example 5-60 Determine which certificate the DS8000 is using for data encryption

```
dscli> showkeygrp 1
ID                           1
numranks                     4
numpools                     4
state                        accessible
reckeystate                  configured
reckeydate                   10/10/2019 14:48:09 CEST
datakeydate                  10/15/2019 11:45:50 CEST
grpstatus                    normal
mgrstatus                    normal
label                        -
label2                       -
certificate                  GEN2
certificate not valid before 08/22/2019 21:55:02 CEST
certificate not valid after  08/12/2032 02:36:55 CEST
certificate issuer           O=ibmDisk,C=US
certificate subject
CN=2107-75KMW81,O=ibmDisk,C=US,uid=DS8K-2107-75KMW81
endpoint enabled             Yes
certificate install state    Not Imported
```

```
pending cert valid not before -
pending cert valid not after  -
pending cert issuer           -
pending cert subject          -
pending cert endpoint enabled -
uuid                          KEY-4d74e7f-4cbabdc7-2d25-465f-b4c5-d065c2e8a097
keyprotocol                   KMIP
type                          DAR
name
DAR_1
```

2. Export the Gen 3 certificate from the DS8900F by using GUI or CLI:
   – Use DS CLI, as shown in Example 5-61.

   Example 5-61 Export certificate with CLI

   ```
   dscli> managekeygrp -action exportcert -certType GEN3 -loc CMUC00490I managekeygrp:
   The certificate for key group 1 has been exported.
   ```

   – Select **Settings** → **Security** → **Data at rest encryption** and click **Export Certificate**, as shown in Figure 5-187.

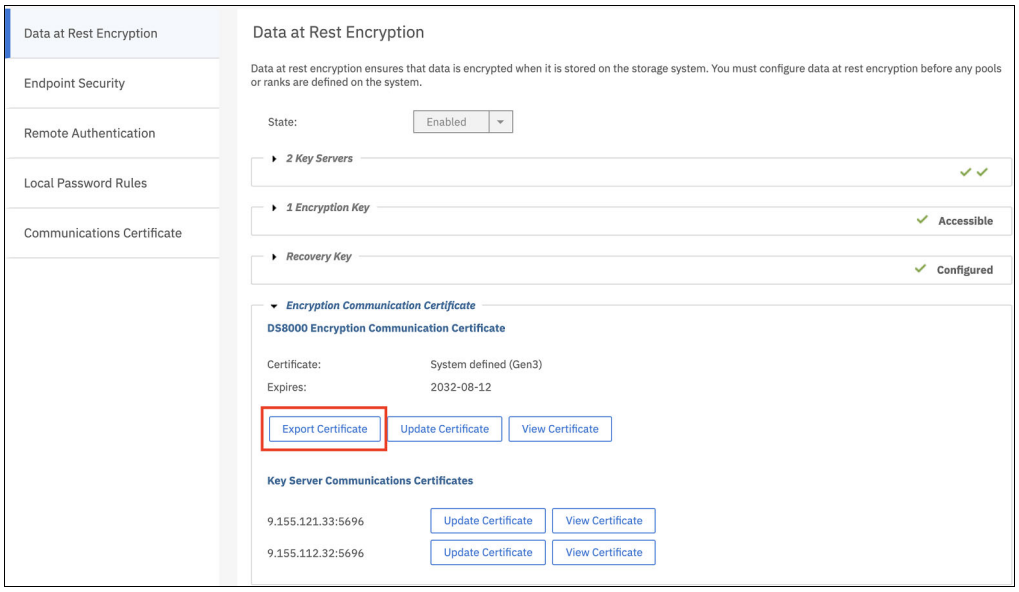


Figure 5-187 Export certificate

3. Transfer the file to the IBM Security Key Lifecycle Manager server.

> **Note:** You need a method to transfer the certificate file that you exported from the DS8900F to the file system of the server the IBM Security Key Lifecycle Manager runs on. You also need a UID that has sufficient access to add a file to the IBM Security Key Lifecycle Manager import directory. On UNIX systems, this directory defaults to `/opt/IBM/WebSphere/AppServer/products/sklm/data`.

Example 5-62 shows the use of the **scp** command to transfer the certificate file and the target directory `/opt/IBM/WebSphere/AppServer/products/sklm/data` on IBM Security Key Lifecycle Manager where the file must reside.

Example 5-62 Certificate transfer to IBM Security Key Lifecycle Manager

```
[root@sklma source]# scp ds8k_75xxxx1_gen3_cert.pem
root@0.0.0.1:/opt/IBM/WebSphere/AppServer/products/sklm/data/ds8k_75BRX70_gen2_
cert.pem 100% 1270     1.2 KBps   00:00
```

4. Import the Gen 3 certificate into the Client Device Certificates. Select **Advanced Configuration - Client Device Certificates**, as shown in Figure 5-188**.**

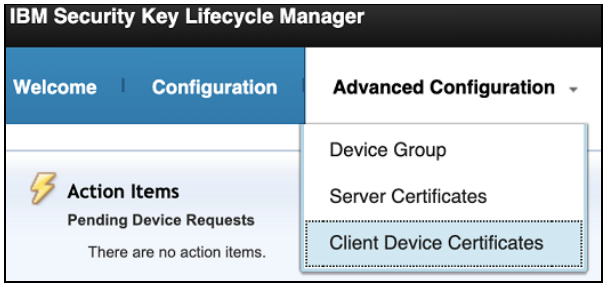


Figure 5-188 Client Device Certificates

From the table header, click **Import**, as shown in Figure 5-189.

Figure 5-189 IBM Security Key Lifecycle Manager - Import certificate

5. Enter a unique certificate name that you can recognize easily; for example, by including the DS8900F serial number. Click **Browse** and locate the certificate that you copied to the IBM Security Key Lifecycle Manager import directory. Finish the import process by clicking **Import** (Figure 5-190 on page 203).

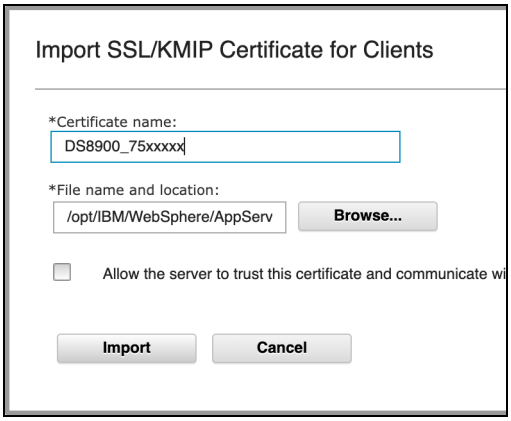


Figure 5-190 IBM Security Key Lifecycle Manager - Import certificate details

6. If TCT is activated, update TCT device group by completing the following steps:
   a. In the IBM Security Key Lifecycle Manager welcome window, select **DS8000-TCT** → **Manage keys and devices**, as shown in Figure 5-191.

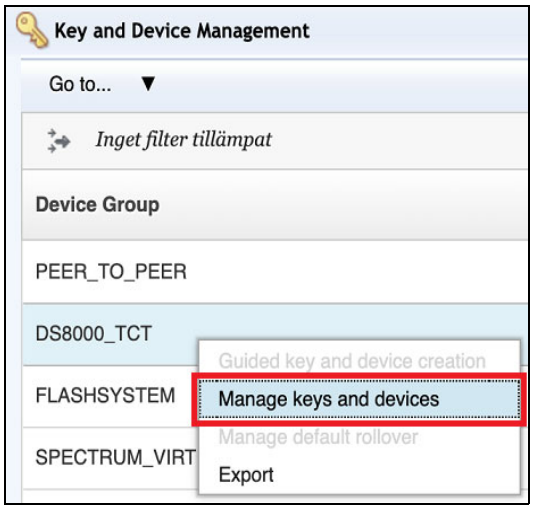


Figure 5-191 Select Manage Keys and devices

   b. Select **Add** and then click **Certificate**.

c. Select **Browse**, select the certificate, and **Select**. Click **Add**, as shown in Figure 5-192. Gen 3 certificate is now added to the DS8000 TCT Device Group.

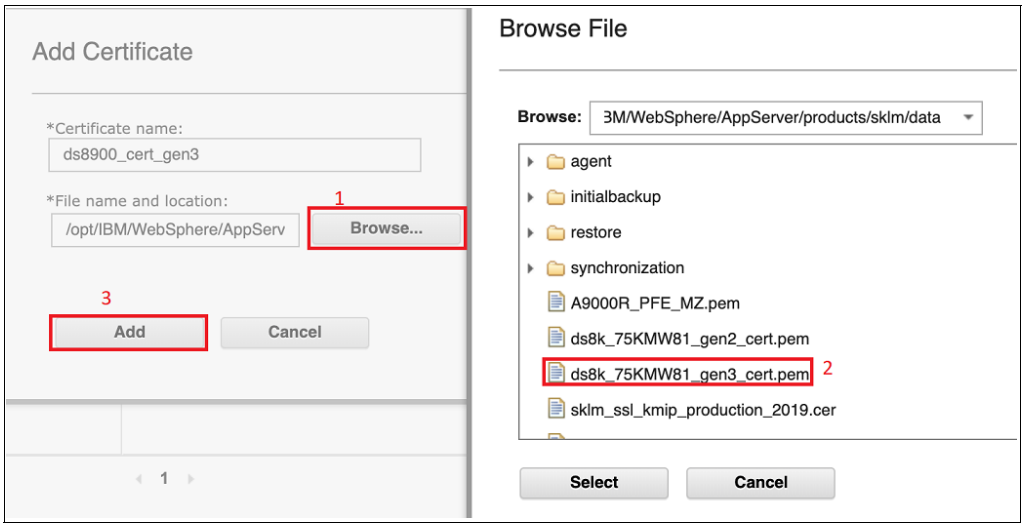


Figure 5-192 Add Certificate

7. If IBM Fibre Channel Endpoint Security is activated, update the diagnostic device group and all peer-to-peer groups in IBM Security Key Lifecycle Manager. Complete the following steps:

   a. In the welcome tab of IBM Security Key Lifecycle Manager, filter for the DS8900F WWNN in the Key and Device Management section.

      The DS8900F WWNN should appear in at least two peer-to-peer device groups. One is the diagnostic device group with the DS8900F WWNN repeated twice. The others are the IBM Z CPC - DS8900F association device groups whose names consist of the WWNN of the IBM Z CPC (as owner) and the WWNN of the DS8900F (as partner). One device group exists for each IBM Z and DS8900F pairing.

   b. Start with the z/DS8900F device groups. Highlight one, right-click, and select **Manage Keys and Devices**, as shown in Figure 5-193 on page 205.

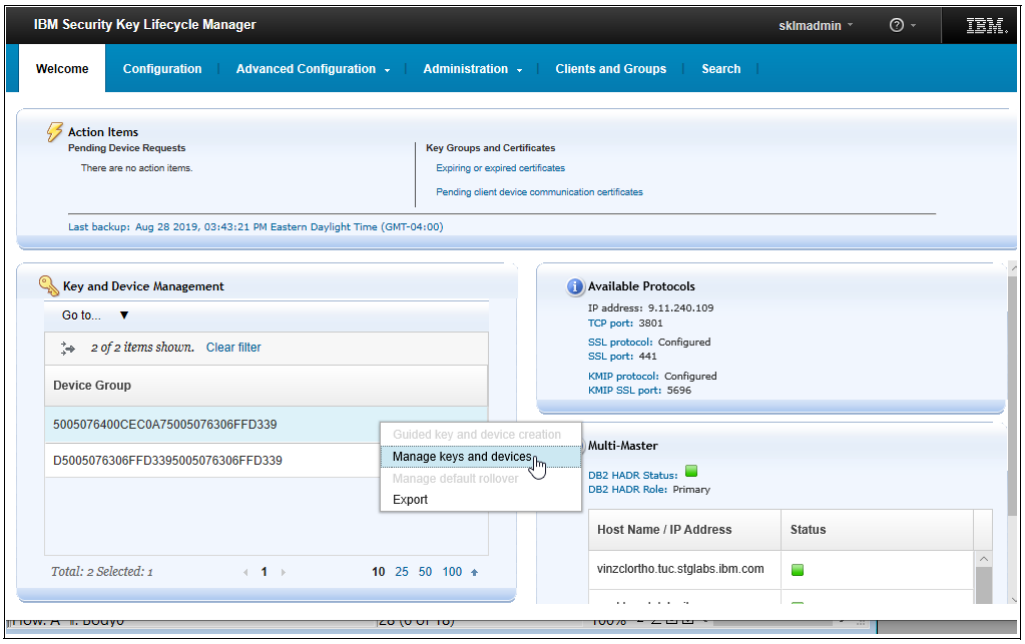


Figure 5-193 IBM Security Key Lifecycle Manager - Manage Keys and Devices

c. In the window, highlight the second item, which is the Partner device type, right click, and select **Modify**, as shown in Figure 5-194.

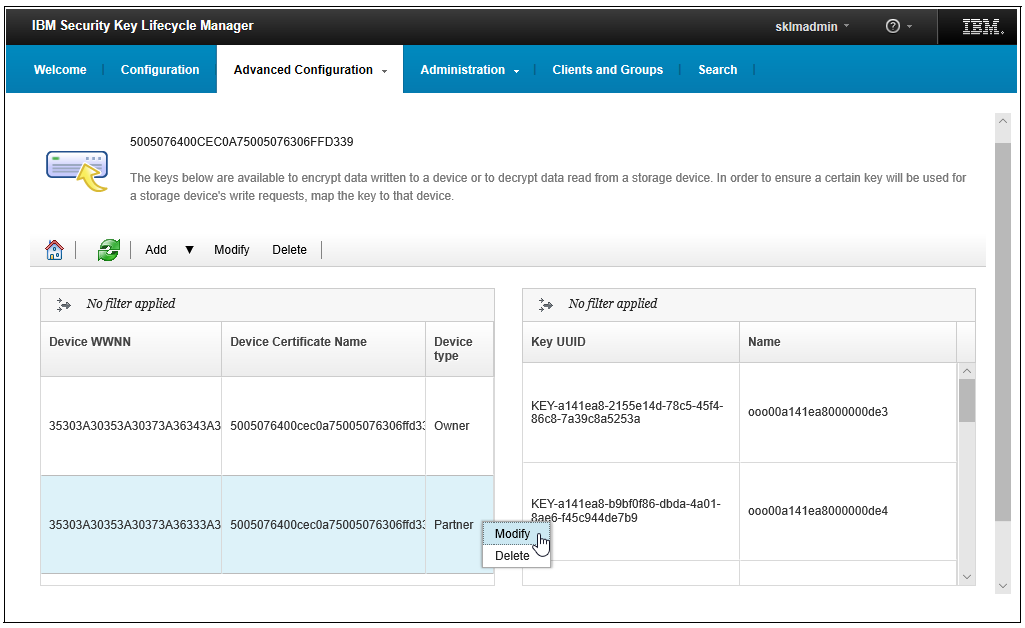


Figure 5-194 IBM Security Key Lifecycle Manager - Modify Device Group

d. Browse for the new DS8900F certificate that you imported and click **Modify** (Figure 5-195).

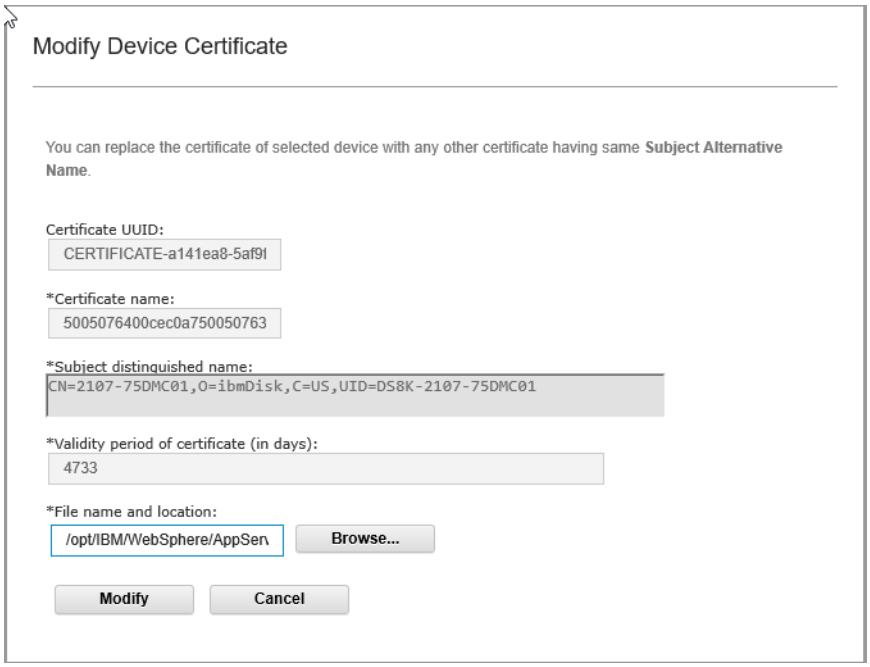


Figure 5-195 IBM Security Key Lifecycle Manager - Modify Device Certificate

Repeat these steps for all device groups to which the DS8900F belongs.

e. In the Welcome window, filter for the DS8900F WWNN again, and highlight the diagnostic device group whose name begins with the letter “D” followed by the DS8900F WWNN repeated twice. Right-click and select **Manage Keys and Devices**.

For this group, modify the owner and partner. For each, right-click and modify the certificate.

8. Upgrade the certificate on the DS8000.

If CLI is used, run the **`managekeygrp`** command, as shown in Example 5-63, to change from the Gen 2 to Gen 3 certificate on the DS8000.

Example 5-63 Update the certificate from Gen 2 to Gen 3 on the DS8000

```
dscli> managekeygrp -action updatecert -certType GEN3 -key data 1
CMUC00472I managekeygrp: The certificate for key group 1 has been updated.
```

Run the **`showkeygrp 1`** command to verify that the Gen 3 certificate is now being used for data encryption (see Example 5-59 on page 200).

Example 5-64 Verify that the certificate was updated from Gen 2 to Gen 3

```
dscli> showkeygrp 1
ID                          1
numranks                    4
numpools                    4
state                       accessible
reckeystate                 configured
reckeydate                  10/10/2019 14:48:09 CEST
datakeydate                 10/15/2019 18:19:51 CEST
grpstatus                   normal
```

```
mgrstatus                      normal
label                          -
label2                         -
certificate                    GEN3
certificate not valid before   08/22/2019 21:55:04 CEST
certificate not valid after    08/12/2032 02:36:55 CEST
certificate issuer             CN=IBM Disk Intermediate
CA,OU=Storage,O=IBM,ST=Arizona,C=US
certificate subject
CN=2107-75xxxx1,O=ibmDisk,C=US,uid=DS8K-2107-75xxxx1
endpoint enabled               Yes
certificate install state      Not Imported
pending cert valid not before  -
pending cert valid not after   -
pending cert issuer            -
pending cert subject           -
pending cert endpoint enabled  -
uuid                           KEY-4d74e7f-ba1db885-9ca3-49a7-b33c-94454a57eb3e
keyprotocol                    KMIP
type                           DAR
name                           DAR_1
```

If the DS GUI is used, complete the following steps:

a. In the DS8000 GUI Welcome window, click **Settings** and then, **Security** and then, **Data at rest**.

   The System defined (Gen 2) certificate is listed. Select **Update Certificate**, as shown in Figure 5-196.

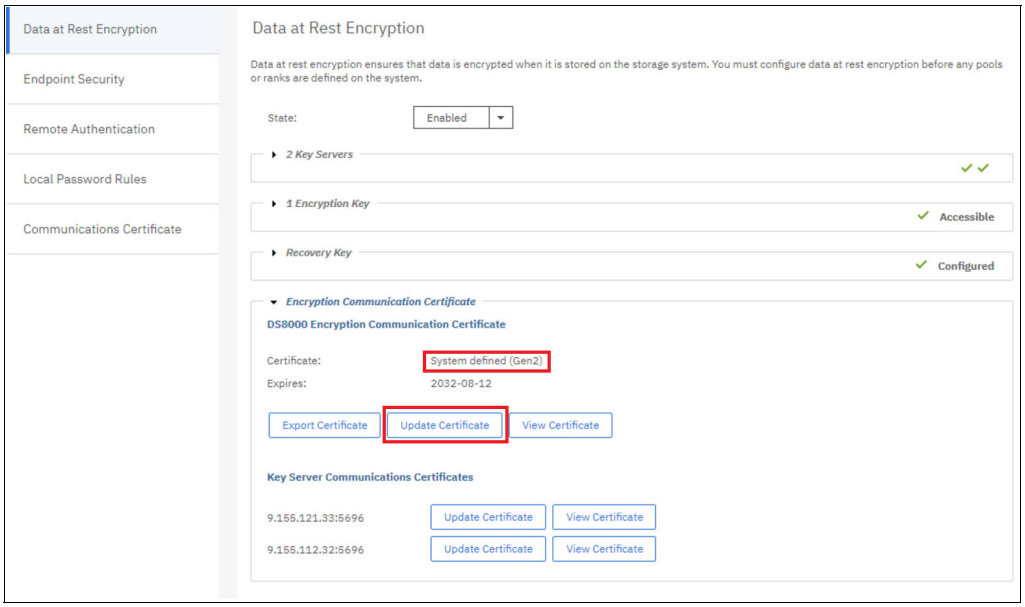


Figure 5-196 Update certificate

b. Select **System Defined (Gen3)** and **Update** (see Figure 5-197).

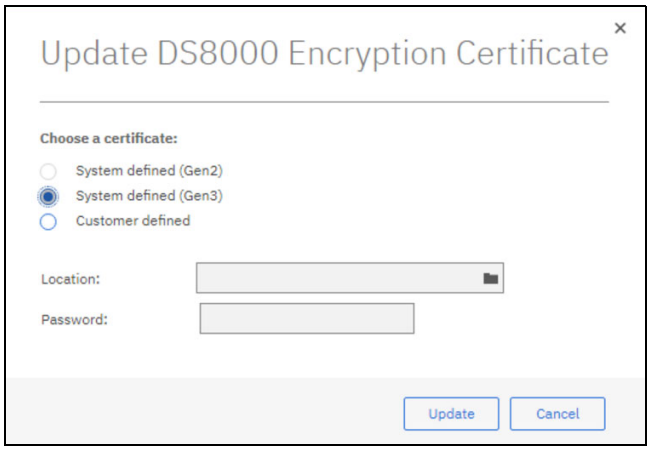


Figure 5-197 Update DS8000 Encryption Certificate

If the SSL certificate was updated from the key server and the data encryption certificate was updated to Gen 3, the DS8000 encryption configuration is now compliant with NIST SP 800-131a.

5.11 Using a custom-generated Gen 1 or Gen 2 certificate

A custom certificate can be used for communication between the encryption key servers (typically IBM Security Key Lifecycle Manager) and the storage system. You can update to a custom defined certificate by way of DS GUI or DS CLI.

Note: If the current DS8000 encryption certificate is Gen 1, before you update to a customer defined certificate, ensure that the 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 Gen 2 but not Gen 1.

5.11.1 Configuring a custom certificate by using the DS GUI

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 **Settings** → **Security** → **Encryption** then, select **Enable Encryption** to start the wizard (see Figure 5-198 on page 209).

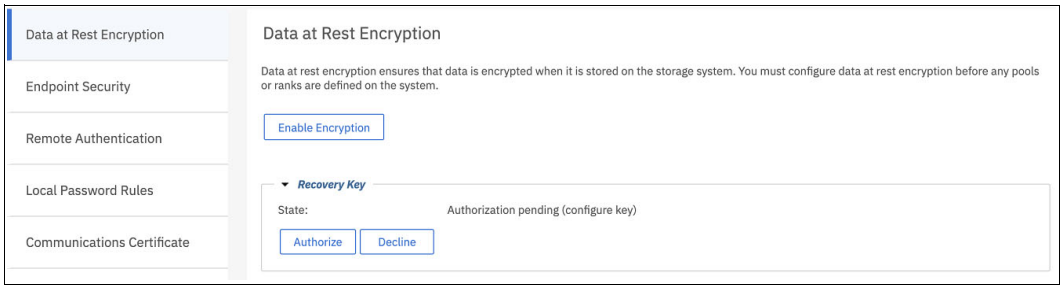


Figure 5-198 Enable Encryption

- Update Certificate on the Encryption Settings page when encryption is configured:

  a. To update the certificate, on the DS GUI home page select **Settings** → **Security** → **Encryption.** The Encryption window is shown in Figure 5-199.

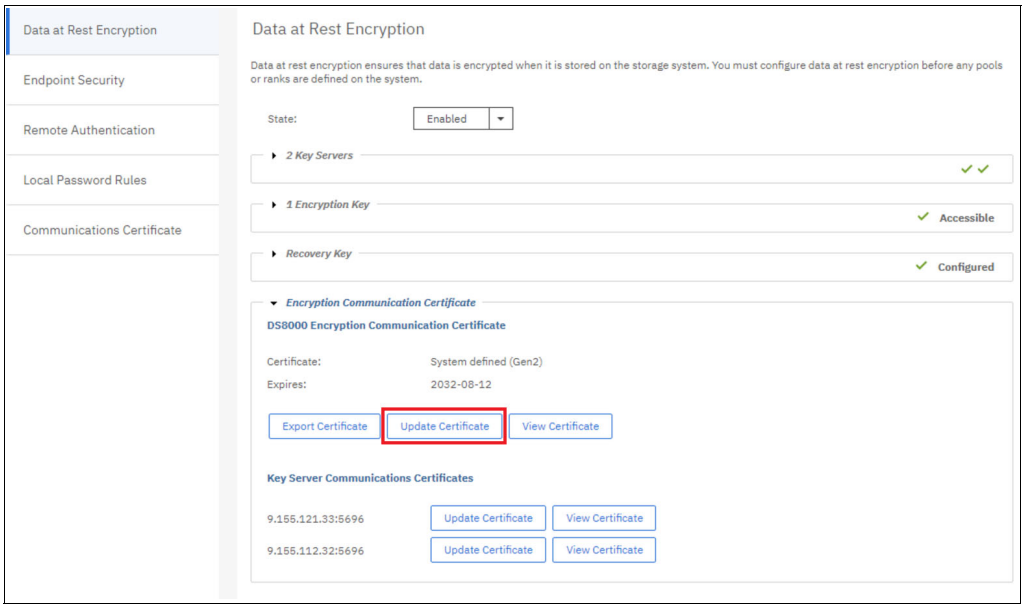


Figure 5-199 Settings Encryption window

b. Click **View Certificate** to view the DS8000 Encryption Certificate. Click **Update Certificate**. The Update DS8000 Encryption Certificate window opens (see Figure 5-200).

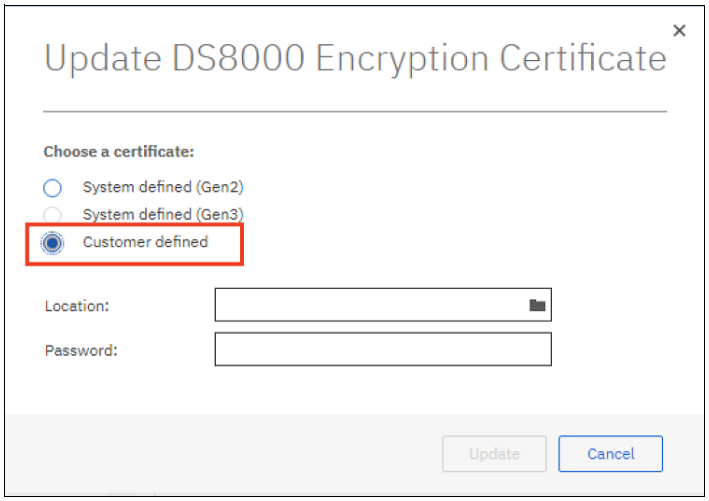


Figure 5-200 Encryption Settings Update Certificate

c. Click the **Customer defined** option In the Update DS8000 Encryption Certificate window. Select Customer that is defined. Browse for the certificate location and enter a password for the certificate. Click **Update** to update the certificate.

Note: The maximum length of customer-generated machine password is 128 characters. An example of a file name for a customer-generated file name is either 2107-75YZ123.mfg or 2107-75YZ123.p12 and password is XyZABcPQRstu.

5.11.2 Configuring a custom certificate by using DS CLI

Custom defined certificate can be specified by using **`managekeygrp`**, as follows:

- **`managekeygrp -action importcert -loc`** location **`-pw`** password encryption_group_ID (see Example 5-65).

Example 5-65 Importing custom certificate

```
dscli> managekeygrp -action importcert -loc /home/hscroot/rashm/da6.p12 -pw
blah 1
CMUC00489I managekeygrp: The certificate for encryption key group 1 has been
imported.
```

- **`managekeygrp -action updatecert -certType`** customer **`-key`** data encryption_group_ID (see Example 5-66).

Example 5-66 Updating custom certificate

```
dscli> managekeygrp -action updatecert -certType CUSTOMER -key data 1
CMUC00472I managekeygrp: The certificate for encryption key 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 Gen 2 option. Parameter **`-key`** is also required with the **`updatecert`** action.

6

Maintaining the IBM DS8000 encryption environment

This chapter provides information about the maintenance and use of your IBM DS8000 encryption environment and focuses on data at rest (DAR) encryption with IBM Security Key Lifecycle Manager.

Note: This chapter applies to DAR encryption only. Some illustrations in this chapter are based on former versions of IBM Security Key Lifecycle Manager and the DS8000 GUI, but the processes that described are still valid.

This chapter includes the following topics:

- 6.1, “Rekeying the data key for data at rest encryption” on page 214
- 6.2, “Recovery key use and maintenance” on page 221
- 6.3, “Recovery key state summary” on page 239

Important: For more information about maintaining the IBM Security Key Lifecycle Manager environment, see IBM Knowledge Center.

In particular, pay attention to the backup tasks. Failure to back up your keystore and other critical data properly can 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.

© Copyright IBM Corp. 2009, 2021. All rights reserved.

6.1 Rekeying the data key for data at rest encryption

The process of rekeying the data key (DK) depends on the communication protocol (IBM Proprietary Protocol or Key Management Interoperability Protocol (KMIP)) between IBM Security Key Lifecycle Manager and the DS8000.

6.1.1 Rekeying the data key when using the IBM Proprietary Protocol

The Rekey Data Key option is available on the DS8000. You can use this option with the Storage Administrator role to rekey the DK by changing the DK label. A client might want to use this function to change periodically the DK.

The following procedure describes:

- How to define a new certificate in the IBM Security Key Lifecycle Manager key servers.
- How to rekey the DK labels in the DS8000.

Defining a new key label/certificate in the IBM Security Key Lifecycle Manager key servers

To create a new certificate, log in to the Primary/Master IBM Security Key Lifecycle Manager key server, complete the following steps:

1. Highlight **DS8000** under **Device Group** in the **Key and Device Management** page, as shown in Figure 6-1.

> **Important:** You must use the predefined DS8000 Device Group. The use of custom-defined device groups is not supported.

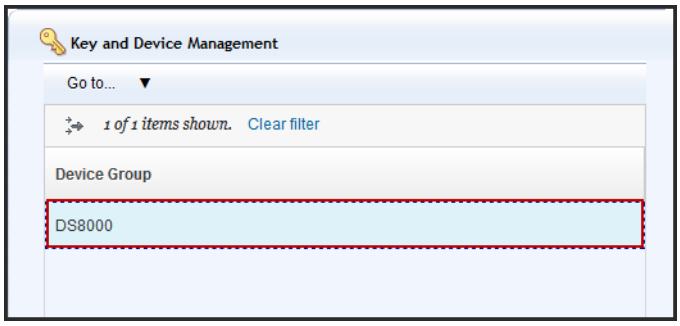


Figure 6-1 Key and Device Management

2. Right-click **DS8000** and select **Manage keys and devices**, as shown in Figure 6-2 on page 215.

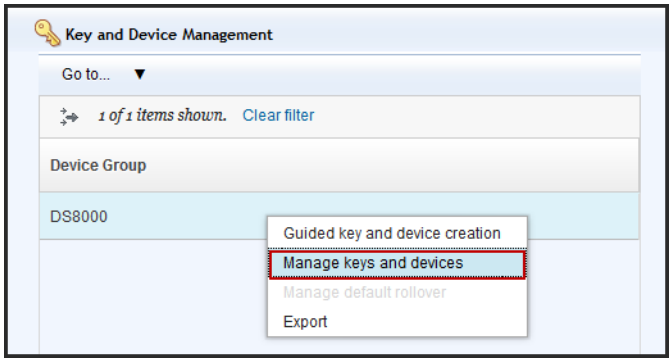


Figure 6-2 Manage keys and devices

The certificate that is in use is displayed on the left side. On the right side, the association to the Storage Image is indicated, as shown in Figure 6-3.

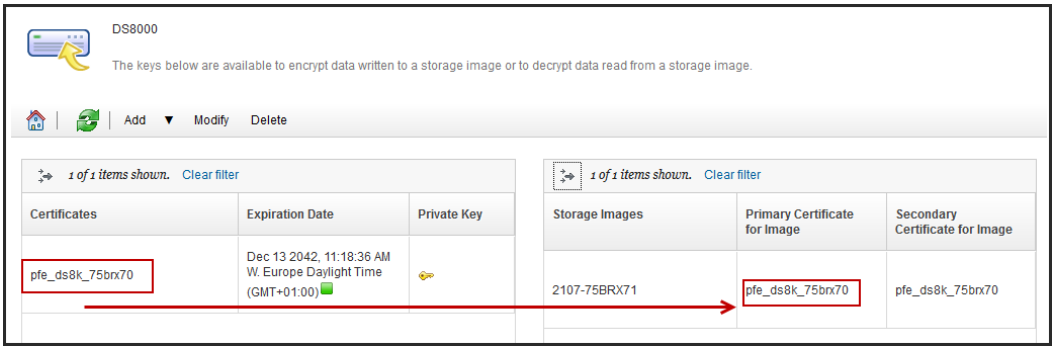


Figure 6-3 DS8000 devices and certificates

3. Select **Add** → **Certificate** to create a certificate, as shown in Figure 6-4.

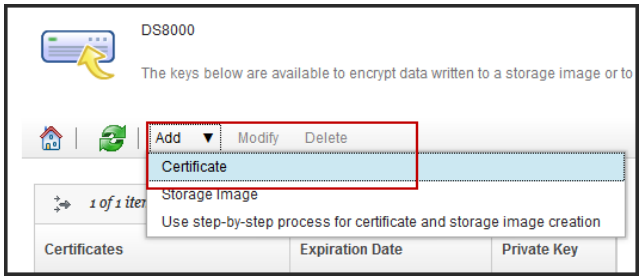


Figure 6-4 Add certificate

4. Create a self-signed certificate or Request certificate from a third-party provider, as explained in “Creating an SSL/KMIP server certificate” on page 95.
5. The new certificate appears in the list, as shown in Figure 6-5.

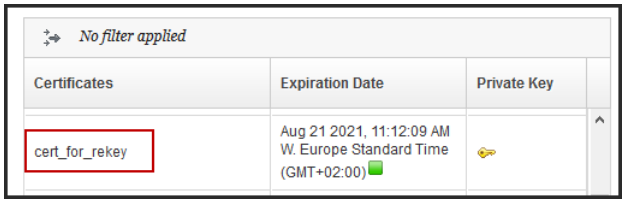


Figure 6-5 New certificate

6. Highlight the Storage Image to be rekeyed on the right side, right-click, and select **Modify**, as shown in Figure 6-6.

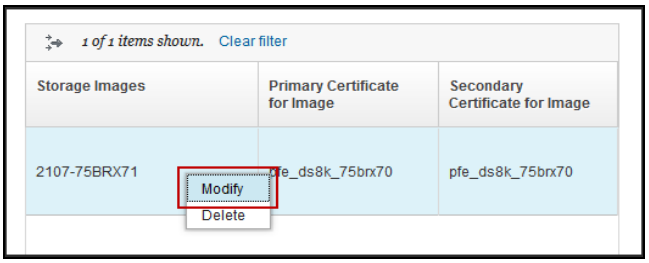


Figure 6-6 Modify Storage image

7. The Storage Image properties are displayed. Locate the Primary certificate for image section and click **Select** to assign a new primary certificate to the storage image, as shown in Figure 6-7.

Figure 6-7 Select primary certificate

8. Select the new certificate from step 4 on page 215, as shown in Figure 6-8.

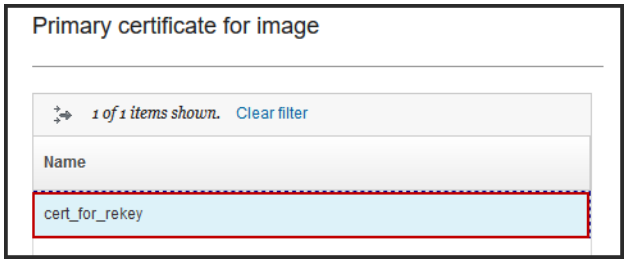


Figure 6-8 Select the new certificate

9. Repeat steps 7 and 8 for the Secondary certificate for image, as shown in Figure 6-9.

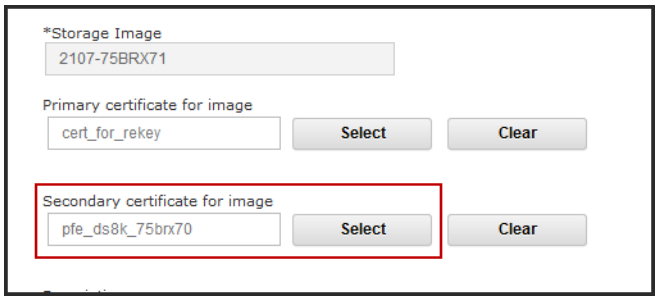


Figure 6-9 Select secondary certificate

10. After the new certificate is assigned as primary and secondary image certificate, click **Modify Storage Image**, as shown in Figure 6-10 on page 217.

Figure 6-10 Save: Modify Storage Image

The Storage Image now shows the new assigned certificate as Primary and Secondary, as shown in Figure 6-11.

Figure 6-11 New certificates for Image

Rekeying the data key in the DS8000

To rekey the DK, complete the following steps:

1. Log in to the DS8000 as a user with Administrator privileges, select **Settings** → **Encryption**, and expand **Key Labels**, as shown in Figure 6-12 (the figures are taken from the DS GUI R8.5, but the process has remained similar in the current DS GUI release).

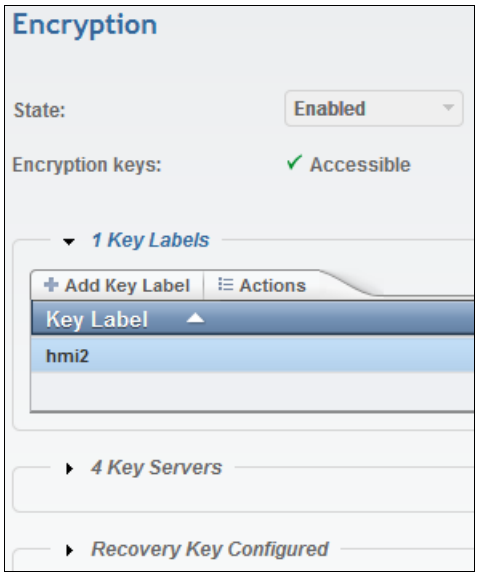


Figure 6-12 Starting the Rekey Data Key function

2. Select the key label that you want to change and from the **Actions** menu, select **Modify** (see Figure 6-13).

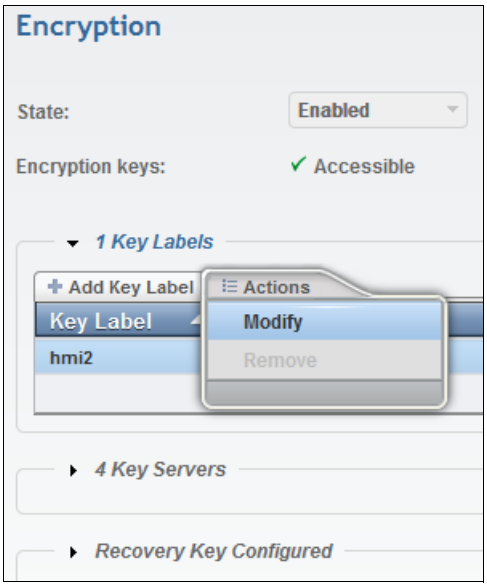


Figure 6-13 Rekey Data Key: Select Modify

3. The Modify Key Label window opens, as shown in Figure 6-14. Enter the new DK label that is defined in step 4 on page 215 into the provided field, and then, click **Modify**.

Figure 6-14 Rekey Data Key window

4. When the rekey task is complete, the confirmation message displays (see Figure 6-15 on page 219). Click **Close** to return to the main Encryption window.

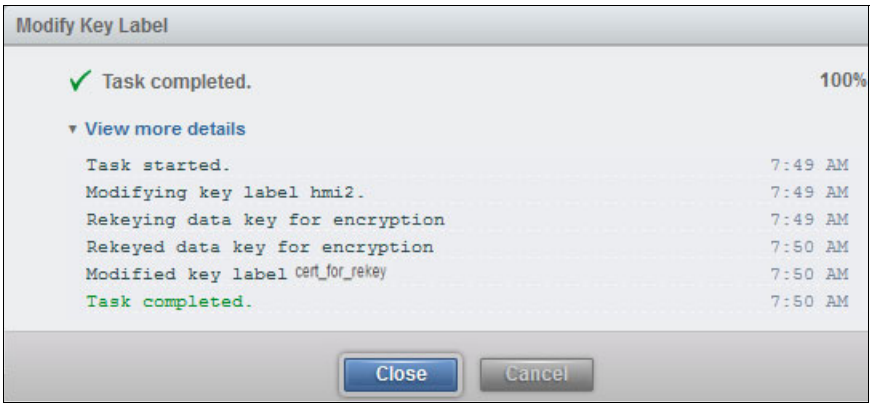


Figure 6-15 Rekey Data Key task complete

In the Encryption window, the new key label is displayed under the Key Labels section.

6.1.2 Rekey the data key when using the KMIP protocol

The Rekey Data Key option is available on the DS8000.

You can use this option with the Storage Administrator role to rekey the DK. A client might want to use this function to change the DK periodically.

To rekey the DK, complete the following steps:

1. Log in to the DS8000 as user with Administrator privileges and click **Settings** → **Encryption**.
2. Expand the Encryption key section, as shown in Figure 6-16. In the example, the current key is `KEY-4d74e7f-3eebe180-3194-474d-a2af-cc96f12e999a`.

Figure 6-16 Data at rest encryption

3. Click **Rekey** to start the Encryption Key Rekey process, as shown in Figure 6-17.

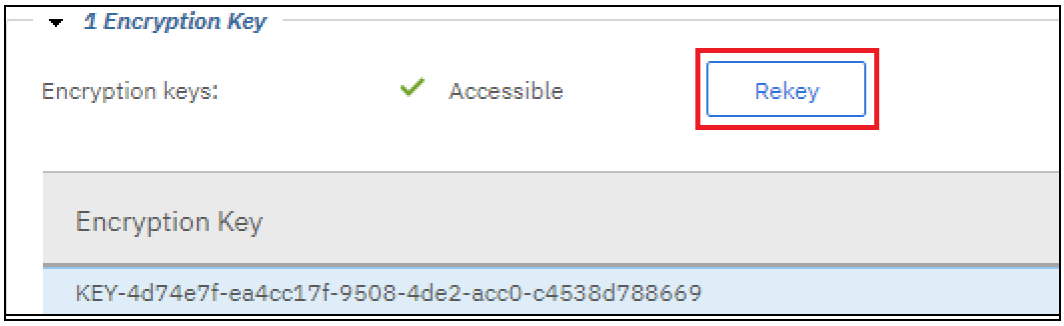


Figure 6-17 Rekey button

4. A window opens in which the rekey can be confirmed, as shown in Figure 6-18. You must decide whether to delete the old keys from the server. Make your choice and click **Yes**.

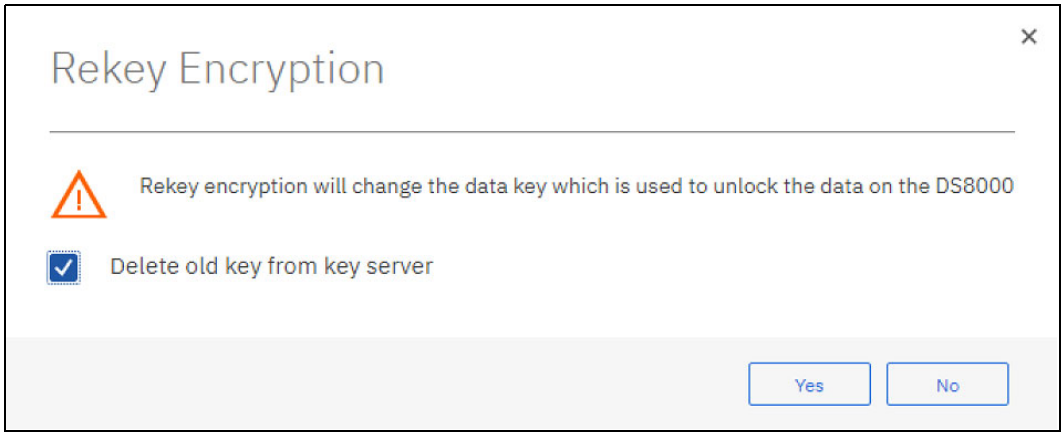


Figure 6-18 Confirm rekey

A confirmation window is displayed to confirm a successful rekey operation, as shown in Figure 6-19.

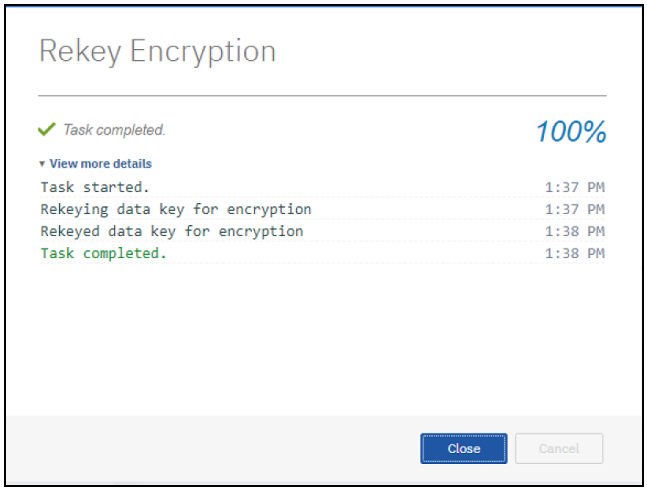


Figure 6-19 Rekey successful

5. Verify the new DK, as shown in Figure 6-20. The new key is `KEY-4d74e7f-3eebe180-3194-474d-a2af-cc96f12e999a`.

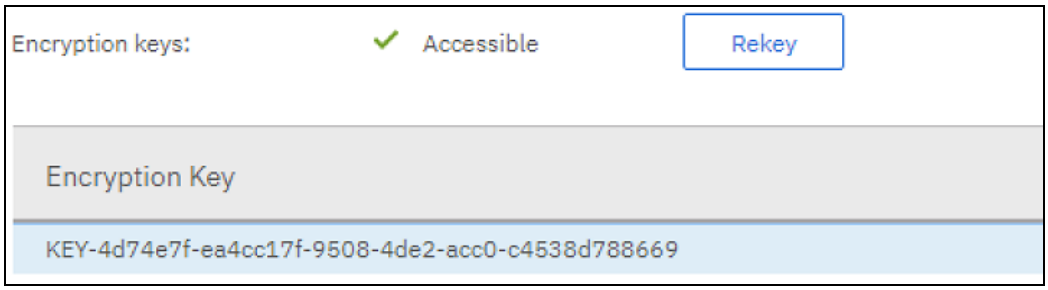


Figure 6-20 New data key

6.2 Recovery key use and maintenance

Starting with DS8870 Licensed Internal Code level 65.10.xx.xx, the following recovery-key-related options are available:

- Recovery key (RK) enabling
- RK disabling

When the RK is enabled, several functions are available to manage and use the RK 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 deconfigure a recovery key

When the RK is disabled, RK enablement is still possible.

The details of each function are described in this section.

6.2.1 Validating or testing a recovery key

Part of the RK creation process is the verification of a newly created RK. Verification ensures that the RK was written correctly. However, after the encryption environment is operational, the RK is used only in a deadlock situation. Therefore, verify that the stored RK 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 RK at any time. Only users with the Security Administrator authority can validate the RK. Verifying the RK does not change anything in the system, and Storage Administrator approval is not needed.

To validate or test an RK, 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 6-21.

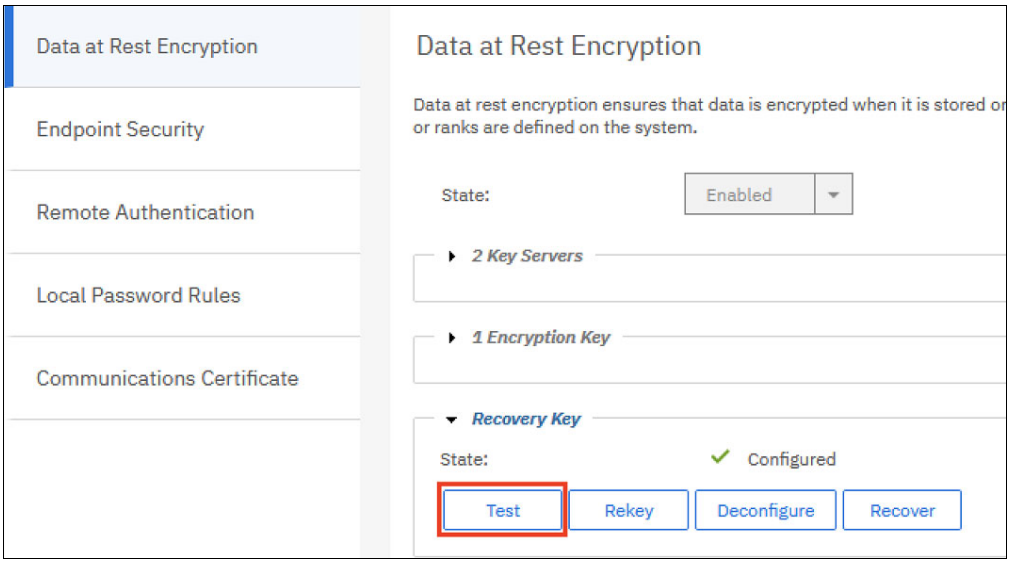


Figure 6-21 Start the Validate /Test Recovery Key function

2. The Test Recovery Key window opens (Figure 6-22). Enter the key into the field and click **Test**.

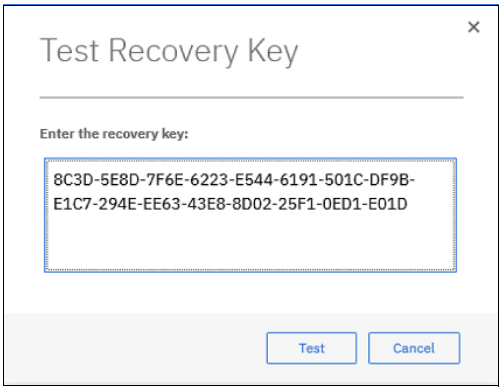


Figure 6-22 Test Recovery Key window

3. When the task is complete, the window in Figure 6-23 on page 223 opens. Click **Close**.

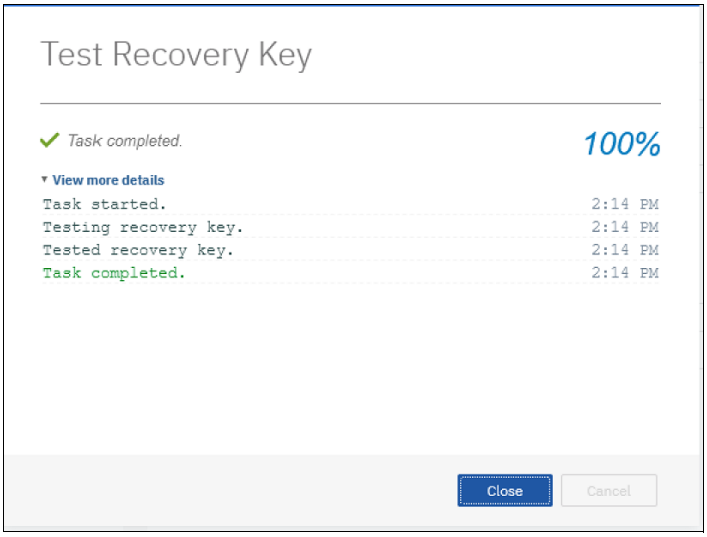


Figure 6-23 Test Recovery Key: Task completed

4. If the key is valid, a successful confirmation message is displayed, as shown in Figure 6-24. Click **OK** to return back to the Encryption window.

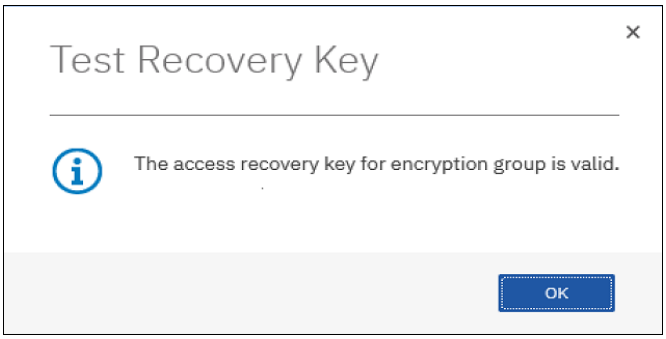


Figure 6-24 Test Recovery Key: Recovery key is valid

6.2.2 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.
- Start the recovery process to unlock the DS8000 and to enable the volumes to be accessible again.

Consider the first option and check the IBM Security Key Lifecycle Manager servers status first. Attempt 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 RK to initiate the process to unlock DS8000 volumes.

Simulating an IBM Security Key Lifecycle Manager failure

This section describes 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 6-25 (the preceding figures are from the DS GUI R8.5, but the process is the same).

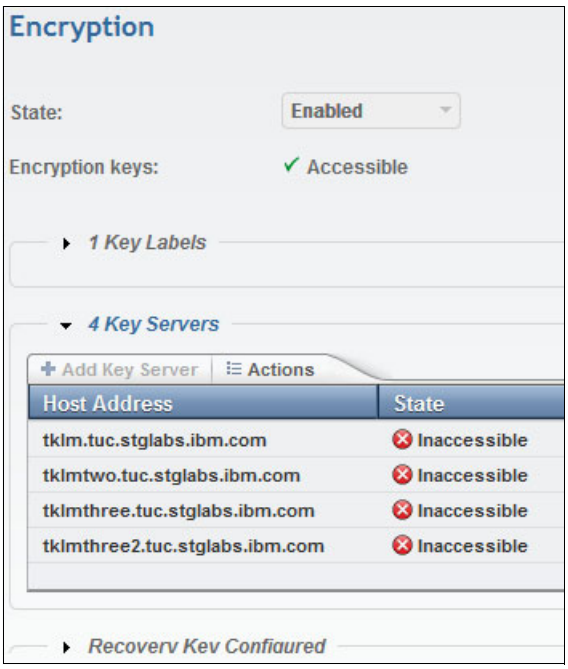
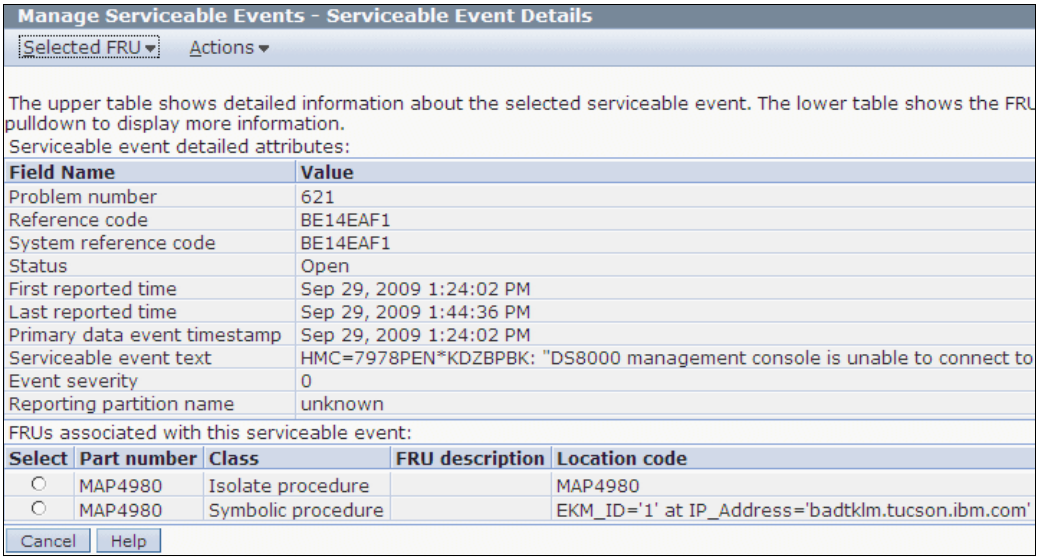


Figure 6-25 IBM Security Key Lifecycle Manager servers status is Inaccessible

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.

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 6-26), 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.



Manage Serviceable Events - Serviceable Event Details

Selected FRU Actions

The upper table shows detailed information about the selected serviceable event. The lower table shows the FRU pulldown to display more information.

Serviceable event detailed attributes:

Field Name	Value
Problem number	621
Reference code	BE14EAF1
System reference code	BE14EAF1
Status	Open
First reported time	Sep 29, 2009 1:24:02 PM
Last reported time	Sep 29, 2009 1:44:36 PM
Primary data event timestamp	Sep 29, 2009 1:24:02 PM
Serviceable event text	HMC=7978PEN*KDZBPBK: "DS8000 management console is unable to connect to
Event severity	0
Reporting partition name	unknown

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.ibm.com'

Cancel Help

Figure 6-26 BE14EAF1 SRC on DS8000 HMC

The IBM Security Key Lifecycle Manager key servers are accessed only when the key is required by 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.

The following SRC codes are available:

- 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 6-27 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 the FRUs
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 configured er
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.ibm.com'
○	MAP4980	Symbolic procedure		EKM_ID='2' at IP_Address='badtklm2.tucson.ibm.com'
○	MAP4980	Symbolic procedure		EKM_ID='0' at IP_Address='null'

Figure 6-27 BE14E008 SRC on DS8000 HMC

- BE1E2058: Config Error - Global Data Inaccessible, as shown in Figure 6-28.

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 6-28 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 6-29 (the figures are from the DS GUI R8.5, but the process is the same), the Encryption keys status is Inaccessible because of the broken communication between DS8000 and IBM Security Key Lifecycle Manager key servers. Click **Recover** to start the recovery process by using the RK.

Figure 6-29 Start the recovery process

2. The Recover Storage System window opens (see Figure 6-30). The recovery process requires the valid RK. Enter the key into the input field (uppercase characters with dash separation) and click **Recover**.

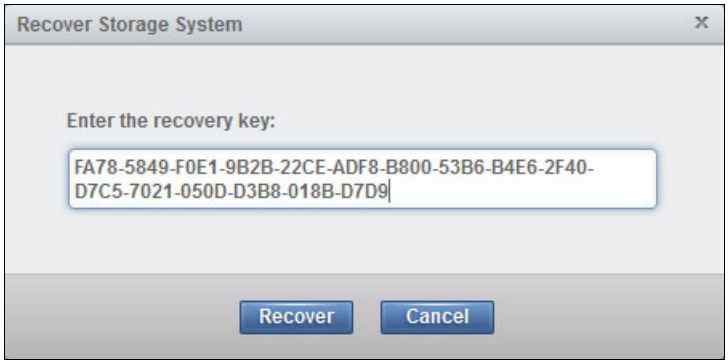


Figure 6-30 Enter the recovery key

3. A task completion message is displayed when the recovery task completes. Click **Close** to continue (see Figure 6-31).

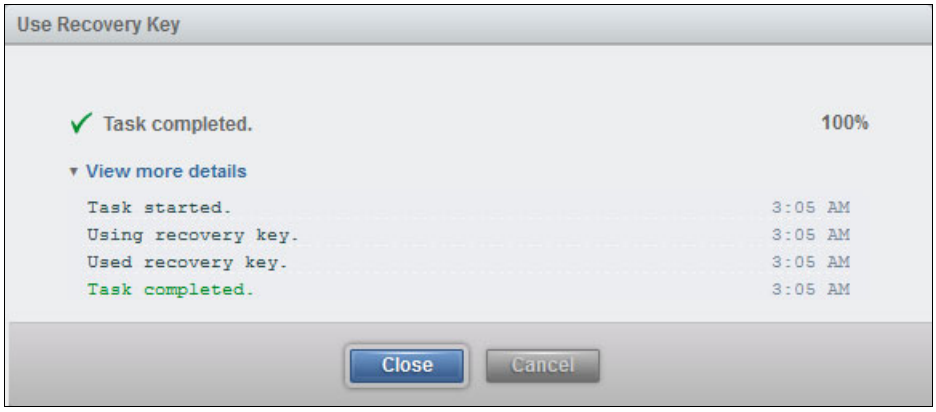


Figure 6-31 The recovery process was initiated successfully

Figure 6-32 shows that the RK has a pending authorization request that is addressed to Storage Administrator user. The Encryption keys are still in the Inaccessible state.

Figure 6-32 Request Recovery Authorization Pending state

4. 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. Two options available: Authorize and Decline (see Figure 6-33).

Figure 6-33 Start Authorize Recovery Key Update

5. After the authorization task completes, click **Close** to complete the recovery action (see Figure 6-34).

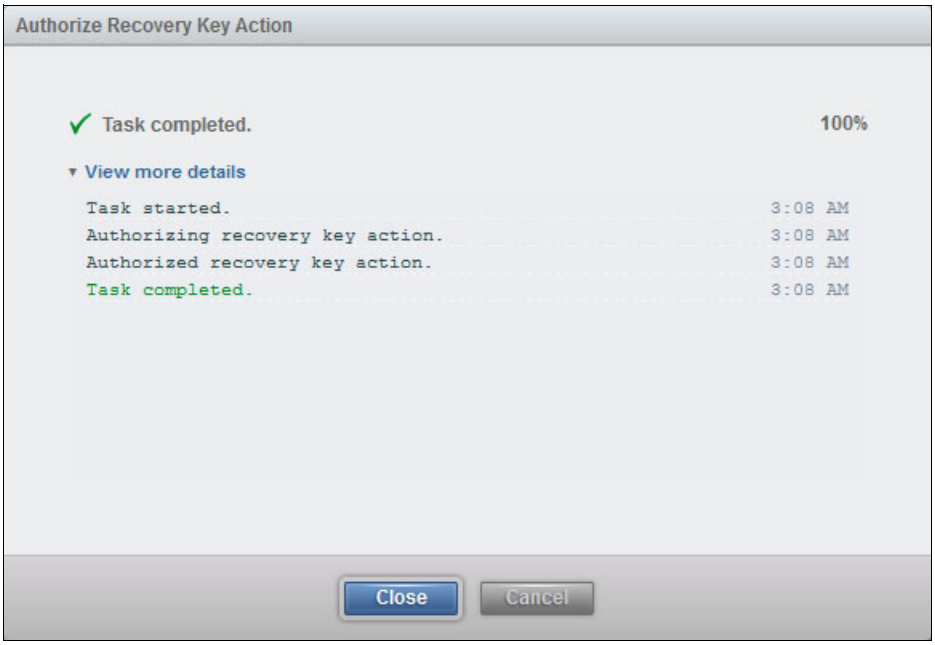


Figure 6-34 Authorize Recovery Key task completed

The recovery process is now completed and you are logged off because of the DS8000 restart process.

You can follow the restart process by viewing the DS8000 status messages at the HMC (see Figure 6-35).

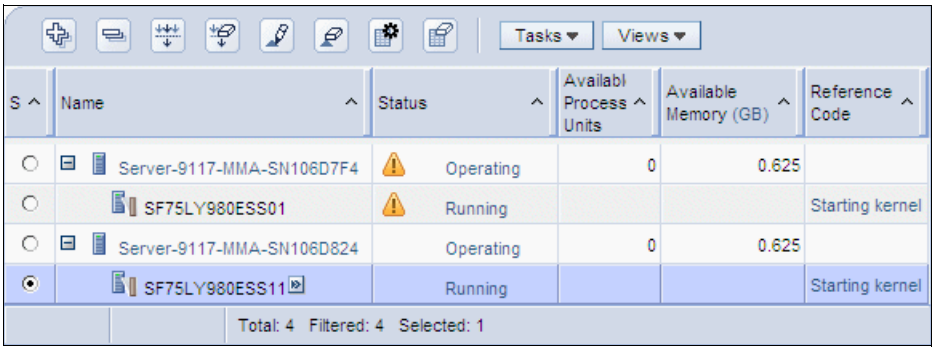


Figure 6-35 Storage facility image restart is in progress

6. 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 6-36).

Figure 6-36 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 6-37 shows the flowchart and corresponding DS Command-Line Interface (DS CLI) commands of the recovery process.

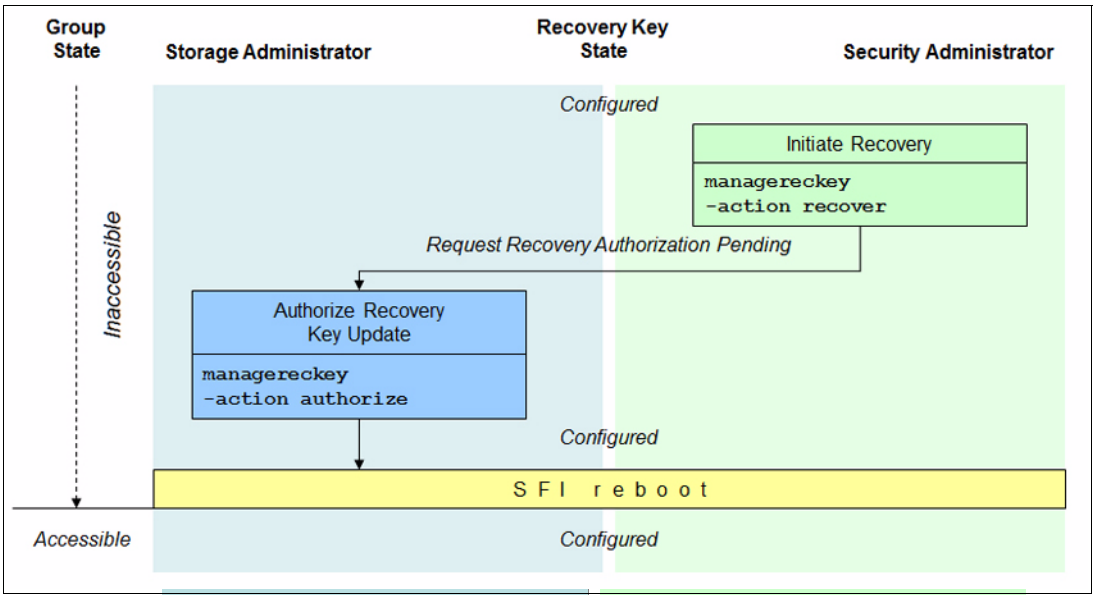


Figure 6-37 Initiate Recovery flowchart

6.2.3 Rekeying the recovery key

In the case of a lost RK or an unauthorized person is suspected to gain access to the key, the Security Administrator can generate a new RK. The old key is revoked and cannot be used anymore. The name of this process is *rekey RK*.

To rekey the RK, 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 6-38.

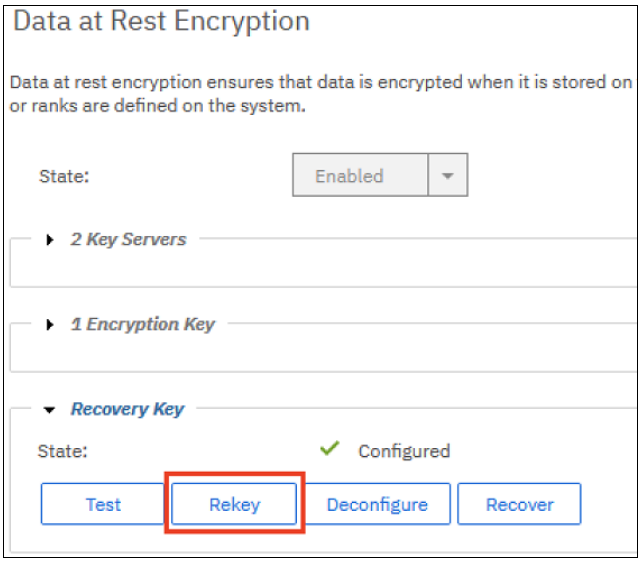


Figure 6-38 Rekey recovery key

2. The rekey task starts. After it completes, the task completion message displays (see Figure 6-39). Click **Close** to continue.

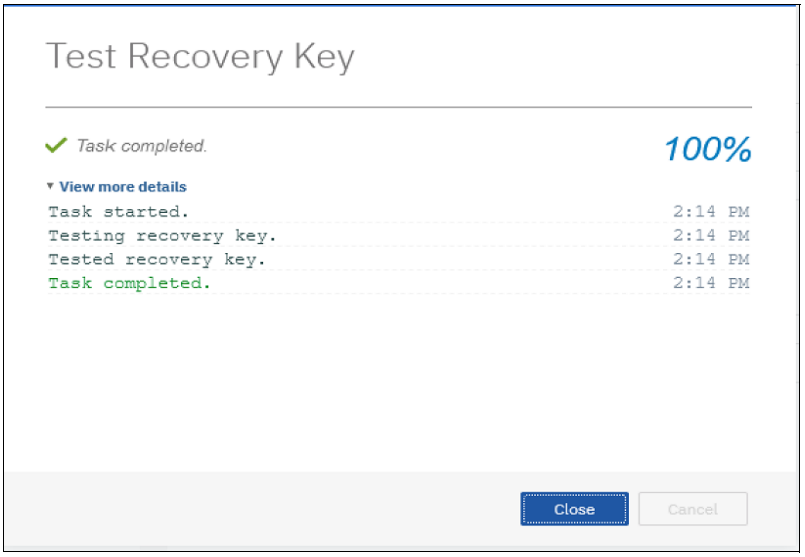


Figure 6-39 Rekey recovery key: Task completion

3. The next window (see Figure 6-40) displays the newly generated RK. You must record the key (select and copy). It is required for validation in step 4. Click **Rekey**.

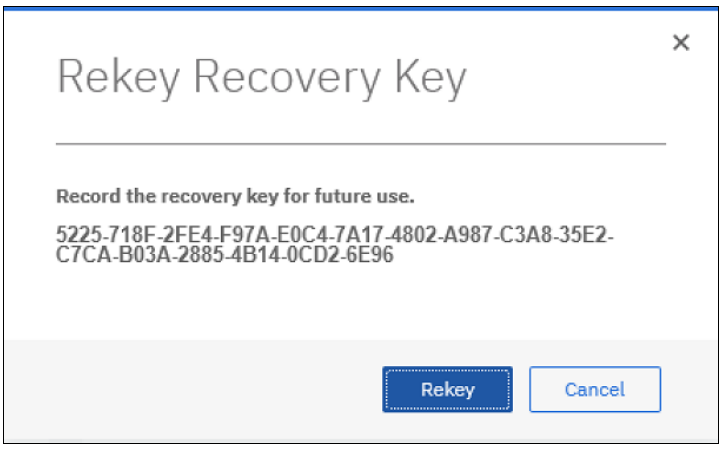


Figure 6-40 Rekey recovery key: Generated

4. Although the new key replaces the old one, do not destroy the old key yet because the old key is still active until the Storage Administrator approves the new RK. The process is similar to the process of creating a key (see “Creating the recovery key” on page 161).

Verify that the new key is written correctly by entering the key text into the input field and clicking **Test** (see Figure 6-41).

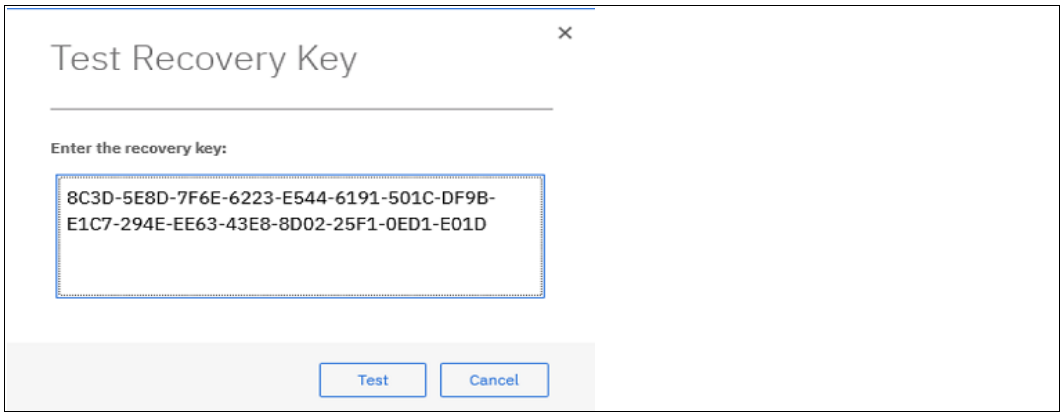


Figure 6-41 Rekey recovery key: Verification

5. When the RK verification task completes, the confirmation message displays (see Figure 6-42). Click **Close**.

Figure 6-42 Rekey recovery key: Verification

6. The result is shown with instructions about what to do next (see Figure 6-43).

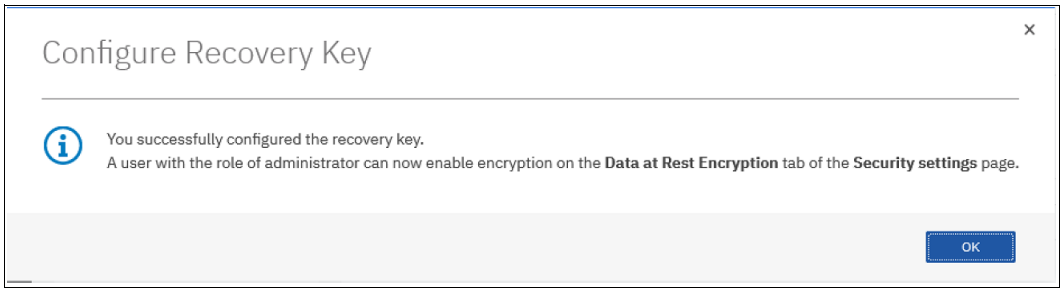


Figure 6-43 Configure Recovery Key window

7. In the Encryption window (see Figure 6-44), the RK 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 RK.

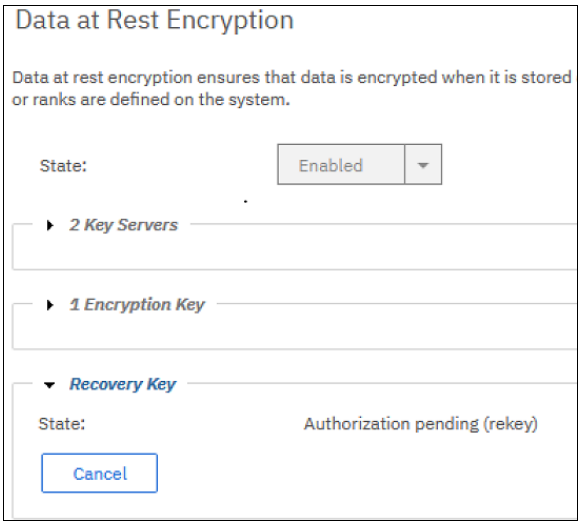


Figure 6-44 Rekey recovery key: Authorization pending

8. A user with Storage Administrator authority logs on and clicks **Settings** → **Encryption**. Expand the Recovery Key section and click **Authorize**, as shown in Figure 6-45.

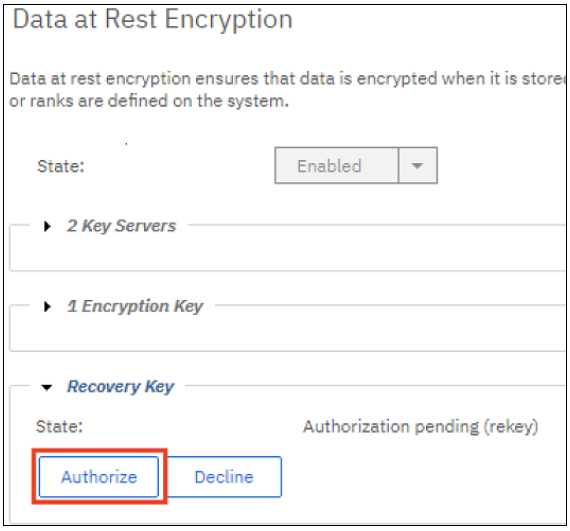


Figure 6-45 Authorize the recovery key

9. The RK authorization task starts. After it completes, a window with the confirmation message opens, as shown in Figure 6-46. Click **Close**.

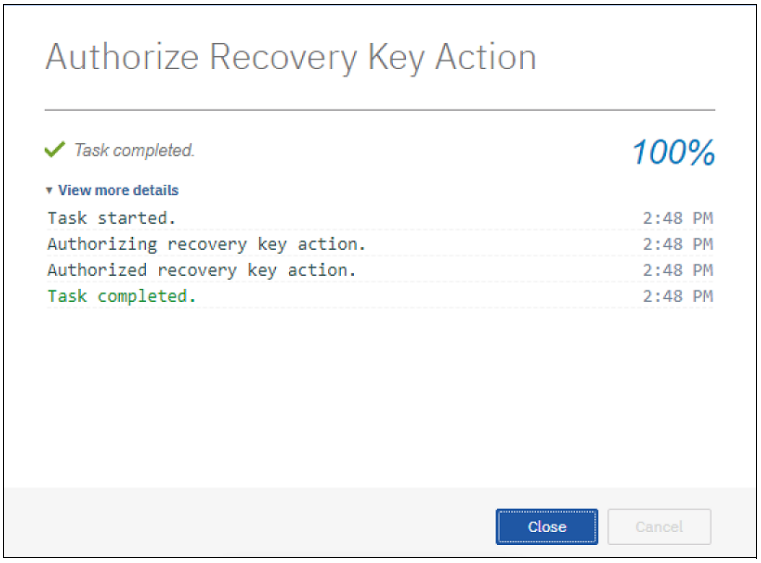


Figure 6-46 Authorize recovery key: Task completion

Now, only the new RK is valid. The old key is revoked. The state of the key changes back to a Configured state, as shown in Figure 6-47.

Figure 6-47 Recovery key configured

The flowchart of the rekey RK process is shown in Figure 6-48. The corresponding DS CLI commands are also provided.

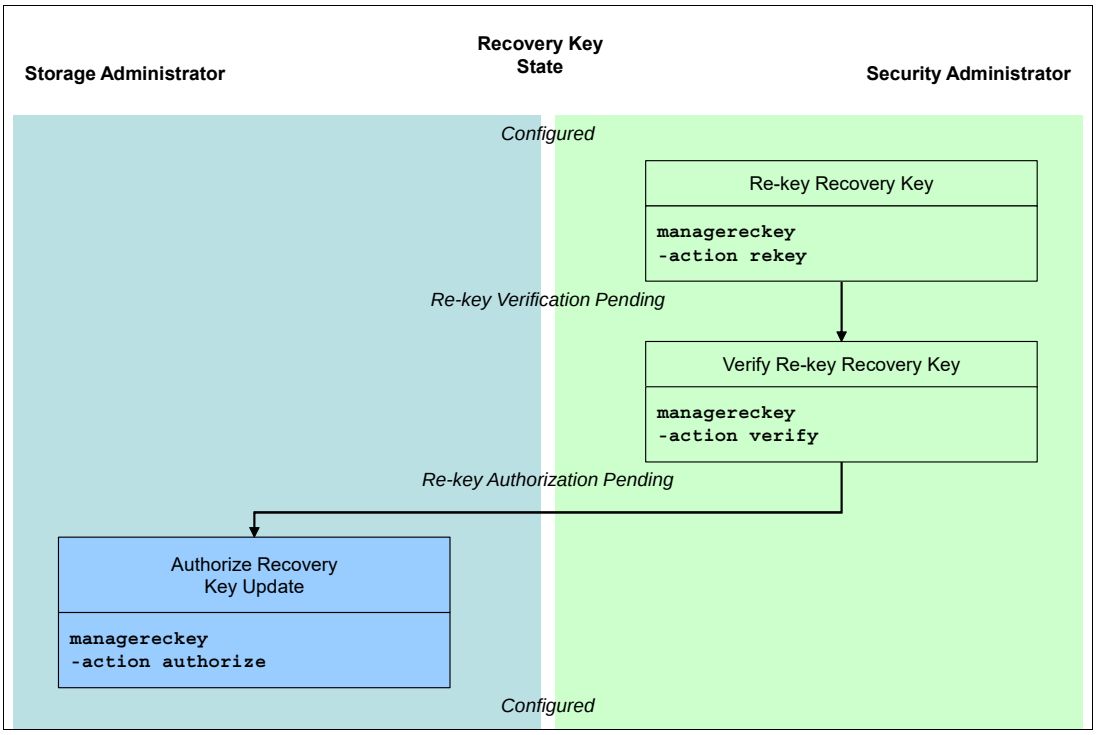


Figure 6-48 Rekey recovery key flowchart

6.2.4 Deleting or deconfigure a recovery key

Deleting the actual RK 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 an RK, 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 6-49 on page 237.

Figure 6-49 Delete/deconfigure recovery key

2. The Deconfigure Recovery Key task starts. After it completes, the window with the confirmation message opens, as shown in Figure 6-50. Click **Close**.

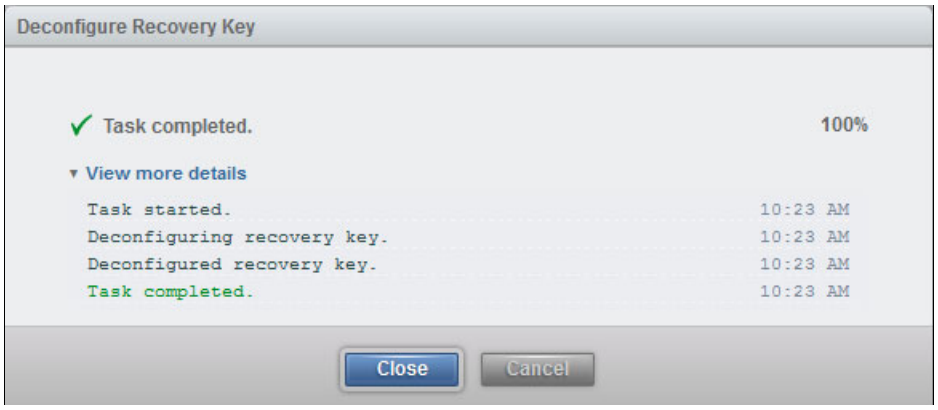


Figure 6-50 Delete Recovery Key window

Figure 6-51 shows that the Recovery Key State changed to a Deconfigure Key Authorization Pending state.

Figure 6-51 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 RK. The encryption is disabled and starting from this state, non-encrypted arrays, and ranks can be created on this storage system.

Figure 6-52 shows the flowchart and the corresponding DS CLI commands of the delete RK process.

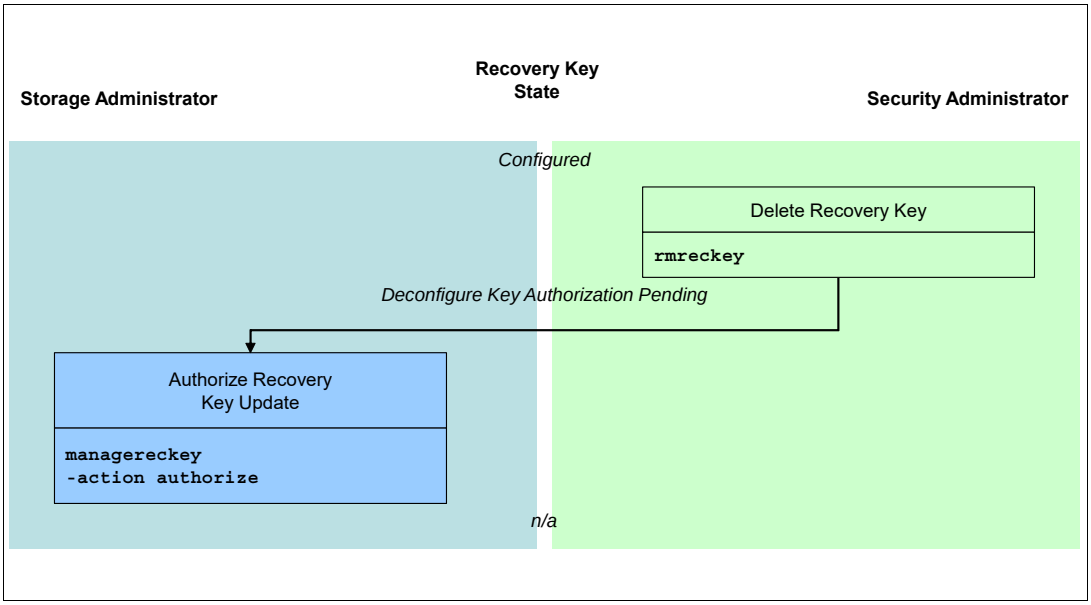


Figure 6-52 Delete recovery key flowchart

6.3 Recovery key state summary

This chapter introduced RK functions. In most cases, the RK has multiple temporary states. Figure 6-53 summarizes the possible RK states and their relationships.

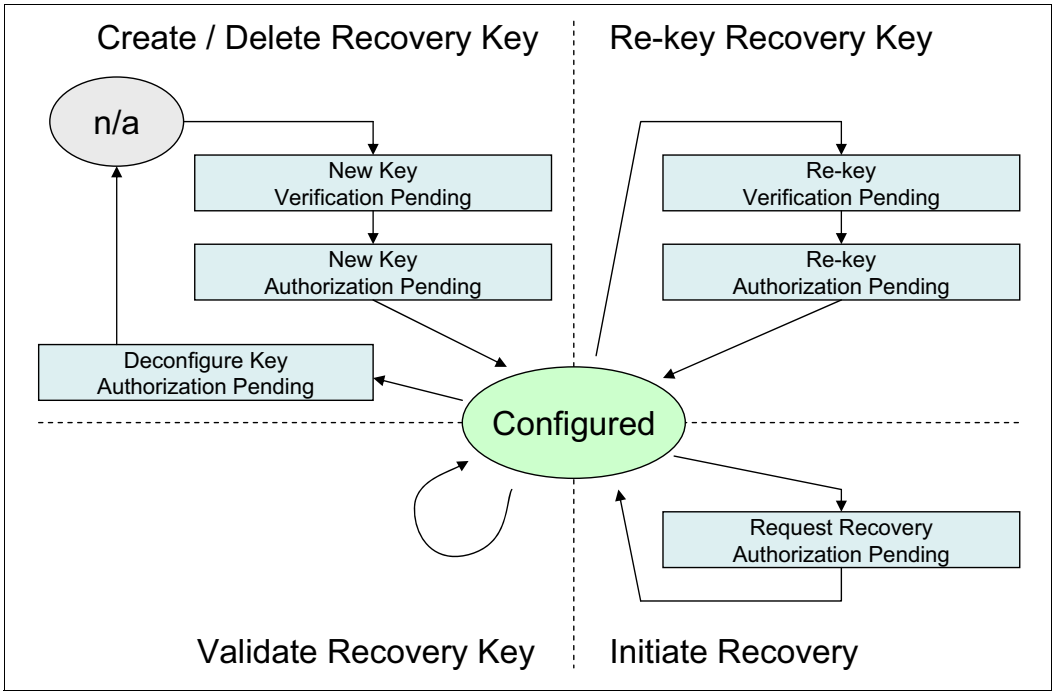


Figure 6-53 Overview of the recovery key states

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 publications provide more information about the topic in this document. They might be available in softcopy only:

- *IBM DS8000 and Transparent Cloud Tiering (DS8000 Release 9.1)*, SG24-8381
- *IBM DS8880 Architecture and Implementation (Release 8.51)*, SG24-8323
- *IBM DS8900F Architecture and Implementation Release 9.1*, SG24-8456
- *IBM Fibre Channel Endpoint Security for IBM DS8900F and IBM Z*, SG24-8455
- *IBM z15 Technical Introduction*, SG24-8850

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

The following publications are also relevant as further information sources:

- *IBM DS8880 Release 8.5 Introduction and Planning Guide*, GC27-8525
- *IBM DS8900 Introduction and Planning Guide,* GC27-9560
- *IBM Security Key Lifecycle Manager for z/OS Version 1.1 Planning, and User's Guide*, SC14-7628

Online resources

The following websites are also relevant as further information sources:

- DS8000 IBM Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/SSHGBU_9.0.0
- IBM Security Key LifeCycle Manager:

  https://www.ibm.com/support/knowledgecenter/SSWPVP

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

© Copyright IBM Corp. 2009, 2021. All rights reserved.

REDP-4500-09

ISBN 073845964X

Printed in U.S.A.

Get connected

Redbooks®
ibm.com/redbooks