

# Data-at-rest Encryption for the IBM Spectrum Accelerate Family

IBM XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R

Roman Fridli

Andrew Greenfield

Bert Dufrasne



**Security**

**Storage**





International Technical Support Organization

**Data-at-rest Encryption for the IBM Spectrum  
Accelerate Family**

December 2016

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

**First Edition (December 2016)**

This edition applies to IBM XIV Storage System Gen3, and IBM FlashSystem A9000 and IBM FlashSystem A9000R with Software Version 12.0.2.

This document was created or updated on December 28, 2016.

© Copyright International Business Machines Corporation 2016. 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

<b>Notices</b> .....	v
Trademarks .....	vi
<b>Preface</b> .....	vii
Authors .....	vii
Now you can become a published author, too! .....	viii
Comments welcome .....	viii
Stay connected to IBM Redbooks .....	viii
<b>Chapter 1. Encryption overview</b> .....	1
1.1 Introduction to data-at-rest encryption .....	2
1.2 Threats and security challenges .....	2
1.3 Need for encryption .....	3
1.4 Encryption concepts .....	3
1.5 Encryption challenges .....	4
<b>Chapter 2. IBM XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R encryption architecture</b> .....	5
2.1 Architecture overview .....	6
2.2 Encryption architecture .....	7
2.2.1 Configuring encryption .....	8
2.2.2 Digital certificates .....	9
2.2.3 Encryption techniques .....	10
2.3 XIV disk encryption .....	11
2.3.1 Self-encrypting drives .....	11
2.4 IBM FlashSystem A9000 or IBM FlashSystem A9000R encryption .....	13
<b>Chapter 3. Planning</b> .....	15
3.1 Planning and implementation process flow .....	16
3.2 Required and optional tasks .....	17
3.3 IBM Security Key Lifecycle Manager licensing .....	17
3.3.1 IBM Security Key Lifecycle Manager licensing outside of virtualization .....	17
3.3.2 IBM Security Key Lifecycle Manager licensing under virtualization .....	18
3.4 Preferred practices for encrypting storage environments .....	19
3.4.1 Security .....	19
3.4.2 Availability .....	19
3.4.3 Encryption administration .....	20
3.5 Multiple IBM Security Key Lifecycle Managers for redundancy .....	22
3.5.1 Setting up IBM Security Key Lifecycle Manager servers .....	23
<b>Chapter 4. Implementing IBM XIV encryption</b> .....	25
4.1 Encryption process overview .....	26
4.2 IBM Security Key Lifecycle Manager installation .....	27
4.3 XIV data-at-rest encryption configuration .....	28
4.3.1 Overview of configuration steps .....	28
4.3.2 Detailed configuration steps .....	28
4.4 Recovery key use and maintenance .....	45
4.4.1 Process for recovery keys .....	46
4.4.2 Recovery key generation with the XIV GUI .....	46

4.4.3	Recovery key generation with XCLI	49
4.4.4	Recovery key verification by using the XIV GUI	50
4.4.5	Recovery key verification by using the XCLI	51
4.4.6	Recovery key rekey	52
4.4.7	Using a recovery key to unlock an XIV system	53
4.5	Activating or deactivating encryption	55
4.5.1	Activating data-at-rest XIV encryption	55
4.5.2	Deactivating XIV data-at-rest encryption	56
4.6	Verifying encryption state	57
<b>Chapter 5. Implementing IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption</b>		
5.1	Encryption process overview	60
5.2	IBM Security Key Lifecycle Manager installation	61
5.3	IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption configuration	62
5.3.1	Overview of configuration steps	62
5.3.2	Detailed configuration steps	62
5.4	Recovery key use and maintenance	78
5.4.1	Process for recovery keys	79
5.4.2	Recovery key generation with XCLI	80
5.4.3	Recovery key verification	81
5.4.4	Recovery key rekey	82
5.4.5	Using a recovery key to unlock an IBM FlashSystem A9000 or IBM FlashSystem A9000R system	83
5.5	Activating or deactivating encryption	85
5.5.1	Activating data-at-rest encryption	85
5.5.2	Deactivating data-at-rest encryption	86
5.6	Verifying the encryption state	86
<b>Chapter 6. Maintaining</b>		
6.1	Automated replication	92
6.2	Starting and stopping an IBM Security Key Lifecycle Manager server	93
6.3	Server rekey	95
6.3.1	XIV system rekey by using the XIV GUI	95
6.3.2	XIV and FlashSystem A9000 or A9000R server rekey by using XCLI	97
6.4	Encryption deadlock	97
6.5	Disk and module replacement	99
<b>Related publications</b>		
	IBM Redbooks	101
	Other publications and online resources	101
	Help from IBM	102

# Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US*

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

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

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

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

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

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

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

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


## COPYRIGHT LICENSE:

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

# Trademarks

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

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

AIX®	IBM Spectrum Virtualize™	Storwize®
DB2®	MicroLatency®	System Storage®
DS5000™	Power Systems™	Tivoli®
IBM®	Redbooks®	XIV®
IBM FlashSystem®	Redpaper™	
IBM Spectrum™	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

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

# Preface

With the ever-growing landscape of national, state, and local regulations, industry requirements, and increased security threats, ensuring the protection of an organization's information is a key part of operating a successful business.

Encrypting data-at-rest is a key element when addressing these concerns. Most storage products offer encryption at an additional cost. IBM® XIV® Storage System, IBM FlashSystem® A9000, and IBM FlashSystem A9000R systems provide data-at-rest encryption at no charge. Clients can take advantage of encryption and still benefit from the lower total cost of ownership (TCO) that the IBM Spectrum™ Accelerate family offers.

This IBM Redpaper™ publication explains the architecture and design of the XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption solution. Details are provided for configuring and implementing encryption.

**Note:** IBM Security Key Lifecycle Manager Version 2.5 and 2.6 were used in preparation of this paper.

## Authors

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

**Roman Fridli** is a certified IBM XIV Product Field Engineer based in Switzerland. He joined IBM in 1998 as a Customer Engineer for IBM Power Systems™ and Intel Servers, including point-of-sale devices. Since 2012, he has worked for the XIV PFE EMEA-Team based in Mainz, Germany. He holds a degree in Electrical Engineering and multiple certifications in the storage solution and networking areas.

**Andrew Greenfield** is an IBM Global XIV and Flash Solution Engineer who is based in Phoenix, Arizona. He holds numerous technical certifications from Cisco, Microsoft, and IBM. He is also responsible for many of the photos and videos that are featured in this book and at <http://www.ibm.com>. Andrew brings over 24 years of data center experience with Fortune 100 companies to the team. He graduated Magna cum Laude, Honors, from the University of Michigan, Ann Arbor. Andrew has also written other XIV Gen3 IBM Redbooks® publications.

**Bert Dufrasne** is an IBM Certified Consulting IT Specialist and Project Leader for IBM System Storage® disk products at the International Technical Support Organization (ITSO), San Jose Center. He has worked at IBM in various IT areas. He has authored many IBM Redbooks publications and has also developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect. He holds a master's degree in Electrical Engineering.

Thanks to the following people for their contributions to this project:

Markus Oscheka and Bob Liu  
**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 customer 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 at:

[ibm.com/redbooks/residencies.html](http://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 found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- Mail your comments to:

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

## Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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



# Encryption overview

With the ever-growing landscape of national, state, and local regulations, industry requirements, and increased security threats, ensuring the protection of an organization's information is a key part of operating a successful business. Businesses today need tools to protect against the known threats and to guard against as yet unknown threats. Effective threat and vulnerability management must be proactive rather than reactive, preventing problems rather than responding to them.

Encrypting “data-at-rest” is a key element when addressing these concerns. Most storage products offer encryption at an additional cost. IBM XIV Storage System and IBM FlashSystem A9000 and IBM FlashSystem A9000R systems include data-at-rest encryption at no charge.

**Note:** XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R systems provide data-at-rest encryption at no charge. However, they require an external key server, such as the IBM Security Key Lifecycle Manager with appropriate licensing.

This chapter gives an overview of encryption.

It covers the following topics:

- ▶ 1.1, “Introduction to data-at-rest encryption” on page 2
- ▶ 1.2, “Threats and security challenges” on page 2
- ▶ 1.3, “Need for encryption” on page 3
- ▶ 1.4, “Encryption concepts” on page 3
- ▶ 1.5, “Encryption challenges” on page 4

## 1.1 Introduction to data-at-rest encryption

Support for data-at-rest encryption provides advantages and has certain characteristics:

- ▶ Future non-destructive hot encryption is applied to the data already stored on the system without data rewrite.
- ▶ It offers flexibility in the business decision-making process with the option to deploy the storage system today and decide to apply encryption later, when the need for encryption arises.

The XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R systems offer *hot encryption*. When encryption is enabled on the systems, all data that is on it is encrypted within minutes, with no performance impact.

The XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption solution requires at least one external key server, such as the IBM Security Key Lifecycle Manager server that is used in the context of this publication. For best data protection, it is better to have more than one key server that is installed, preferably in different locations. The IBM Security Key Lifecycle Manager server does not need to be dedicated to a particular storage system and can be shared across multiple products in the data center.

**Important:** Encryption must be deployed with careful planning and a full understanding of the interaction among the required products to avoid deadlock situations. In addition, compatibility between different operating system types and versions and keyserver types and versions must be considered.

## 1.2 Threats and security challenges

Companies face many threats and security challenges:

- ▶ Increasing number and sophistication of threats. 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 customer 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 be able to 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 employee who inadvertently 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.

## 1.3 Need for encryption

Organizations experience a continual 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 the subsequent demand to simplify the complexity of encryption keys management throughout the lifecycle.

In particular, security exposures occur when data storage devices such as disk drives and IBM MicroLatency® Modules leave the company's premises, which usually happens when a data storage device fails and the IBM Service Support Representative (IBM SSR) replaces it with a new part. Sometimes, data storage devices are replaced proactively and the data can still be accessed. IBM has a procedure to delete all data on those parts; however, this task is no longer under the control of the client. Some clients buy back the drives and destroy them themselves, but this procedure can be expensive. A similar concern is when clients return the whole storage system to IBM. IBM erases all data, but this step is not sufficient for some clients. IBM offers a service that is called IBM Certified Secure Data Overwrite Service to erase all data, with several passes, in compliance with United States Department of Defense regulations (DoD 5220.22-M).

When data on the storage system is encrypted, these concerns become resolved. Without the proper decryption key, the data on the storage device or even on the entire storage system is available only as ciphertext, and is therefore unreadable.

The question of what to encrypt and what to leave in clear text often arises. With overall system performance that is not affected by encryption on IBM FlashSystem A9000, IBM FlashSystem A9000R, and XIV systems, it makes sense to encrypt everything. This is easier than choosing which data falls under 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 important.

**Important:** IBM FlashSystem A9000, IBM FlashSystem A9000R, and XIV systems provide storage device-based encryption for data at rest. If encryption over the network is required, additional encryption services must be investigated and deployed.

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

For more information about IBM security solutions in general, see the IBM security site:

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

## 1.4 Encryption concepts

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, and is therefore unreadable.

Computer technology has enabled increasingly sophisticated encryption algorithms. Working with the U.S. Government National Institute of Standards and Technology (NIST), IBM invented one of the first computer-based algorithms, Data Encryption Standard (DES), in 1974. Today, several widely used encryption algorithms exist, including triple DES (TDES) and Advanced Encryption Standard (AES).

## 1.5 Encryption challenges

Encryption, as described previously, depends on encryption keys. Those keys must be, at the same time, kept secure and available, and responsibilities must be split:

- Key security

To preserve the security of encryption keys, the implementation must be such that no one individual (person or system) has access to all of the information that is required to determine the encryption key. In a system-based solution, the encryption data keys are encrypted with a wrapping key (that is, another key to encrypt and decrypt the data keys). This *wrapped key* method is used with XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R by separating the storage of a wrapped data key that is stored on the data device from the storage of the wrap or unwrap keys within a key server (IBM Security Key Lifecycle Manager).

- 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

XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R offer a split recovery key to get access to data if none of the key servers are available. To prevent one person from gaining access to the data, the handling of a recovery key requires at least two people with the role of Security Administrator, which ensures that one person cannot access the data, and it also ensures separation between the Security Administrator and Storage Administrator roles. XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R also enable operation without a recovery key, but this is not preferable because it puts data at risk if the key servers are no longer accessible.

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

One critical consideration with a key server implementation is that the key server must not depend on any storage system to which it provides keys. The key server must not store any of its code or any information about keys that it manages for storage systems on that storage system.

If this consideration is not account for, it becomes possible to experience *encryption deadlock*, where a key server cannot function because it depends on storage that cannot release data because it needs to communicate with that key server. 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.



# **IBM XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R encryption architecture**

This chapter describes the XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R encryption architecture.

It covers the following topics:

- ▶ 2.1, “Architecture overview” on page 6
- ▶ 2.2, “Encryption architecture” on page 7
- ▶ 2.3, “XIV disk encryption” on page 11
- ▶ 2.4, “IBM FlashSystem A9000 or IBM FlashSystem A9000R encryption” on page 13

## 2.1 Architecture overview

This section reviews elements of the encryption architecture for XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R systems.

### Security Administrator role

User roles for XIV, IBM FlashSystem A9000, and IBM FlashSystem A9000R systems include a *Security Administrator* role. The goal is to split security-related tasks from the Storage Administrator user role. The Security Administrator is the only role with authority to configure and enable encryption. The Security Administrator cannot reach any other menu items, such as Volume view or Create.

### Activating encryption

You can configure an XIV system with self-encrypting drives (SEDs) to enable encryption, and then all data that is stored on the XIV is encrypted. The same applies to the MicroLatency modules and vaulting devices in IBM FlashSystem A9000 and IBM FlashSystem A9000R.

You can activate encryption concurrently with data that is already stored in the storage systems. This capability is referred to as *hot encryption*. When hot encryption is started in an XIV system, data in the flash cache is erased. Therefore, after encryption is finished, the flash cache must start “learning” again. The two solid-state drives (SSDs), known as *vaulting devices*, in any grid controller of IBM FlashSystem A9000 and IBM FlashSystem A9000R systems are used for cache and metadata vaulting and destages the cache data that is stored in the memory and metadata triplicated to three grid controllers.

**Important:** Deactivating encryption cryptographically erases all data on the drives and MicroLatency modules. Therefore, you must back up any data that must be kept, or migrate it to another system, before deactivating encryption on XIV or IBM FlashSystem A9000 and IBM FlashSystem A9000R systems.

### Copy Services function considerations

If volumes on an encrypted XIV or IBM FlashSystem A9000 and IBM FlashSystem A9000R system are mirrored to a non-encrypted system, the data is not encrypted on the target storage. Therefore, it is not secured. Also, if the target XIV or IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption is activated, the data is not encrypted when it is transferred between the two systems unless you take suitable measures to protect data in transit.

### IBM Security Key Lifecycle Manager

To enable encryption, an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system must be configured to communicate with *at least one* key server, such as IBM Security Key Lifecycle Manager server. For redundancy, at least two servers are necessary.

**Important:** The IBM Security Key Lifecycle Manager server can be installed as a virtual machine (VM). In that case, make sure that it does not use the encrypted XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system as a storage device. Such configuration can lead to an encryption deadlock situation, where an IBM Security Key Lifecycle Manager server cannot function because it depends on storage that cannot release data because it needs to communicate with that same server.

After the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system starts, it must be able to communicate with at least one of the IBM Security Key Lifecycle Manager servers to obtain the encryption keys. Communication between the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system and the IBM Security Key Lifecycle Manager server is through a Key Management Interoperability Protocol (KMIP) over Secure Sockets Layer (SSL) protocol. The physical connection between the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system and the key server is through a TCP/IP network, as shown in Figure 2-1.

**Important:** If the IBM Security Key Lifecycle Manager server is not reachable when an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R (with encryption activated) is powering on (or is restarted), the storage system is not accessible to read or write data for the hosts. This is why it is important to have at least two IBM Security Key Lifecycle Manager servers in your IP network.

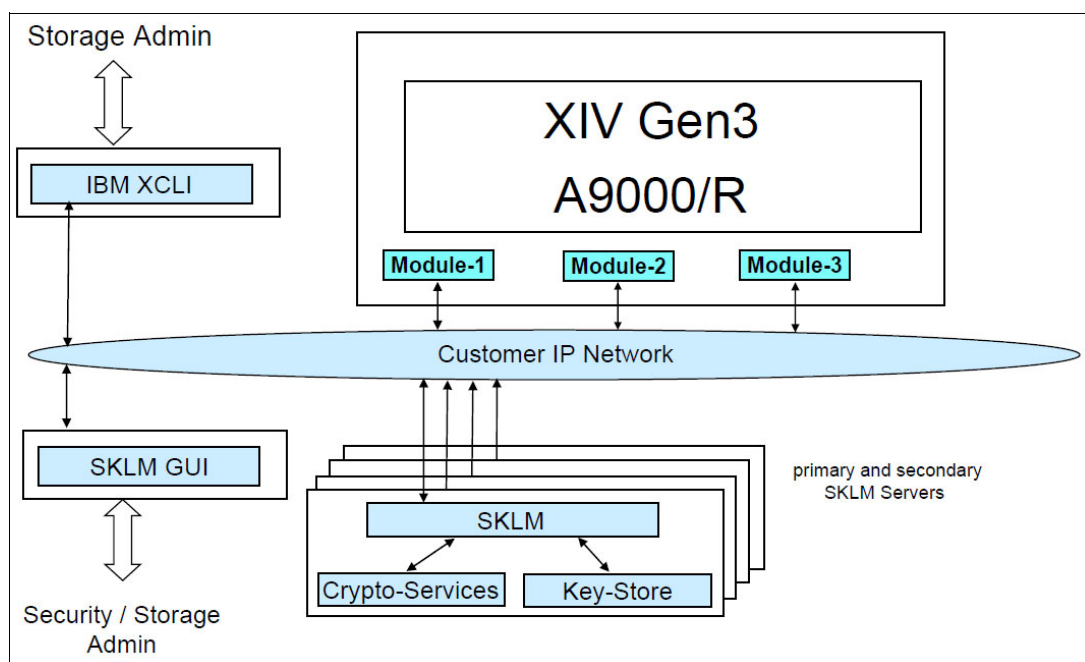


Figure 2-1 XIV and A9000/R connection with IBM Security Key Lifecycle Manager (SKLM)

## 2.2 Encryption architecture

XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R encryption use symmetric key encryption for the data-at-rest solution.

Symmetric key encryption uses the same key to encrypt plain text to ciphertext and to decrypt the ciphertext to regenerate the plain text. This method is called *symmetric encryption* because same key is used for both encryption and decryption.

Anyone who obtains 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. 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 2-2 shows an encryption and decryption data flow path. The symmetric key is used to encrypt a secret file. The decryption of the text uses the same symmetric key to decrypt the data back to readable text.

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

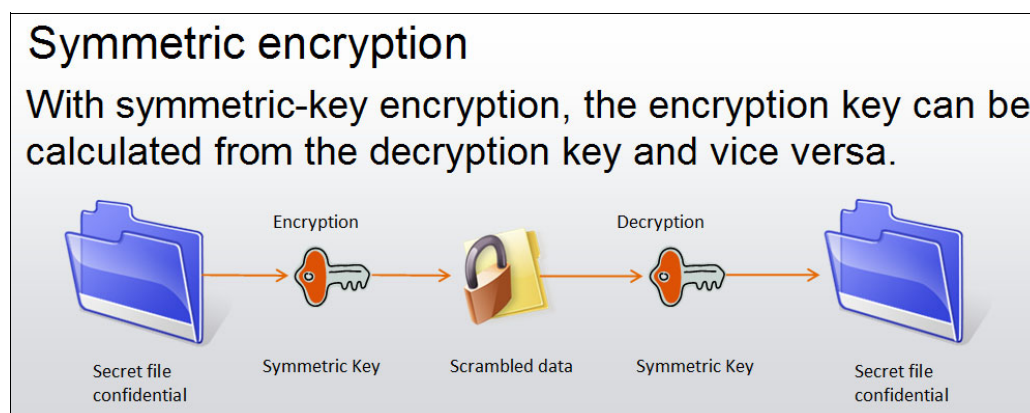


Figure 2-2 Symmetric encryption

## 2.2.1 Configuring encryption

When the user enables encryption, a random Master Key (XMK) is generated for an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system. This key is protected (*wrapped*) by an external key server-provided key (ESK), creating an Encrypted Master Key (EXMK).

Optionally, a manual recovery key (REK) can also be configured and used to wrap the XMK, creating a Recovery Encrypted Master Key (REXMK) and providing the possibility to recover the master key if the key server becomes unavailable.

**Important:** A recovery key can be created only if data-at-rest encryption is not yet enabled. You cannot create a recovery key when encryption is activated.

When you enable encryption on an XIV system, the following keys are generated:

- ▶ For each SED in all modules, a Data Access Key (DAK) is generated that is based on the XMK and the serial number of the XIV module.
- ▶ For the second-level read cache (SSDs), the device access key is generated based on the XMK and the modules serial number.

When you enable encryption on an IBM FlashSystem A9000 or IBM FlashSystem A9000R system, the following keys are generated:

- ▶ For each MicroLatency module in all flash enclosures, a TAK (Access Key) is generated that is based on the XMK and the serial number of the flash enclosure.
- ▶ For the vaulting devices (SSDs), the device access key is generated that is based on the XMK and the grid controllers MegaRaid adapter serial number.

The encryption for the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R can be enabled during the installation of the system or later. If encryption is not enabled, the system might not meet the customer's compliance standards or the legal compliance standards and the data might not be protected against security threats.

Encryption enablement on the XIV system encrypts the SEDs and the cache SSDs. This non-destructive encryption process is applied to the data that is already stored on the system without data rewrite. On IBM FlashSystem A9000 or IBM FlashSystem A9000R systems, encryption enablement encrypts the MicroLatency modules and the SSDs in a hot encryption manner.

**Important:** The encryption for XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R can be disabled only when no pools and volumes are defined.

Therefore, you must back up any data that must be kept, or migrate it to another system, before you deactivate encryption on any of those systems.

## 2.2.2 Digital certificates

*Digital certificates* are a way to bind information with an identity. Digital certificates are exchanged between the IBM Security Key Lifecycle Manager and XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R systems so that each one can verify the other's identity before sending the sensitive keying information. This ensures that the sender can trust the receiver.

Certificates can be signed by a *certificate authority* (CA). If users trust the CA and can verify the CA's signature, they can also verify that certain information belongs to a person or an entity that is identified in the certificate.

These items are part of the information that is stored in a digital certificate:

- ▶ Name of the issuer
- ▶ Subject distinguished name (DN)
- ▶ Public key that belongs to the owner
- ▶ Validity date for the public key
- ▶ Serial number of the digital certificate
- ▶ Digital signature of the issuer

**Digital certificates:** Each XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system has a unique digital default device certificate that is installed at the time of manufacture. In addition, users can install their own digital certificates if they choose. MD5 signature algorithm-based certificates should not be used because of their vulnerability.

## 2.2.3 Encryption techniques

For XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R systems, an HMAC-SHA256 algorithm is used to create a hash message authentication code (HMAC) for corruption detection, as shown in Figure 2-3, and it is additionally protected by a system-generated cyclic redundancy check (CRC).

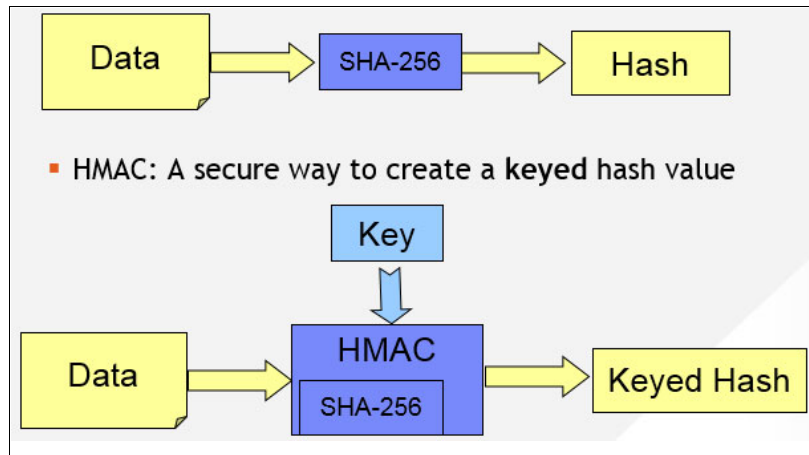


Figure 2-3 HMAC keyed hash creation

Each key defines a one to one mapping to the ciphertext, and without the key, as shown in Figure 2-4, deriving the mapping is unfeasible. Key management is the most complex part of an encryption system.

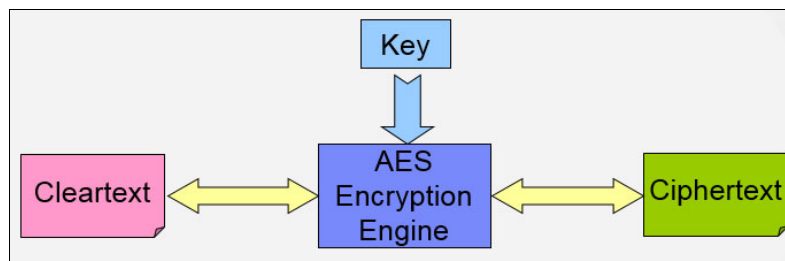


Figure 2-4 Key mapping of ciphertext

Managing keys with IBM Security Key Lifecycle Manager (abbreviated as SKLM in paths and file names) offers production-ready key management. It offers three advantages:

- ▶ Separated, centralized, and simplified key management
- ▶ Separation of key storage from data storage
- ▶ XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R provide KMIP Version 1.2 support

The SED for XIV and the MicroLatency modules and vaulting devices use a symmetric data key to encrypt and decrypt data. The symmetric data key is not available in plain text when the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system communicates with the IBM Security Key Lifecycle Manager. For details, see 2.2, “Encryption architecture” on page 7.

## 2.3 XIV disk encryption

The IBM XIV Storage System Gen3 with software Version 11.4.0 and later for machine types 2810 and 2812 offers a data-at-rest encryption solution that uses SEDs and flexible key manager software. It helps secure data with industry-standard encryption for data at rest, without performance impact. When encryption is enabled, the optional SSDs that are used as flash cache are also encrypted by using software-based AES 256-bit keys encryption. The XIV system secures data at rest and offers a simple, cost-effective solution (cryptographic erasure) for securely erasing any disk drive that is being retired or repurposed.

Encryption support is offered starting with XIV System Software Version 11.4 on XIV Gen 3 Model 214 systems, with 1 TB (stripped-down 2 TB), 2 TB, 3 TB, 4 TB, and 6 TB drive capacities, if those drives are SED. All 4 TB and 6 TB drives available in the XIV system are SED, but not all 1 TB, 2 TB, and 3 TB drives are SED. Any system that is ordered after 8 October 2013, regardless of capacity, has SED. The flash cache encryption is software-based. On an XIV system, upon hot encryption, the flash cache is emptied, and the XIV system must relearn the workload.

This feature is also supported through a system software upgrade to Version 11.4 or higher on all XIV 214 systems that include SED. No additional hardware changes are required to apply data-at-rest encryption functions on those systems. All SED drives can operate transparently in non-encrypting systems.

In software Version 11.4 or higher, by enabling encryption, an XIV system can provide the level of protection that you want without disruption in minutes.

### 2.3.1 Self-encrypting drives

The XIV system supports data encryption with the SEDs. All disks in the XIV system must be the SED type and of the same capacity. No intermixture is allowed. These disks have encryption hardware, and they can encrypt and decrypt data at full disk speed without affecting the performance.

Encryption-capable XIV systems with SEDs can also be used without encryption that is activated. By default, SEDs encrypt data by using a default access key, but because the drive is not enrolled, the key is not protected, and the data remains readable. In this context, *enrolling* means configuring the drive to lock its encryption key with an externally provided key, as described in “Enrolling” on page 12. After a drive becomes enrolled, the access key is locked, and data on the drive is no longer readable without the external key.

## Safe Drive Retirement

With SEDs in the XIV system, Safe Drive Retirement (Figure 2-5) is another feature. When systems are retired, moved, or sold, the keys can be discarded. No data can be read when you use that feature. The IBM XIV is cryptographically erased, as described in 2.1, “Architecture overview” on page 6.

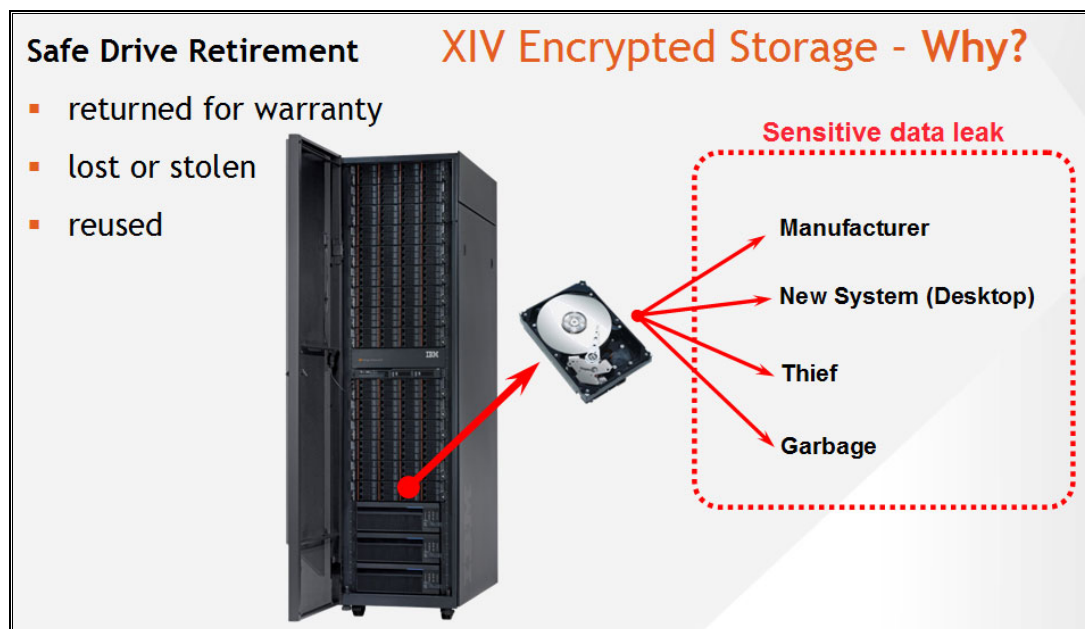


Figure 2-5 Safe Drive Retirement

**Cryptographically erased:** If all copies of the encryption key are lost (whether intentionally or accidentally), it is no longer possible 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 is unfeasible to be decrypted without the key.

## Banding

A *band* is a contiguous region on the disk. *Banding* is the process of defining one or more bands on the drive. Only SED drives can be banded.

In XIV systems, each drive is configured with two bands: One for internal use by the XIV system and one for the user data, which is always encrypted with a drive-unique Data Encryption Key (DEK). That DEK is never accessible from outside the drive.

When a band is defined for user data, a new encryption key is generated and associated with this band. This process effectively and permanently “erases” all data that was previously stored in the band.

## Enrolling

*Enrolling* is a process of instructing the key server to encrypt the key (DEK) that is associated with a specific band. This is accomplished by an externally provided key that is called the Data Access Key (DAK).

The enrolling is performed when the encryption is activated either through the XIV GUI or XIV command-line interface (XCLI).

Before enrollment, the DEK is encrypted with the Manufactured Secure ID (MSID), which is a hardcoded known value in the drive firmware. The MSID is set by the disk manufacturer. It is unchangeable and readable.

*Unenrolling* is instructing the disk to wrap the key with the MSID again, which makes the data readable.

When a disk's band is enrolled, the band becomes unreadable or locked during power-on or reset.

Unlocking a disk requires the same DAK that was used to enroll it. After the DAK is provided, the drive decrypts the DEK and uses it to access the data. The XIV software is responsible for providing each drive with its DAK.

Figure 2-6 illustrates the enrolling process, showing the authentication and encryption key relationship.

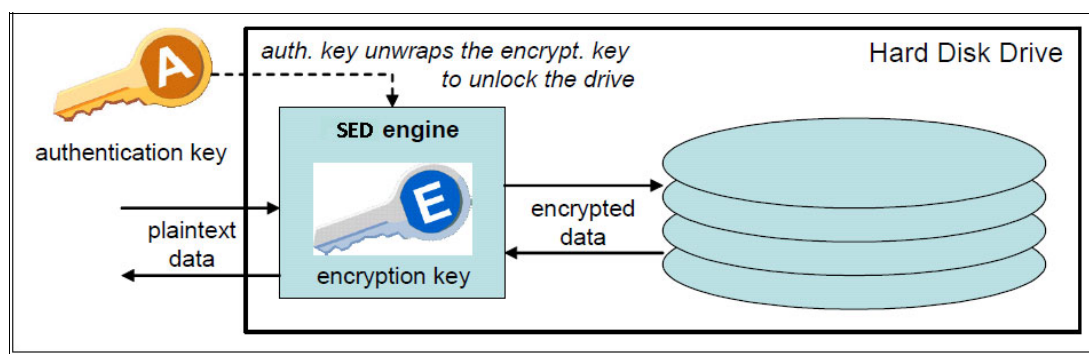


Figure 2-6 Enrolling a SED Hard Disk Drive

## 2.4 IBM FlashSystem A9000 or IBM FlashSystem A9000R encryption

An IBM FlashSystem A9000 or IBM FlashSystem A9000R system offers a data-at-rest encryption solution by using its MicroLatency modules in partnership with an external keyserver.

To provide centralized and simplified key management and the separation of key storage from data storage, the key management is accomplished with IBM Security Key Lifecycle Manager. IBM Security Key Lifecycle Manager offers production-ready key management and complies with the KMIP Version 1.2.

Data-at-rest encryption protects the data that is stored on the grid controller SSD disks and flash enclosure MicroLatency modules against unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording, or destruction. Data-at-rest encryption protects the data if the vaulting SSDs or flash enclosure MicroLatency modules are stolen, improperly discarded, or accessed without authorization. The SSDs in grid controllers act as vaulting devices that hold the destaged cache data and metadata, when an IBM FlashSystem A9000 or IBM FlashSystem A9000R system is shut down.

Encryption support for MicroLatency modules and SSD vaulting devices is offered with IBM FlashSystem A9000 or IBM FlashSystem A9000R systems with software Version 12.0.1.a or greater.

The vaulting SSDs are also securely erased and encrypted after each successful devault. The encryption of system data and metadata is not required, so system data and metadata are not encrypted.

The storage system secures all written data with industry-standard AES-256 encryption for data-at-rest. Encryption is carried out at the hardware level to avoid any performance impact.

Each MicroLatency module has field programmable-gate array (FPGA) control for data-at-rest encryption. Enabling encryption has no performance impact. Data at rest is protected by an Advanced Encryption Standard (XTS-AES) algorithm by using the 256-bit symmetric option in xor-encrypt-xor (XEX)-based tweaked-codebook mode with ciphertext stealing (XTS) mode, as defined in the IEEE1619-2007 standard.

**SEDs:** Certain IBM products, which implement encryption of data at rest, which is stored on a fixed-block storage device, implement encryption by using SEDs. IBM FlashSystem A9000R flash module chips do not use SEDs. The flash module data encryption and decryption are performed by the IBM MicroLatency modules, which can be thought of as the functional equivalent of Self-Encrypting Flash Controller (SEFC) cards.



# Planning

This chapter explains the planning of data-at-rest encryption with an IBM XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system together with IBM Security Key Lifecycle Manager.

It covers these topics:

- ▶ 3.1, “Planning and implementation process flow” on page 16
- ▶ 3.2, “Required and optional tasks” on page 17
- ▶ 3.3, “IBM Security Key Lifecycle Manager licensing” on page 17
- ▶ 3.4, “Preferred practices for encrypting storage environments” on page 19
- ▶ 3.5, “Multiple IBM Security Key Lifecycle Managers for redundancy” on page 22

### 3.1 Planning and implementation process flow

Figure 3-1 shows the planning and implementation process for a data-at-rest encryption-capable XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system. The details for this process are described in subsequent sections of this chapter. Also, see Chapter 4, “Implementing IBM XIV encryption” on page 25 and Chapter 5, “Implementing IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption” on page 59. Figure 3-1 shows the overall decision flow and outcomes.

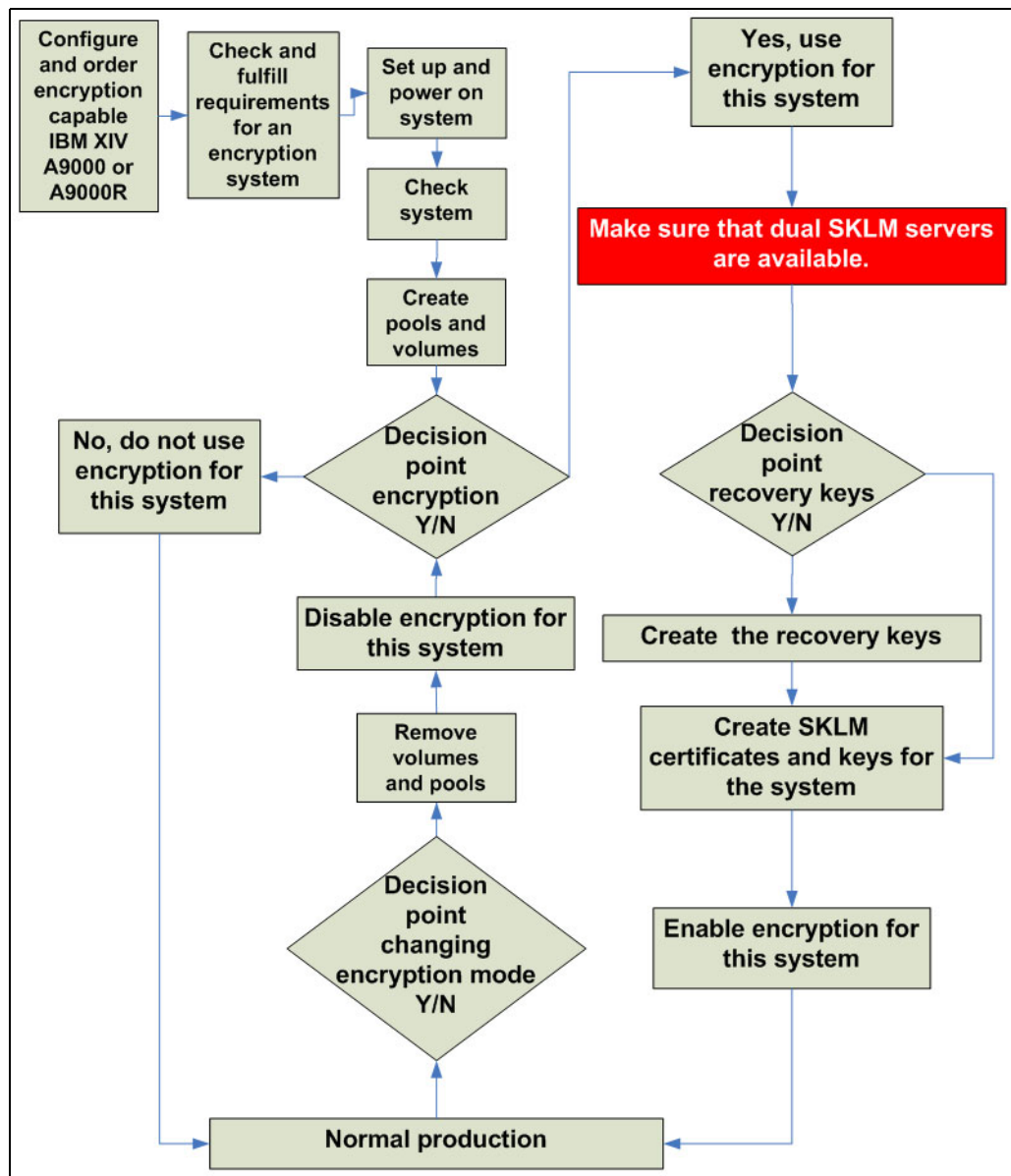


Figure 3-1 Implementation process flow

## 3.2 Required and optional tasks

After the IBM Security Key Lifecycle Manager installation, a few tasks are required to implement and activate encryption on the storage system.

To deploy an encryption-capable XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system, the following requirements must be strictly respected:

- ▶ Configuring the recovery key is the strongly advised first step after the key server is added on the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system. The recovery key is generated from the XIV command-line interface (XCLI). For an XIV system, it can also be created from the XIV GUI.
- ▶ Any XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system that aims to be encryption-enabled must be configured to connect to at least one key server.

The key server can be a separately purchased hardware product that can support the IBM Security Key Lifecycle Manager. Clients must acquire a license for use of the IBM Security Key Lifecycle Manager software that is ordered separately from the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system.

As previously indicated (see “IBM Security Key Lifecycle Manager” on page 6), the IBM Security Key Lifecycle Manager server can be installed as a virtual machine (VM). In this case, make sure that it does not use the encrypted XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system as the storage device.

An encryption-enabled XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system requires at least one key server to be configured, but the preferred practice is to have at least two key servers configured (a primary server and a backup server). A key server can be configured to serve keys to any device that IBM Security Key Lifecycle Manager supports, including other encryption-enabled XIV systems or supported IBM tape drives.

## 3.3 IBM Security Key Lifecycle Manager licensing

IBM Security Key Lifecycle Manager uses Resource Value Units (RVUs) in its licensing for the IBM FlashSystem A9000 or IBM FlashSystem A9000R. However, if the storage system is placed behind an IBM Spectrum Virtualize™ system, such as IBM SAN Volume Controller or IBM Storwize®, such as the V9000, the licensing uses standard Effective Capacity. These options are explored in the following sections.

### 3.3.1 IBM Security Key Lifecycle Manager licensing outside of virtualization

Because XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R offer compression and deduplication features, the licensing model is based on RVUs. These are not linear, so the following tables are useful to order the correct RVU based on the size of the array. For the A9000 (pod) model, the table is shown in Figure 3-2.

FlashSystem A9000	
Modules	RVU
12 x 1.2 TB	15
12 x 2.7 TB	31
12 x 5.7 TB	51

Figure 3-2 IBM FlashSystem A9000 RVU values

However, the much larger IBM FlashSystem A9000R has many other configuration options. Figure 3-3 shows the proper values to be ordered for IBM Security Key Lifecycle Manager for each IBM FlashSystem A9000R.

FlashSystem A9000R With 2.9 TB MicroLatency <sup>®</sup> modules			FlashSystem A9000R: With 5.7 TB MicroLatency <sup>®</sup> modules		
Grid elements	Effective (TB)	RVU	Grid elements	Effective (TB)	RVU
2	300	52	2	600	80
3	450	67	3	900	106
4	600	81	4	1200	131
5	750	94	5	1500	152
6	900	108	6	1800	173

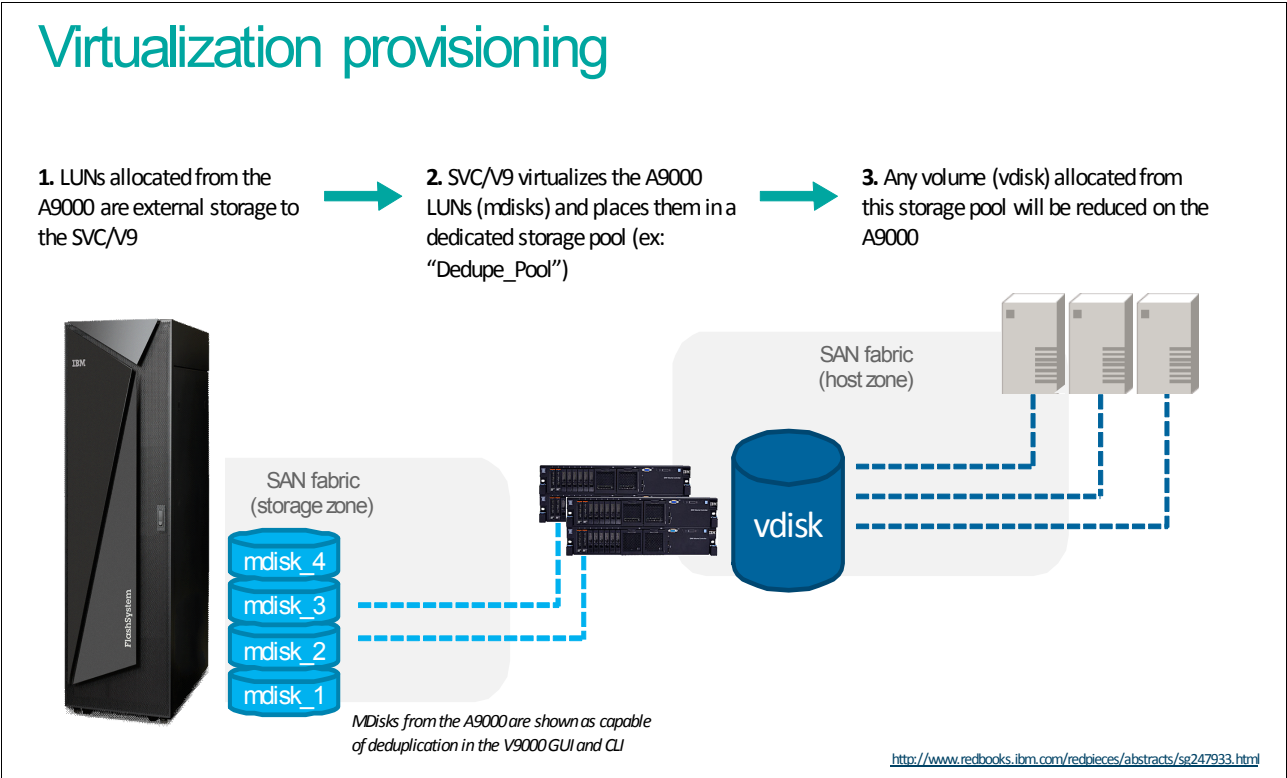
Figure 3-3 IBM FlashSystem A9000R RVU values

Additionally, there are other options that an IBM Sales Executive can use to license based on physical capacity when ordering.

### 3.3.2 IBM Security Key Lifecycle Manager licensing under virtualization

With the explosion of storage virtualization, and the modern ability for a single encrypted volume to span literally multiple storage systems, a difference licensing model is used when IBM Spectrum Virtualize is in front of an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system.

In this virtualized environment, storage functions are typically handled by the highest layer, in this case by IBM Spectrum Virtualize, as shown in Figure 3-4.



As shown in Figure 3-4 on page 18, IBM Spectrum Virtualize (or SAN Volume Controller) has no awareness of data reduction or over-provisioning on the back-end storage that it virtualized.

With IBM Spectrum Virtualize, you can use standard external virtualization licensing per TB (SVC 5641-VC7 or VSC 5608-AE1). However, in special conditions, it might make more sense to license based on Physical Capacity instead of Effective Capacity. Have your IBM Sales Executive contact their Storage Virtualization champion.

## 3.4 Preferred practices for encrypting storage environments

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

### 3.4.1 Security

Here are the considerations and preferred practices:

- General

Ideally, a preferred practice is to manage the physical security of access to hardware through an LDAP implementation. This approach allows close monitoring of who, when, and what actions were taken by monitoring the events of the IBM XIV or Spectrum Virtualize. With a basic security policy, having a single person who handles the *storageadmin* and *secadmin* role of an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system is still possible. With LDAP, you can set up a policy in the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system that does not allow the same user ID for both roles.

- Keystore

During the setup of the IBM Security Key Lifecycle Manager key server, a password is specified for access to the keystore. Clients must decide whether the IBM Security Key Lifecycle Manager password is provided manually or whether a mechanism is in place to automatically provide the password to the IBM Security Key Lifecycle Manager. If a startup script that contains the password is used at the IBM Security Key Lifecycle Manager key server, the script file must have access controls to prevent unauthorized access to the file and password.

### 3.4.2 Availability

Here are the considerations and preferred practices:

- IBM XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system

The XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system must be configured with all management modules or grid controllers IP addresses to provide redundant access to the client's network. An exception is the A9000 (pod) version where only 2 out of 3 management modules (grid controllers) contain management IP addresses.

- ▶ IBM Security Key Lifecycle Manager key server
  - Configure redundant key servers to each encrypting storage device. Have independent and redundant key servers on each site.
  - To initiate the IBM Security Key Lifecycle Manager key server operation after power-on without human intervention, the key server must be set up to automatically power on when power is available and to automatically initiate the key server application. The application must be configured to automatically boot.

### 3.4.3 Encryption administration

Here are the considerations and preferred practices:

- ▶ General:
  - The change management processes at the client installation must cover any procedures that are necessary to ensure adherence to guidelines that are required to ensure the correct configuration of key servers and encrypted storage.
  - All personnel who have any of the following assignments or capabilities are required, at least annually, to review a client document that describes these risks and the processes that are adopted to mitigate them:
    - Responsibility for the implementation of 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 recovery key of the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system, if used.
  - The client must implement automated monitoring of the availability of any equipment that is associated with the management of key services and act appropriately to keep them operational. This equipment can include but is not limited to key servers, SNMP managers, LDAP servers, and domain name servers.
  - Pay particular attention to disaster recovery plans and scenarios, and consider the availability of key servers, key server backups, and key server synchronization. A preferred practice is to establish the independence of each recovery site from the other recovery site.
  - If recovery key management is enabled, the client must have a documented process to handle and maintain the recovery keys of each XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system. This key is the last resort to unlock the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system if the IBM Security Key Lifecycle Manager environment is destroyed or totally inaccessible. The recovery key is *not* used while IBM Security Key Lifecycle Manager remains available.
- ▶ IBM Security Key Lifecycle Manager key server:
  - Configuration of redundant key servers (at least two) is required. Redundancy implies independent servers and independent storage devices.
  - Configuration of one key server with dedicated hardware and non-encrypted storage resources at each recovery site is required.

**Two key servers:** XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R systems require at least one key server to be configured, but a preferred practice is to use two for redundancy.

The following tasks must be accomplished:

- Implementing a key server environment that is independent from non-key server applications so that management of the key server can be restricted to those personnel that are specifically authorized to manage key servers.
  - Implementing a key server that is physically and logically isolated from other applications that might require access to encrypting storage so that the key server environment does not need to be configured with access to any encrypting storage.
  - Implementing a key server that is physically and logically isolated from encrypting storage so that the risk of storing (initially or through data migration) code and data objects that are required by the key server on encrypting storage is eliminated.
  - Ensuring that a recovery site can operate independently of any other sites by configuring a secondary key server that is not dependent on the availability of the primary key server.
- Configuration of additional key servers on generalized server hardware and generalized storage is allowed. Establish appropriate procedures and controls to prevent these key servers from having their data access compromised by storing the data on key server-managed encrypting storage. These key servers are referred to as *general key servers*.
  - Configuration of key servers at independent sites is a preferred practice and 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 and communicate with have a 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.
  - Back up key server data after it is updated. Do not store the backups on encrypted storage media that depend on a key server. For more information, see 6.1, “Automated replication” on page 92.
  - Periodically audit to ensure that all online and backup data that is required to make each key server operational is stored on media that is not dependent on the key server to access the data.
  - Under normal circumstances, clients must not delete keys on the key server. Deletion of all copies of a key is a cryptographic erase operation. It affects all data that is encrypted under this key.

- ▶ XIV and IBM FlashSystem A9000 family:
  - The XIV and IBM FlashSystem A9000 family monitors all configured IBM Security Key Lifecycle Manager key servers. Customer notification is provided through the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R client notification mechanism (SNMP traps, email, Short Message Service (SMS)), or combinations of them, when configured, when a key validation issue with the key servers is detected. 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. Such a condition reflects a degraded key server environment.

The following conditions are monitored and reported:

- If the storage system cannot receive a required data key during power-on from the key servers, it reports the error condition to the client and to IBM. In this case, the associated XIV system that has encryption activated is inaccessible to attached hosts because the storage devices cannot be unlocked during a power-on. If the storage system can obtain the required data key from a key server, after reporting the error, it reports the condition to the client and to IBM, and an XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system reboot is required to make the data accessible.
- The ability of each key server to serve data keys that are configured on the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system is verified at hourly intervals. Loss of the ability to unwrap a configured data key is reported to the client and to IBM.

## 3.5 Multiple IBM Security Key Lifecycle Managers for redundancy

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

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 meet the workload.

This is not a failover or clustered server from an IBM Security Key Lifecycle Manager point of view. The redundancy is managed by setting up multiple key manager destinations at the XIV, IBM FlashSystem A9000, or IBM FlashSystem A9000R system.

Synchronization is achieved through operating system independent UI-based replication. Starting with Version 2.6, IBM Security Key Lifecycle Manager provides a GUI for automated replication configuration. You can configure the replication program to replicate IBM Security Key Lifecycle Manager critical data across clone servers when new keys are added to the master server.

The automated replication process enables cloning of IBM Security Key Lifecycle Manager environments to multiple servers in a manner that is independent of operating systems and directory structure of the server. For example, you can replicate data from a master server on a Windows system to a clone server on a Linux system. You can clone a master IBM Security Key Lifecycle Manager server with up to 20 copies.

For more information, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/admin/cpt/cpt\\_ic\\_admin\\_replication\\_clone\\_master.html](http://www.ibm.com/support/knowledgecenter/en/SSWPVP_2.6.0/com.ibm.sk1m.doc/admin/cpt/cpt_ic_admin_replication_clone_master.html)

On older versions of the IBM Security Key Lifecycle Manager server, back up one server and restore the backup configuration on the other server, assuming that both servers have the same operating system. If you have servers with different operating systems, you must use the export/import function. Plan to perform this backup/restore or export/import operations when the following events take place:

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

### 3.5.1 Setting up IBM Security Key Lifecycle Manager servers

A preferred practice is to complete the pre-installation worksheets that are available in IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/install\\_guide/cpt/cpt\\_insguide\\_worksheets.html](http://www.ibm.com/support/knowledgecenter/SSWPVP_2.6.0/com.ibm.sk1m.doc/install_guide/cpt/cpt_insguide_worksheets.html)

For more information, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/welcome.htm](http://www.ibm.com/support/knowledgecenter/en/SSWPVP_2.6.0/com.ibm.sk1m.doc/welcome.htm)





# Implementing IBM XIV encryption

This chapter describes how to configure and implement data-at-rest encryption for the XIV system.

It covers these topics:

- ▶ 4.1, “Encryption process overview” on page 26
- ▶ 4.2, “IBM Security Key Lifecycle Manager installation” on page 27
- ▶ 4.3, “XIV data-at-rest encryption configuration” on page 28
- ▶ 4.4, “Recovery key use and maintenance” on page 45
- ▶ 4.5, “Activating or deactivating encryption” on page 55
- ▶ 4.6, “Verifying encryption state” on page 57

**Note:** Several illustrations in this chapter are based on Version 2.6 of the IBM Security Key Lifecycle Manager GUI. Older versions, including IBM Tivoli® Key Lifecycle Manager were supported when this paper was written.

For information specific to IBM FlashSystem A9000 and IBM FlashSystem A9000R, go to Chapter 5, “Implementing IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption” on page 59.

## 4.1 Encryption process overview

The XIV data-at-rest encryption initial configuration starts with installing and configuring the external key server. In our testing, we use IBM Security Key Lifecycle Manager Version 2.6.

After the key server is installed and configured, the XIV system and the key server must be able to connect to one another. They establish a trusted connection by exchanging their certificates, as explained in 4.3.1, “Overview of configuration steps” on page 28. Then, the XIV system generates a random XIV master key (XMK), which is used to create the Disk Access Keys (DAKs). Next, the XIV system requests and receives the externally stored key (ESK) from the key server. The ESK is used to wrap (encrypt) the XMK that is stored in the XIV system. Figure 4-1 illustrates the process.

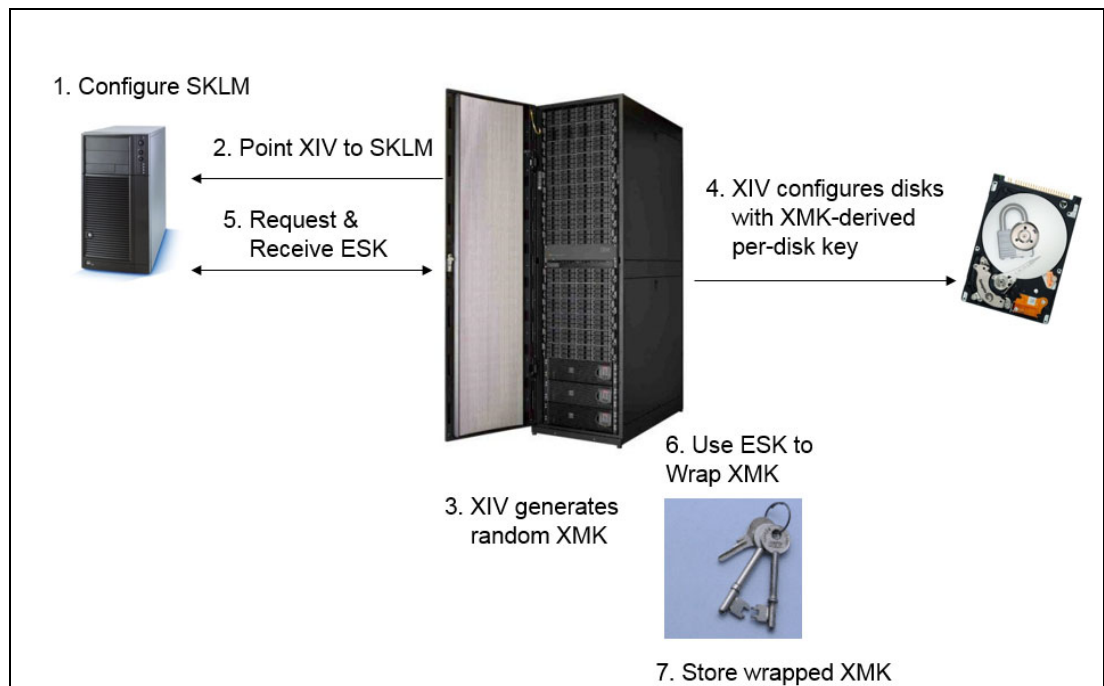


Figure 4-1 Initial configuration

After the XIV data-at-rest encryption is activated, upon booting (after power maintenance, for example), the main encryption startup sequence is as shown in Figure 4-2 on page 27. The self-encrypting drives (SEDs) are locked during a reboot. Therefore, you need a valid connection to the key server.

When the XIV system is booting, it establishes a Secure Sockets Layer (SSL) tunnel that is based on the Key Management Interoperability Protocol (KMIP) if the certificates on the XIV system and on the key server match. The XIV system then requests the ESK, which the key server provides. It is used to unwrap the XMK to derive the DAKs that unlock the SEDs and the encryption-activated flash cache.

**Note:** If there is no valid key server available, the XIV system boots into maintenance mode with no host I/O possible, and all disks are locked. However, a simple XIV system reboot does not lock the disks because they are not power-cycled.

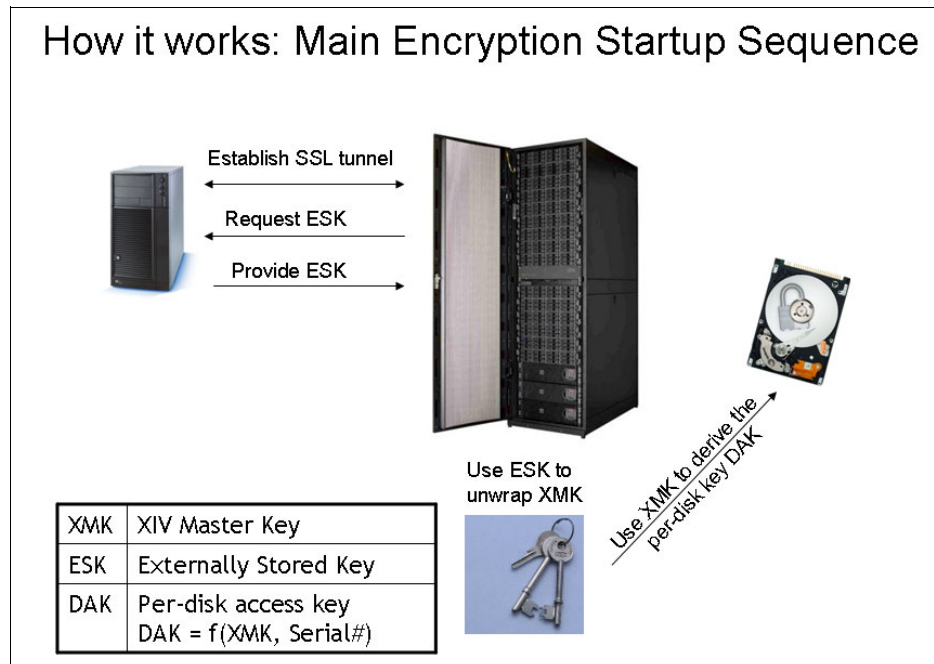


Figure 4-2 Encryption startup sequence

## 4.2 IBM Security Key Lifecycle Manager installation

Starting with the IBM Security Key Lifecycle Manager Version 2.6, the supported operating systems are SUSE Linux, Red Hat Linux, AIX, and Windows.

For specific supported operating system requirements, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/install\\_guide/cpt/cpt\\_ic\\_release\\_oview\\_sw.html?view=embed](http://www.ibm.com/support/knowledgecenter/SSWPVP_2.6.0/com.ibm.sk1m.doc/install_guide/cpt/cpt_ic_release_oview_sw.html?view=embed)

Use the REST interfaces as an alternative to the command-line interface, which might become deprecated in future versions of IBM Security Key Lifecycle Manager.

All references to the alias property of cryptographic keys and certificates in the GUI, CLI, and REST interface will be deprecated in the later versions of IBM Security Key Lifecycle Manager.

You can find installation instructions online in IBM Knowledge Center and chose your desired IBM Security Key Lifecycle Manager version:

[https://www.ibm.com/support/knowledgecenter/en/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/install\\_guide/top/landing-install.html/](https://www.ibm.com/support/knowledgecenter/en/SSWPVP_2.6.0/com.ibm.sk1m.doc/install_guide/top/landing-install.html/)

A collection of IBM Security Key Lifecycle Manager V2.6 PDF documents can be found online at the following website:

<http://www.ibm.com/support/docview.wss?uid=swg27046975>

## 4.3 XIV data-at-rest encryption configuration

This section describes the steps that are required to prepare IBM Security Key Lifecycle Manager to serve an encryption-enabled XIV system. It is based on the assumption that an IBM Security Key Lifecycle Manager server is installed and ready to be configured.

### 4.3.1 Overview of configuration steps

These steps are required to configure IBM Security Key Lifecycle Manager for the XIV system:

1. Create the IBM Security Key Lifecycle Manager master key store.
2. Verify the date and time of the XIV system and the key server.
3. Create securityadmin role users.
4. Manage certificates:
  - a. Copy the XIV device-specific certificate from the XIV system and add the IBM Security Key Lifecycle Manager (import certificate).
  - b. Create an IBM Security Key Lifecycle Manager self-signed certificate on the IBM Security Key Lifecycle Manager GUI (add the KMIP-based SSL certificate).
  - c. Export the IBM Security Key Lifecycle Manager certificate.
5. Define the IBM Security Key Lifecycle Manager key server on the XIV system (import the cert.pem IBM Security Key Lifecycle Manager Certificate).
6. Create the XIV device in the XIV device group (IBM DS5000™ group type).

### 4.3.2 Detailed configuration steps

This section describes the steps to configure and implement an XIV system with IBM Security Key Lifecycle Manager.

The IBM Security Key Lifecycle Manager solution provides simple-to-use installation options and a management console.

#### Step 1: IBM Security Key Lifecycle Manager master key store

When successfully installed, Version 2.6 of the IBM Security Key Lifecycle Manager generates the AES 256-bit master key for data encryption automatically. To conform to the HIPAA and PCI-DSS standards and for increased data security, a 256-bit length master key is used for encrypting IBM Security Key Lifecycle Manager sensitive data, such as key material. The key store holds all keys and certificates that are managed by IBM Security Key Lifecycle Manager.

If this is a IBM Security Key Lifecycle Manager installation earlier than Version 2.6 or the previous Tivoli Key Lifecycle Manager, you must create the master key store manually.

#### Step 2: Verifying the date and time of the XIV system and the key server

When you implement encryption by using keys and certificates, you must have a matching date and time to avoid validity issues, especially when the key server and the system to be encrypted are in different time zones. A mismatch might lead to certificates becoming valid at a future time in the other time zone.

Complete the following steps:

1. Log in to the XIV CLI (XCLI) by using an administrator role-based user and run the command **time\_list**, as shown in Figure 4-3.

```
XIV_ITS0>>time_list
Time      Date      Time Zone  Daylight Saving Time
16:07:38  2016-10-17  UTC       no
```

Figure 4-3 The `time_list` command

2. You can match times between the key server and the XIV system by adjusting the time zone or setting the time manually. If you choose to match the time zone, make sure that you use a valid time zone. Valid time zones can be shown by running the **timezone\_list** command, as shown in Figure 4-4.

```
XIV_ITS0>>timezone_list
Timezone
Universal
Egypt
Africa/Nairobi
Africa/Mbabane
Africa/Niamey
Africa/Dakar
Africa/Khartoum
Africa/Casablanca
...
Europe/Ulyanovsk
NZ-CHAT
EET
GB
GMT
```

Figure 4-4 Excerpt of the `timezone_list` command

3. Run the **timezone\_set** command to adjust the time zone of the XIV system with one of the valid time zones that are listed, as shown in Figure 4-5.

```
XIV_ITS0>>timezone_set timezone=UTC
Command executed successfully.
```

Figure 4-5 The `timezone_set` command

4. If the date and time do not match between the key server and the XIV, you can set it on the XIV system by running the **time\_set** command, as shown in Figure 4-6.

```
XIV_ITS0>>time_set time=2016-10-18.12:02:00
Command executed successfully.
```

Figure 4-6 The `time_set` command

5. Check that your XIV system shows the encryption support in the output of the **state\_list** command, as shown in Figure 4-7.

```
XIV_ITS0>>state_list
Category          Value
system_state      on
target_state      on
safe_mode         no
shutdown_reason   No Shutdown
data_protection_status Fully Protected
encryption      Supported
data_reduction_state Online
```

Figure 4-7 Show encryption support by running the `state_list` command

If the encryption category does not state a supported value, contact IBM Support.

### Step 3: Creating the securityadmin role users

Managing and configuring encryption with an XIV system requires a minimum of two securityadmin role users. To create them, you can either use the XIV GUI, the IBM HSM UI, or the XCLI.

Complete the following steps:

1. When using the XCLI, add the security admin role users by running the **user\_define** command, as shown in Figure 4-8.

```
user_define user=secadmin1 category=securityadmin password=passw0rd
password_verify=passw0rd

user_define user=secadmin2 category=securityadmin password=passw0rd
password_verify=passw0rd
```

Figure 4-8 Define secadmin users in XCLI

2. Run the **user\_list** command to verify that the creation of the secadmin users was successful, as shown in Figure 4-9.

```
XIV_ITS0>>user_list
Name              Category    Group      Active
xiv_development   xiv_development yes
xiv_maintenance   xiv_maintenance yes
admin             storageadmin yes
technician        technician  yes
xiv_hostprofiler   xiv_hostprofiler yes
manager_server_user storageadmin yes
secadmin1          securityadmin yes
secadmin2          securityadmin yes
```

Figure 4-9 The `user_list` command example

You can change certain user's parameters by running the **user\_update** command, as shown in Figure 4-10.

```
XIV_ITS0>>user_update
user=                password=                password_verify=    email_address=
number=              area_code=                exclusive=
```

Figure 4-10 The user\_update command example

**Tip:** After a Security Administrator role base user owns a recovery key, the user name cannot be changed anymore, as shown in Figure 4-11.

```
XIV_ITS0>>user_rename user=secadmin1 new_name=secadmin1b
Error:  USER_OWNS_RECOVERY_KEY
Details: User owns recovery key and therefore cannot be deleted or renamed
```

Figure 4-11 User owns a recovery key

### Step 4: Managing certificates

To manage and configure the IBM Security Key Lifecycle Manager certificates, log in to the XCLI. You must log on to your XIV system as a Security Administrator to complete the following steps:

1. First, copy the XIV device-specific certificate from the XIV system by exporting the certificate, and add it to the IBM Security Key Lifecycle Manager. To get the certificate name that is required in step 2 on page 32, run the **pki\_list** command, which shows the XIV default device-specific certificate that was installed during manufacturing, as shown in Figure 4-12.

```
XIV_ITS0>>pki_list
Name Fingerprint                                Has signed certificate Services
XIV  2c4cd4f951e48d664ebd978357e5a16b yes
                                   XCLI,CIM,IPSEC,KMIP
```

Figure 4-12 The pki\_list command

- Using the name that you get in the output from the command that you run in Figure 4-12 on page 31 (where this example says `<default_certificate>`), run `pki_show_certificate name=<default_certificate>`, as shown in Figure 4-13.

```
XIV_ITS0>>pki_show_certificate name=XIV
Certificate:
...
-----BEGIN CERTIFICATE-----
MIIDVjCCAj6gAwIBAgIEAJiaaDANBgkqhkiG9w0BAQsFADAiMQswCQYDVQQGEwJV
UzETMBEGA1UEChMKaWJtWE1WRG1zazAeFw0xNjEwMTcyMzIzMDVaFw00MTEwMTEy
MzIzMDVaMDkxCzAJBgNVBAYTA1VTMRMwEQYDVQQKEwppYm1YSVZEaXNrMRUwEwYD
VQDEw50DM1LTYwMDMzMTAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQcJEFedWePwcucU1wvAdRZ1eKLXhD1Hg4fKdQ+BspI4NOX7Rpq3Pr0oi2JpRqRW
G4z1rFecRfOqlaE4U1imKOzC4Cx5YpXsEONG2L98aMLP/qGpDQfRhMABOAXTEKum
p8FryHe2MAOrfXZ5EnnDq9YPmvueBbMBHbuxbow3/XKTLFkA3PGNssA+nS2WucfSI
VcgM7kmzpXWpbqDxUOVigcx+BAOHZNha/9Lur8MR7DSmba9mFdNkVpemHg7Scru5
1Z19Xi+7upY9XZaH1eSC13m2pUmv2ZvNdaDoAPF8W5TwgWP9W09rAX0jdAFcZE3B
8bNuI3qfQWQ2w08kuwdzgNznAgMBAAGjFTB7MEoGA1UdIwRDMGAF0ZrcCLEx/xs
trBUvc+gKcAUMIZroSakJDAiMQswCQYDVQQGEwJVUzETMBEGA1UEChMKaWJtWE1W
RG1za4IBADA0BgNVHQ8BAf8EBAMCBaAwHQYDVRO0BBYEF7sJFutxiki1eR3wSyd
SWnn/FMxMA0GCSqGSIb3DQEBCwUAA4IBAQAHC1yYzEkcmo23UWYNQEa7Z1+7WpF
steIgYx9eWHAXiHceo+ItOVT4bovmjH+JgAjWb44Hj7FVkJG14hokDZPmD+bMy+m
y1ahZCcGzOGExaBjj33CT1UixnK2ZyEOEY8SYrDn1fi fGMTAWBCwtTJzeV00cOp5
aBcZn89DohhoURXMqvXvVklQ+Q2LWJ7fV/zrNxgFQHIEbUdpu2PJwRFWqzbbc3Bm
Jou1rZ2F1mYSwrFbremhA54VtjogP35UC/Vb9wAPTYgNxF822vF1mWE4jVYr2xys
UzWUZM8/HdMz1sWycpedhR6xELTZ2HMrq13XJRHSPk9W/R9XY4UHbEb
-----END CERTIFICATE-----
```

Figure 4-13 The `pki_show_certificate` excerpt

- Copy the portion that includes `-----BEGIN CERTIFICATE-----` through `-----END CERTIFICATE-----`, paste it into a blank text editor, and save the file as `<filename>.pem`.  
As you can see, this manufacturing default certificate is readable. After the data-at-rest encryption is activated, the public key is wrapped, and it is encrypted.
- Select **Keep pending client device communication certificates**, which can be set in the Key Serving Parameters tab of the Configuration tab, as shown in Figure 4-14 on page 33.  
When the setting is disabled, client device communication certificates must be manually imported in the wsadmin CLI and do not show up in the Welcome window automatically.

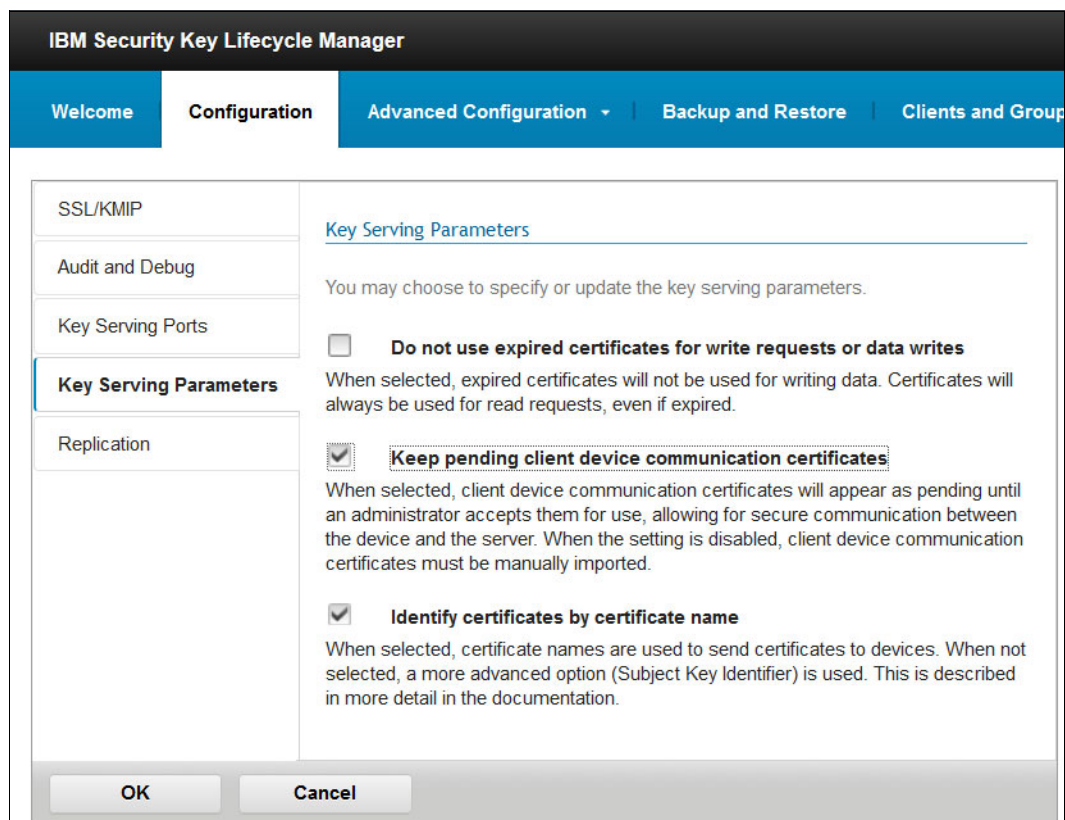


Figure 4-14 Keep pending client device communication certificates

5. In the Welcome window of the IBM Security Key Lifecycle Manager GUI, select the XIV Device Group and then select **Go to...** → **Manage keys and devices**, as shown in Figure 4-15.

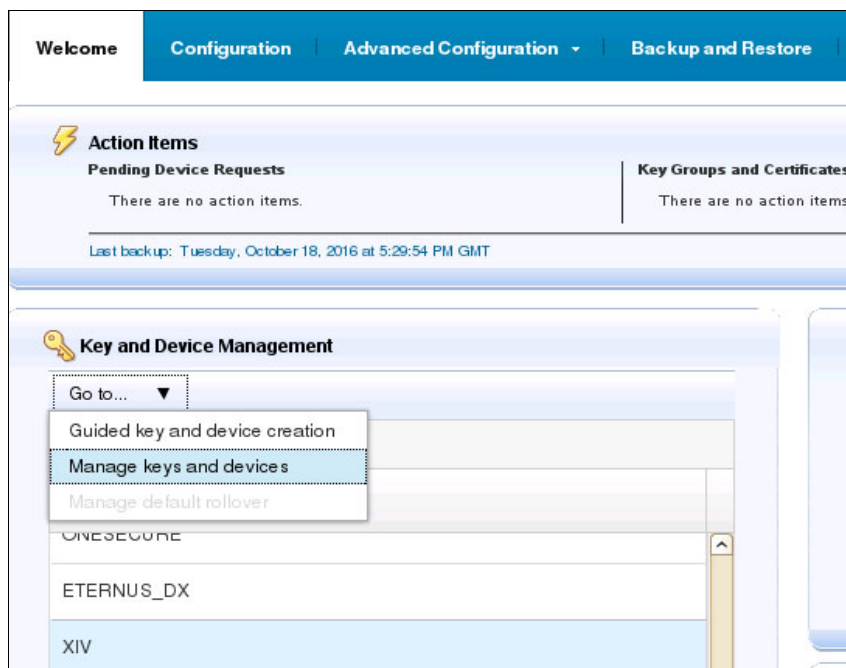


Figure 4-15 Manage keys and devices

Select **Machine affinity** and select **Hold new device requests pending my approval**, as shown in Figure 4-16. This action ensures that when you add the XIV device later that it shows as Pending in the Welcome window.

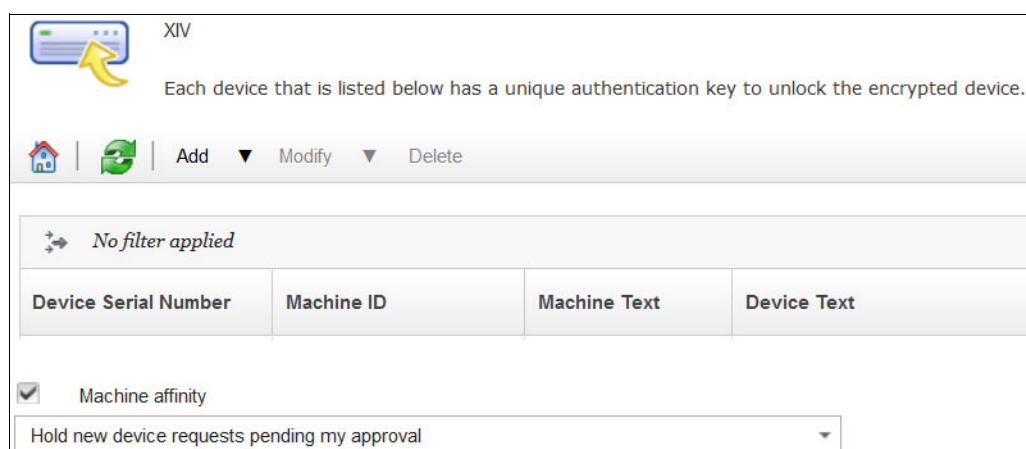


Figure 4-16 Machine affinity and Hold new device requests pending approval

6. Import the newly created SSL certificate as “trusted” in the IBM Security Key Lifecycle Manager web GUI. Click **Advanced Configuration** → **Client Certificates**, and click **Import** (under the SSL/KMIP Certificate for Clients section), as shown in Figure 4-17 and in Figure 4-18 on page 35.

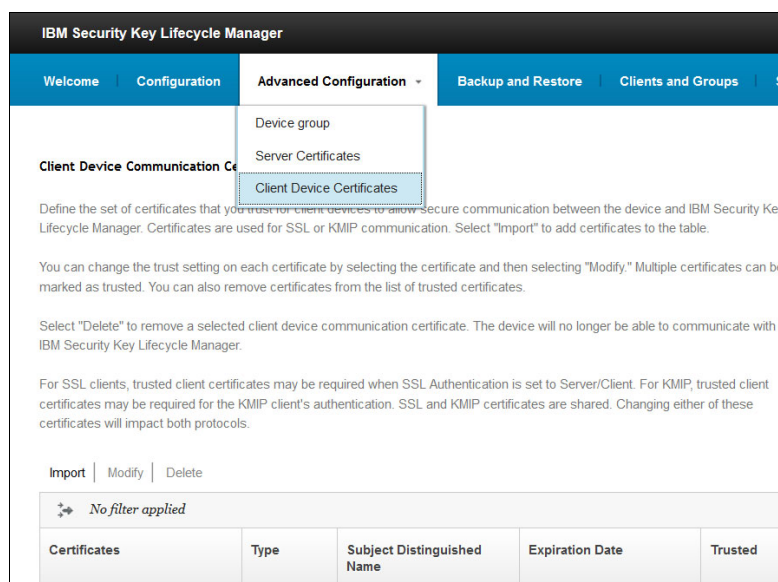
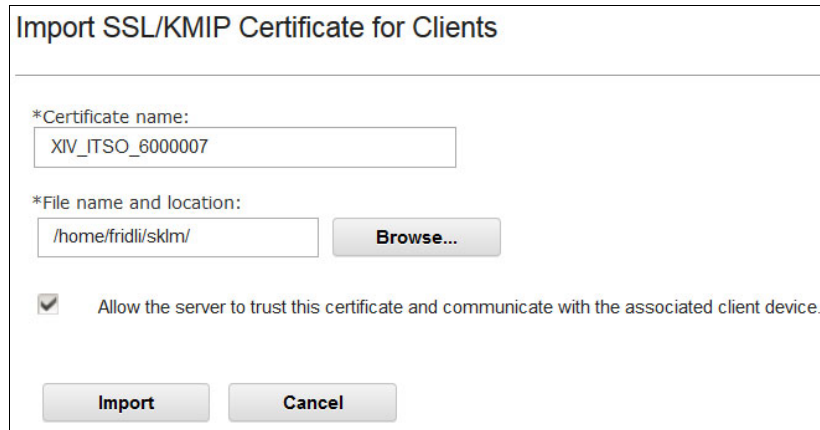


Figure 4-17 Client Device Communication Certificates display



**Import SSL/KMIP Certificate for Clients**

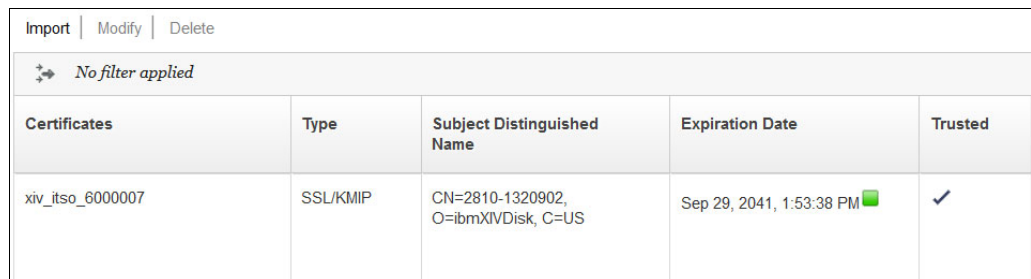
\*Certificate name:

\*File name and location:

☒ Allow the server to trust this certificate and communicate with the associated client device.

Figure 4-18 Import SSL/KMIP Certificate for Clients

The certificate shows as trusted with a type of SSL/KMIP and its name and expiration date are shown, as shown in Figure 4-19.



<div>Import   Modify   Delete</div> <div>  No filter applied         </div>				
Certificates	Type	Subject Distinguished Name	Expiration Date	Trusted
xiv_itso_6000007	SSL/KMIP	CN=2810-1320902, O=ibmXIVDisk, C=US	Sep 29, 2041, 1:53:38 PM	

Figure 4-19 Show the trusted device-specific certificate

- To define the key server in the XIV system, you must create and export the key server certificate on the IBM Security Key Lifecycle Manager and add it to the XIV systems afterward.

Create an IBM Security Key Lifecycle Manager self-signed certificate in the IBM Security Key Lifecycle Manager GUI (add the SSL/KMIP certificate for key serving).

If this is a new key server installation, you must have an Action Item in the IBM Security Key Lifecycle Manager Welcome window to create the server certificate, as shown in Figure 4-20.

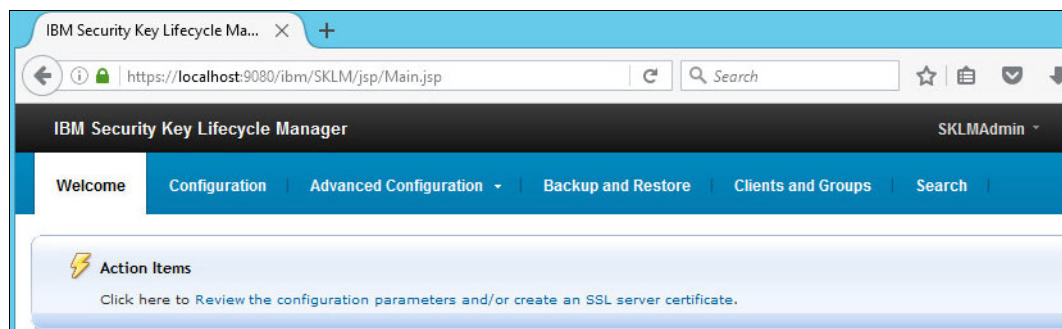


Figure 4-20 IBM Security Key Lifecycle Manager Welcome window with initial configuration Action Item

If the Action Item in the Welcome window is not available anymore, the server certificate can be created by completing the following steps:

- a. In the window that is shown in “Step 2: Verifying the date and time of the XIV system and the key server” on page 28, under Advanced Configuration, click **Create self-signed certification**.
- b. When the window that is shown in Figure 4-21 opens, create the certificate that is used to encrypt data for secure communication over SSL.

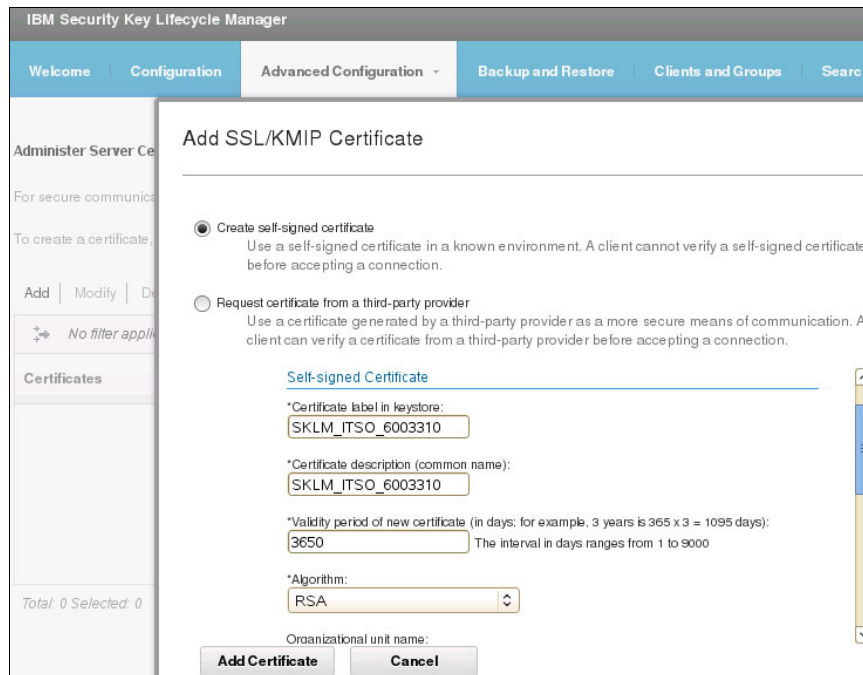


Figure 4-21 Add SSL/KMIP Server Certificate

**Tip:** Do not confuse this certificate with the *device* certificate that is associated with the XIV system.

- c. Select **Create self-signed certificate**. Third-party signed certificates are also supported.

**Caution:** Although using an existing certificate from the key store is possible, using the same certificate to encrypt disk data and protect the communication protocol is *not* recommended.

- d. Choose a descriptive label and a certificate expiration validity in days in accordance with your security guidelines. You also may enter certificate parameters.
- e. Click **Add Certificate**.

As indicated in the Warning notice that is shown in Figure 4-22 on page 37, the SSL/KMIP certificate is updated. For this change to take effect, you must restart the server. Stopping and starting the key server is described in 6.2, “Starting and stopping an IBM Security Key Lifecycle Manager server” on page 93. Also, create a backup to ensure that you can restore this data.

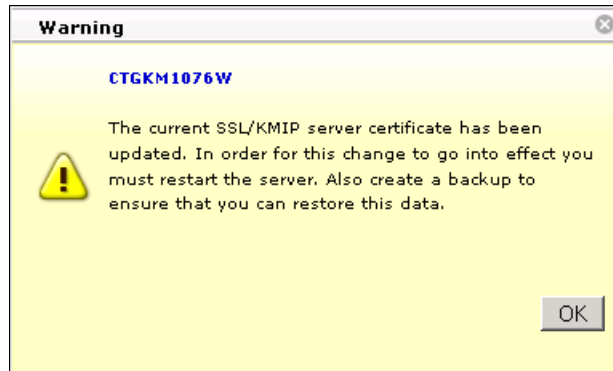


Figure 4-22 Reminder to restart the server

8. In the left pane of the IBM Security Key Lifecycle Manager GUI, click **Welcome** to return to the Welcome window, as shown in Figure 4-23.

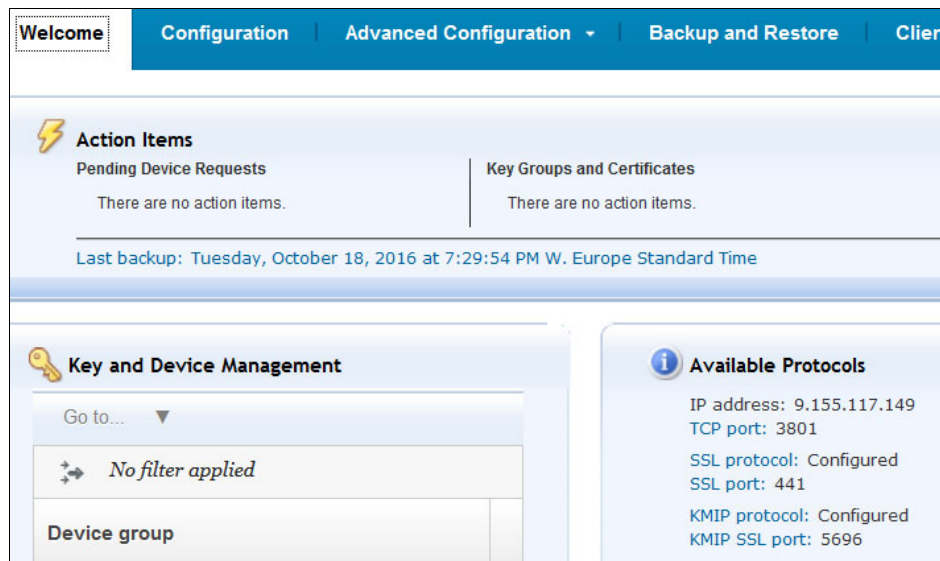


Figure 4-23 Welcome window

You have now created the IBM Security Key Lifecycle Manager master key store and the SSL certificate. As a result, the Available protocols section now has both *SSL protocols* and *KMIP protocols* configured.

After it is created, you must export the certificate. In previous versions earlier than IBM Security Key Lifecycle Manager V2.6, you can do this only by running **wsadmin**.

9. Exporting IBM Security Key Lifecycle Manager server certificates can be done in the IBM Security Key Lifecycle Manager GUI starting with Version 2.6 by completing the following steps:
  - a. Log in to the IBM Security Key Lifecycle Manager server after if restarts and click **Advanced Configuration** → **Server Certificates**, as shown in Figure 4-24.

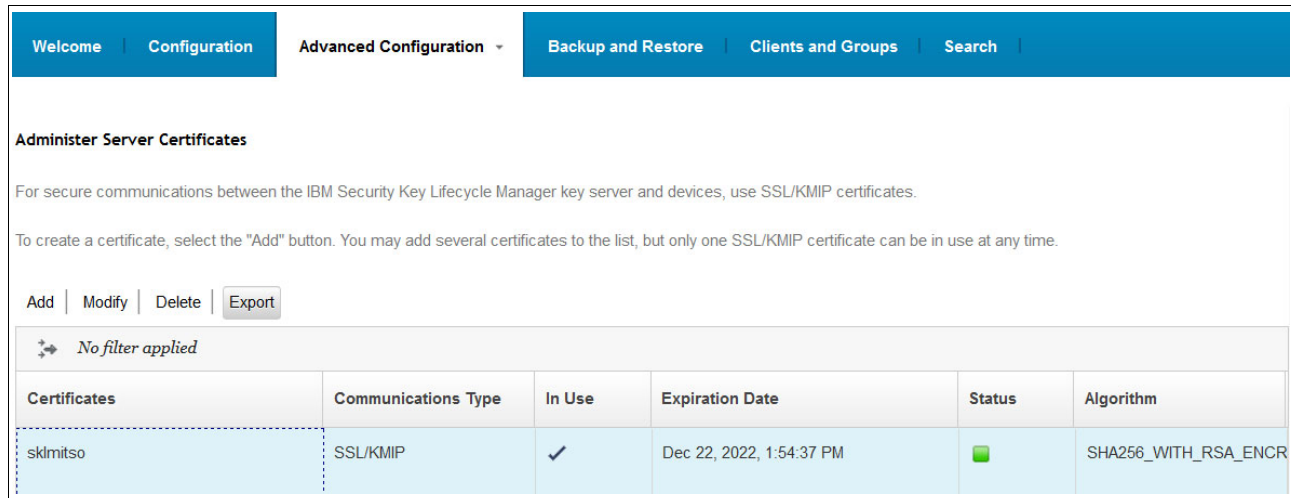


Figure 4-24 Export the server certificate

- b. Select your desired server certificate file name, including the .pem file ending, and browse to the file location where you want it to be saved. Choose certificate type base64 because the DER format is not supported in an XIV system, as shown in Figure 4-25.

### Export Certificate

File name:

File location:

Certificate type:
☒ base64
☐ DER

Figure 4-25 Export certificate type base64

10. For IBM Security Key Lifecycle Manager Version 2.5 and earlier, run **wsadmin** to export the server certificate:

- For a Microsoft Windows operating system, open a DOS prompt with Administrator privileges. Run the following **wsadmin** command:

```
cd <WAS_HOME> wsadmin -username skladmin -password <skladmin password> -lang jython
```

<WAS\_HOME> is, for example, C:\Program Files (x86)\IBM\WebSphere\AppServer.

- For a Linux operating system, open a UNIX terminal session, and run the following command:

```
cd <WAS_HOME>/bin>rm -f ./tmp/cert.der
```

```
./wsadmin.sh -username SKLAdmin -password <skladmin password> -lang jython
```

<WAS\_HOME> is, for example, /opt/IBM/WebSphere/AppServer/bin/.

11. To view all existing certificates, run the **print AdminTask.tklmCertList()** command that is shown in Figure 4-26.

```
wsadmin>print AdminTask.tklmCertList('[-alias XIV_itso_6003310]')
CTGKM0001I Command succeeded.

uuid = CERTIFICATE-23882e94-73b3-4282-b253-185569a76a4d
alias = XIV_itso_6003310
key store name = defaultKeyStore
key state = ACTIVE
issuer name = O=ibmXIVDisk,C=US
subject name = CN=2812-6003310,O=ibmXIVDisk,C=US
creation date = 10/18/16 6:22:45 PM GMT+00:00
expiration date = 10/12/41 1:23:05 AM GMT+00:00
serial number = 10001000
```

Figure 4-26 List all certificates command in wsadmin

12. Print the certificate that is created in step 7 on page 35 by running the **print** command:

```
print AdminTask.tklmCertList('[-alias <label provided in Step 2>]')
```

Figure 4-27 shows an example of the result.

```
wsadmin>print AdminTask.tklmCertList('[-alias XIV_itso_6003310]')
CTGKM0001I Command succeeded.

uuid = CERTIFICATE-23882e94-73b3-4282-b253-185569a76a4d
alias = XIV_itso_6003310
key store name = defaultKeyStore
key state = ACTIVE
issuer name = O=ibmXIVDisk,C=US
subject name = CN=9835-6003310,O=ibmXIVDisk,C=US
creation date = 10/18/16 6:22:45 PM GMT+00:00
expiration date = 10/12/41 1:23:05 AM GMT+00:00
serial number = 10001000
```

Figure 4-27 List a specific certificate in wsadmin

13. Take the Universally Unique Identifier (UUID) information from the output of step 11 on page 39 and use it to export the certification file. You might want to change the **-fileName** option to something other than `/tmp/cert.der` if you want to save it in a different folder.

The specified folder and file name are relative. Therefore, if you specify `/tmp/cert.der`, it is saved in a subdirectory of your IBM Security Key Lifecycle Manager installation directory.

14. Export the certification file by running the following command and format it as base64:

```
print AdminTask.tklmCertExport('[-uuid  
CERTIFICATE-a44aba79-6bcc-47dd-94c0-23ddb5db102c -format base64 -fileName  
/tmp/cert.pem ]')
```

This is a successful output response:

```
CTGKM0001I Command succeeded /tmp/cert.pem
```

This `.pem` file is the certificate that is passed by a parameter in the XCLI **encrypt\_keyserver\_define** command (as described in “Step 5: Defining the key server on the XIV system”).

On a Windows server, you can specify the path as shown in this example:

```
print AdminTask.tklmCertExport  
( '[-uuid CERTIFICATE-a44aba79-6bcc-47dd-94c0-23ddb5db102c  
-format base64 -fileName d:\\mypath\\mycertfilename.pem] ')
```

## Step 5: Defining the key server on the XIV system

You can define a key server on an XIV system by adding the IBM Security Key Lifecycle Manager certificate that you generated and exported to the XIV system.

Complete the following steps:

1. If you are using the XIV GUI, log in as a Security Administrator.
2. Click **Systems** → **System Settings** → **Manage Key Servers** and click the plus sign (+) icon.
3. Enter the name for your key server, the key server IP address or DNS name, and choose the key server certificate that you generated previously, as shown in Figure 4-28.



Figure 4-28 Add Key Server window

4. Click **Create**. The Manage Key Servers dialog in Figure 4-29 indicates that the key server is now trying to establish the connection with the key server.



Figure 4-29 Manage Key Servers

After the connection is established, the Accessible column displays Yes, as shown in Figure 4-30.

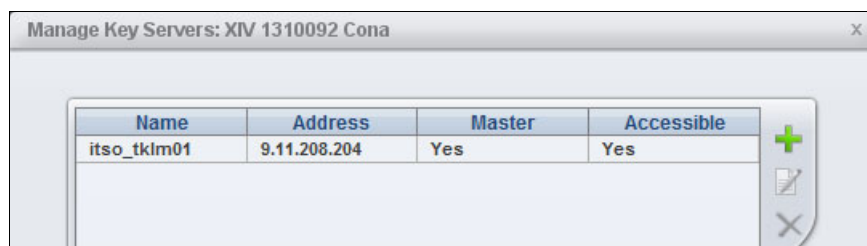


Figure 4-30 Added Key Server

You can also use the XCLI **encrypt\_keyserver\_define** command while logged in to the XIV system as a security admin user to define the key server.

Each operating system has its own way of creating line breaks in a file. Those breaks are not always visible, and if the server certificate is specified to be added as a file name, it can lead to a bad cert status after the key server is added.

One way to eliminate this error is to edit the server certificate in a text editor and create the whole **encrypt\_keyserver\_define** command as one line, including the certificate.

Complete the following steps:

1. At the certificate part of the command, add an asterisk (\*) character at the end of each line, as shown in Figure 4-31.

```
-----BEGIN CERTIFICATE-----*
MIIDVjCCAj6gAwIBAgIEAJiaaDANBgkqhkiG9w0BAQsFADAiMQswCQYDVQQGEwJV
UzETMBEGA1UEChMKaWJtWEIWRG1zazAeFw0xNjAzMjIxNjI2NDJaFw00MTAzMTYx*
NjI2NDJaMDkxCzAJBgNVBAYTA1VTMRMEQYDVQQKEwppYm1YSVZEaXNrMRUwEwYD*
VQQDEwwyODEwLTYwMTM3NzUwggEiMAOGCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB*
AQCF3l+xxvIAv4Kz5yIYNP8IKXL/RlmGcTrJNmGfMzpEYcNwqYMqHzwn1TsXepN9*
vfhI6ZUJSWHH35wemZLm93SseGrZ4fXl0e7k2KLPiggJiQ9p5rPN9WNYViAeCBVY*
QLCowuHBPO7FJqM8TBHJLk49BbxbSJ0cfAP/TyiaCs5HWEYlCwtDqiFRm05WD7l*
ijeMktNLVxgt8hjTF9LeWsY2oF0aNaT7EkMd0IpLwiqeeKqN19X5ovGTBCzo7K5N*
4eDuYtBCL5Ys5HZmEQKMY8eg0LOZLs02ZpSu+PMe0g6cCEBkqjmQehaKvvFkuf37*
3qGroMEANFvGeHj6e/gDKhITAgMBAAGjFTB7MEoGA1UdIwRDMGAF0ZrcCLEX/xs*
trBUvc+gKcAUMIZroSakJDAiMQswCQYDVQQGEwJVUzETMBEGA1UEChMKaWJtWEIw*
RG1za4IBADA0BgNVHQ8BAf8EBAMCBaAwHQYDVRO0BBYEFNOVEsjxUCSOSL+bagge*
bVHZxAJNMAOGCSqGSIb3DQEBCwUAA4IBAQB3VL+g/NICKvSmc4+N9i8TaTe6mL+w*
YQrTvG2MH8bIZEnKyGrddY1QNU8f49ECW7QV/WgbCrYiBRACJiSibiL/7xcDf+G8*
brmUGOPELjEa1h9AKtIMNNft3zUt28VdkuvxQi0telS93DDEtDrcld+e+10eQb5i*
/NnvG1iFH0ZngYBeXCu4B39TDkA4FZ00sb5koj8IVxpzh9WCPZfWfqNUJEgkRX1*
JMs1B0gy4kjsVQvajpH4IIvu8970y5H6heIDnydsR7bsUFGw4edMx823StH1HYJ8*
ODGKj7zxHSrEc7NDHFSYQDnXX3/WmU8y1o1Scntv5dTwmShzzlxcdtbB*
-----END CERTIFICATE-----
```

Figure 4-31 Add asterisks at the end of line of the server certificate

2. Then, remove the line break at the end of the line behind the asterisks so that you end up having the entire command on one line, as shown in Figure 4-32.

```
XIV_ITS0>>encrypt_keyserver_define name=sklm1 ipv4=9.123.123.123 master=yes
certificate="-----BEGIN
CERTIFICATE-----*MIICyTCCAbGgAwIBAgIGKSiyd1FPMAOGCSqGSIb3DQEBCwUAMBQxEjAQBGN
VBAMTCXNrbG1pdHNv*MzAeFw0xNjEwMjEwOTIxMzlaFw0yMjEyMjQwOTIxMzlaMBQxEjAQBGNVBA
MTCXNrbG1pdHNvMzCC*ASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIGeknlONw39+wYw4
ZTtUWYjtNu00eUJlwe0*UgG6pdfZR1N5xj5ijEYENiWMHFxkK06Su87lQLXTEvL3QsQ1xrAmOKMu
it0QxNugguEsLaCim1Nhawv1Da*Rtd1eD1AKeD1AFERReiRaZhaiVnUw8uvURzblOfgg+/iKfnP
eFLBeBqrnkI17FqrdVN0T0bbFxp0V6*DfM49pjG8lM5kAR93R05UysSlz6N0w9srK6eGCfNJGWq
zpsHYK8gBmfxDHco/j/hD4+1TjLbnxna*w8gdQYTFRDkWR4oteFKDBlLM7szIHLZ3VS2CjkYW/VZ
hXGcqzzALy3ZP5uvWFFKWT5TH03I80tJe*xpkCAwEAAAMhMB8wHQYDVRO0BBYEFDUo+63Y6PQCGt
9qM8P60j+cZfC7MAOGCSqGSIb3DQEBCwUA*A4IBAQBwx7WrZGN9E77LcFQmjKwty90y44cPHuf5B
7d064x4q3pZ0qFAuY10BpAExwOghDgW6IeY*nkaWWGfIfTie/NA+U/sP2toVVtR/xuNQSp2WKnxM
nd8ddNosHo2q9Xprx4d2CJS1H7oUj+Maf7OL*nap0baHKyaa3veLH0fxyN/HWFQTF0Kf0qYwq1Cy
px3nLw3XF+a3PKAAZA1hSnMfdNG59RJY+xa4*fzrmAlEvx6iUqZV3jy3Eg2mf3Pam/qP1PbeImz
l4SdJBc+XMFJ2dquidQKedVYymSVy977NF4Tws*erD6HgQHskfr3FEM+b7EfOUPFIbrys8rKtLRb
Wvovebq*-----END CERTIFICATE-----"
```

Figure 4-32 The encrypt\_keyserver\_define one-line command by using the server certificate

3. You can verify that the key server was added successfully by running the `encrypt_keyserver_list` command that is shown in Figure 4-33.

```
XIV_ITS0>>encrypt_keyserver_list
Module Name App/Key Status Last_time_checked Master Port Address Keyserver Type
1 sklm1 NOKEY 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
2 sklm1 NOKEY 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
3 sklm1 NOKEY 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
```

Figure 4-33 The `encrypt_keyserver_list` command

The NOKEY status shows that the XIV has not been accepted on the key server. You see a Pending Devices link in the Welcome window of the key server GUI, as shown in Figure 4-34.

Welcome
Configuration
Advanced Configuration
Backup and Restore
Clients and Groups

Select a device and click "Accept" to allow communication with this device or click "Reject" to remove the request.

Accept | Reject

↔ No filter applied

Date	Device	Device group	Machine
2016-10-21 13:15:46	415-6003310	XIV	0123456789

Total: 1 Selected: 0
1
10 25 50 100

Figure 4-34 Pending Devices in the Welcome window

- Click the **Pending Devices** action item and the XIV that you added appears, as shown in Figure 4-35.

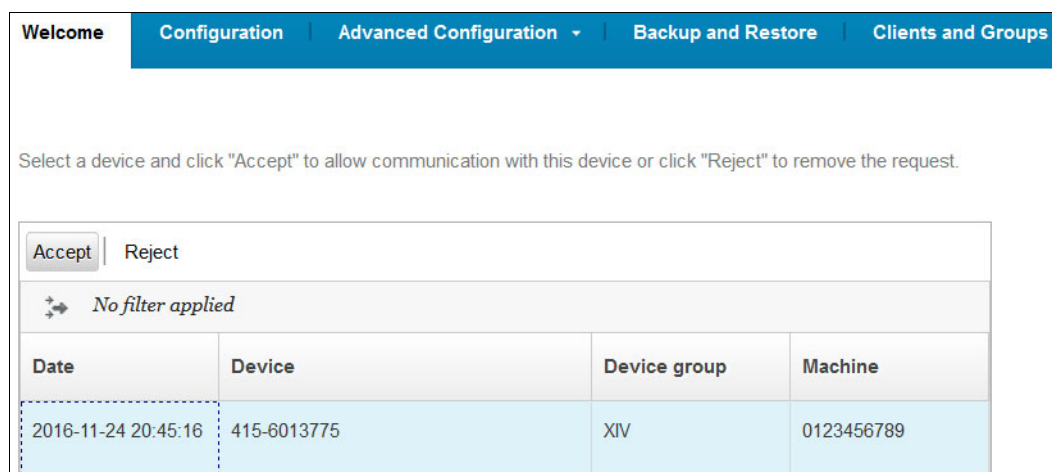


Figure 4-35 Accept the pending XIV device

- Highlight the XIV system and click **Accept** to add it to the key server. If this device served keys in the past from that key server, a warning message, as shown in Figure 4-36, appears and advises you to take a backup before accepting the pending device. This situation can happen during migrations or if encryption was enabled and disabled before. It is a preferred practice to create a backup now to be able to revert to the current state of configuration of the key server.

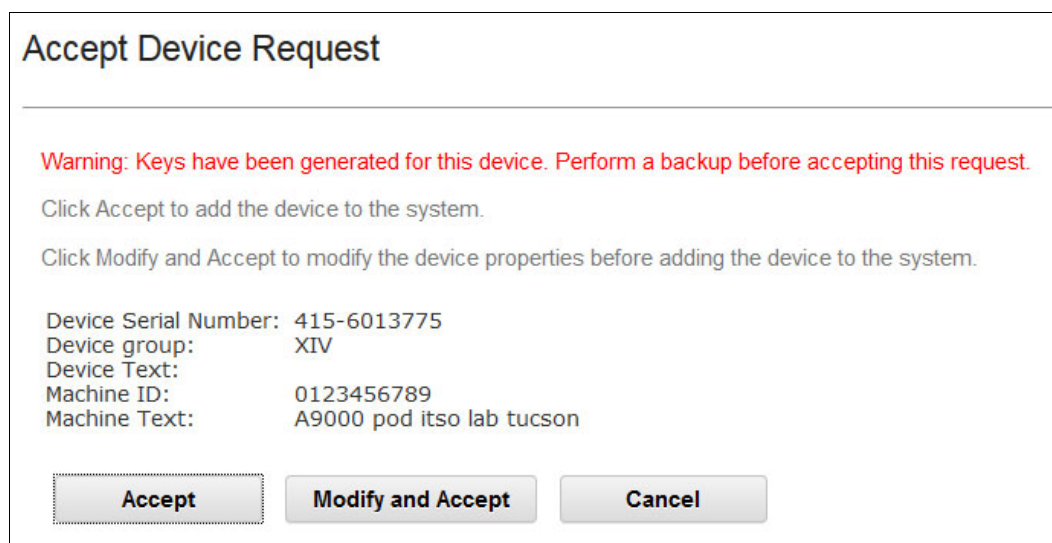


Figure 4-36 Accept Device Request

- After the device request is accepted, it shows as a client device communication certificate in the Advanced Configuration section, as shown in Figure 4-37.

Import   Modify   Delete				
No filter applied				
Certificates	Type	Subject Distinguished Name	Expiration Date	Trusted
xiv_itso_6000007	SSL/KMIP	CN=2810-1320902, O=ibmXIVDisk, C=US	Sep 29, 2041, 1:53:38 PM	

Figure 4-37 Client device communication certificates

- Repeat the same procedure for any additional secondary key servers. If any of the secondary key server states are NOKEY, run **encrypt\_keyserver\_check\_status** and then **encrypt\_keyserver\_list**. If that does not change the NOKEY status, you can replicate the complete configuration of the primary IBM Security Key Lifecycle Manager server to the secondary ones. The same can be accomplished by a manual backup/restore as well.

## 4.4 Recovery key use and maintenance

To protect against the possibility (following a disaster, for example) that all Security Key Lifecycle Managers become unusable and unrecoverable, the XIV system enables you to create a *recovery key*, as shown in Figure 4-38. With a recovery key, Security Administrators can unlock an XIV system without the involvement of a key server.

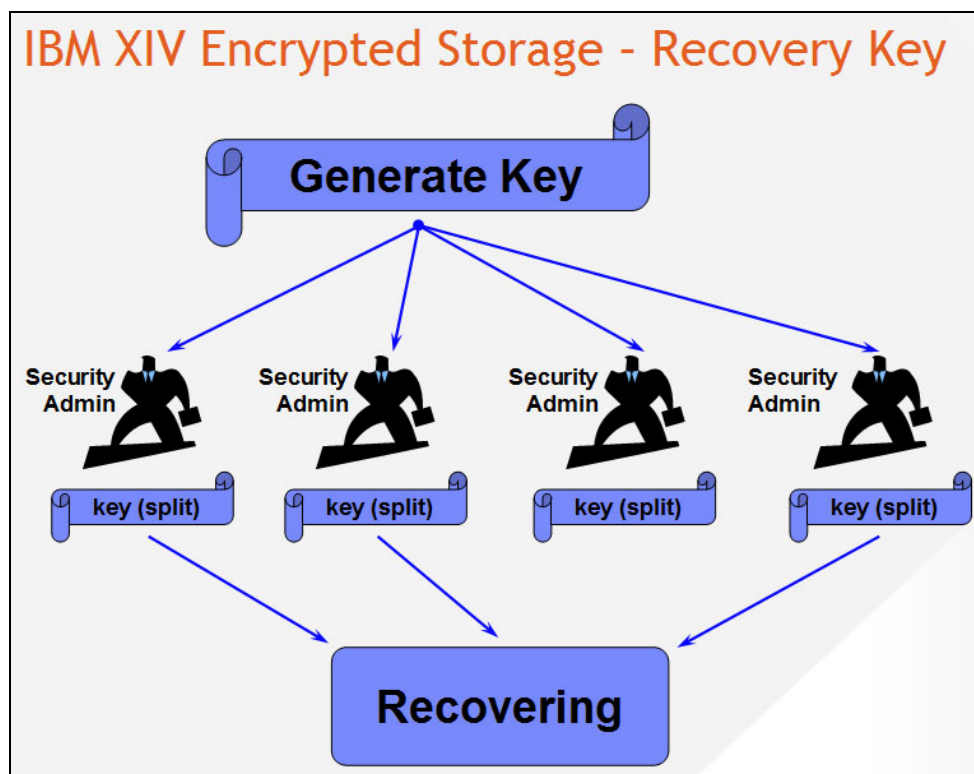


Figure 4-38 Recovery key

Encryption can be activated either in the XIV GUI or through the XCLI. If that action is through the XIV GUI, the recovery key is mandatory. The option to activate data-at-rest encryption without recovery keys is possible, but only through the XCLI by running the **encrypt\_enable** command with the **recovery\_keys=no** flag. The recovery keys are split according to the number of defined Security Administrators and created separately for each Security Administrator.

The recovery key is used to unwrap the XMK, which unlocks the drives.

**Important:** A recovery key can be created only if data-at-rest encryption is not yet enabled. You cannot create a recovery key when XIV encryption is already activated.

Managing the recovery key requires at least two Security Administrators. They maintain the recovery key and keep it safe.

**Client responsibility:** Although an XIV system supports two roles, Storage Administrator and Security Administrator, only the Security Administrator is allowed to use the recovery key. The client is responsible for assigning at least two *separate* individuals as Security Administrators to prevent data access by a single person.

#### 4.4.1 Process for recovery keys

The recovery keys can be generated only if data-at-rest encryption is deactivated on the XIV system. Make sure that at least two XIV Security Administrators are defined on the system. Recovery key creation requires communication with the key server.

The following steps are required to configure recovery keys:

1. Generate recovery keys for each Security Administrator.
2. Get keys for each Security Administrator (make a note of them for later).
3. Verify keys for each Security Administrator to make the keys usable.

The XIV generates a random recovery key and a related wrapping key. The recovery key can also be rekeyed, which generates a new recovery key. That new key must be acquired and verified again by each defined Security Administrator.

#### 4.4.2 Recovery key generation with the XIV GUI

Complete the following steps:

1. In the XIV GUI, log in as a Security Administrator and click **All Systems** → **List**, select your system, and select **Generate Recovery Key**, as shown in Figure 4-39 on page 47.



Figure 4-39 Generate Recovery Key

2. You must choose at least two Security Administrators, add them to the Recovery Key Owners section on the right side, and then click **Start**, as shown in Figure 4-40.

You can add more users who can unlock a locked IBM XIV. If you do that, the least number that you designate in the “Minimum recovery users” field is required to unlock the XIV system. For example, if you define three Security Administrators and add them to Recovery Key Owners, but you select only two as minimum recovery users, only two are necessary to unlock the XIV system, but three of them will be able to do so.



Figure 4-40 Generate Recovery Key pane

3. Now that the recovery key is generated, you can verify whether the process was successful by clicking **Show Results**, as shown in Figure 4-41.

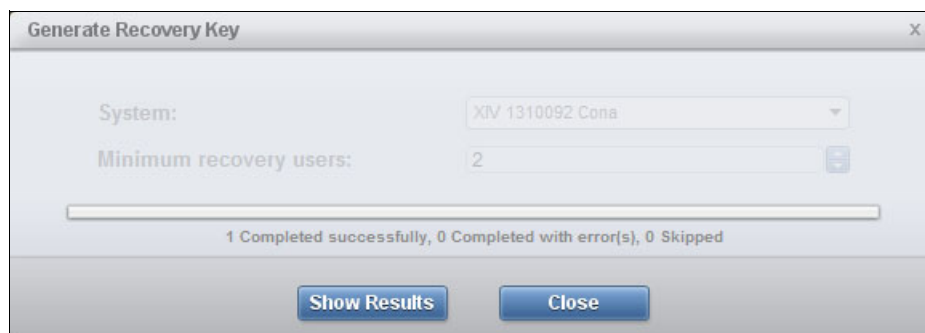


Figure 4-41 Generate recovery keys result

A default text editor opens and shows the Completed Successfully log message that is shown in Figure 4-42.

```
Generating Recovery Keys in 1 systems
-----
[+] | XIV 1310092 Cona - Completed Successfully
```

Figure 4-42 Recovery key generation log

When you close this window, the Generate Recovery Key window that is shown in Figure 4-43 informs you that each of the Security Administrators must log in to acquire the keys that are generated for them.



Figure 4-43 Recovery keys generation completed successfully" message

If the XIV data-at-rest encryption is already enabled, the process continues, but as a result, it completes with an error. When you click **Show Results**, your default text editor opens to display an error message similar to the one in Figure 4-44 on page 49.

```

Generating Recovery Keys in 1 systems
-----
[-] | XIV 1310092 Cona - Completed with Errors -
[-] Failed executing encrypt_recovery_key_generate
    users="itsosecadmin,itsosecadmin2" min_req="2"
[-] reason: Encryption has already been enabled.

```

Figure 4-44 Recovery key generation error log

### 4.4.3 Recovery key generation with XCLI

If you prefer, you can generate the recovery key through the XCLI by completing the following steps:

1. Start with the **encrypt\_recovery\_key\_generate** command that is shown in Table 4-1.

Table 4-1 The *encrypt\_recovery\_key\_generate* command

Category	Command	Description
System	<b>encrypt_recovery_key_generate</b>	Specifies which Security Administrators receive recovery key shares and the minimum number of recovery key shares that must be entered

Here is an example of this command:

```

XIV 1310092>>encrypt_recovery_key_generate min_req=2
users=itsosecadmin,itsosecadmin2,itsosecadmin3
Command executed successfully.

```

2. Each defined Security Administrator must collect and verify the keys that are generated individually by using their credentials to log in to the XCLI, as shown in Table 4-2.

Table 4-2 The *encrypt\_recovery\_key\_get* command

Category	Command	Description
System	<b>encrypt_recovery_key_get</b>	Retrieves the recovery key share that is generated for the current user

Here is an example of this command:

```

XIV 1310092>>encrypt_recovery_key_get
Command executed successfully.
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789

```

3. All defined Security Administrators must verify their keys, as shown in Table 4-3.

Table 4-3 The *encrypt\_recovery\_key\_verify* command

Category	Command	Description
System	<b>encrypt_recovery_key_verify</b>	Confirms that the current user has correctly copied the recovery key share that is presented by <b>encrypt_recovery_key_get</b>

Here is an example of this command:

```
XIV 1310092>>encrypt_recovery_key_verify
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
Command executed successfully.
recovery_status=Key accepted, 1 of 3 fragments have been verified
remaining_fragments=2
```

4. The state of verification can be checked by the **encrypt\_recovery\_key\_status** command that is shown in Table 4-4.

Table 4-4 The *encrypt\_recovery\_key\_status* command

Category	Command	Description
System	<b>encrypt_recovery_key_status</b>	Shows status of recovery keys

Here is an example of this command:

```
XIV 1310092>>encrypt_recovery_key_status
Date Created      User              Status
2013-10-24 11:55:15  itsosecadmin     Verified
2013-10-24 11:55:15  itsosecadmin2    Unverified
2013-10-24 11:55:15  itsosecadmin3    Unverified
```

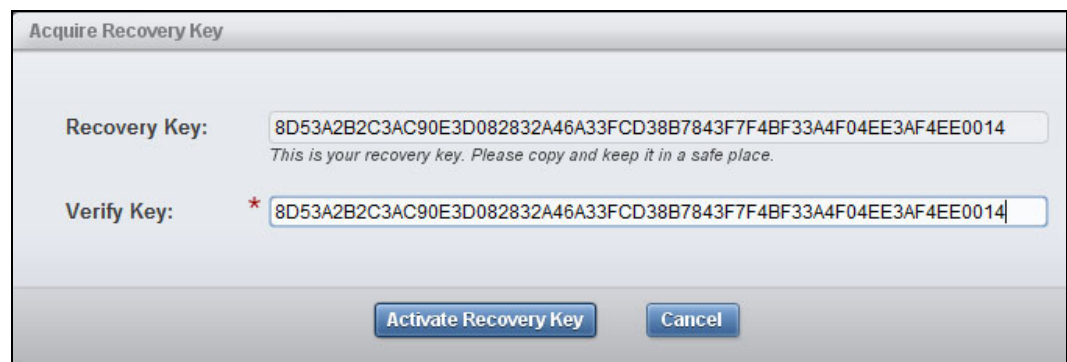
After all defined Security Administrators have collected and verified their keys, the XIV data-at-rest encryption can be activated. For more information, see 4.5.1, “Activating data-at-rest XIV encryption” on page 55.

You can run **encrypt\_recovery\_key\_list** to show the number of shares that have recovery keys and how many of them are required for recovery.

#### 4.4.4 Recovery key verification by using the XIV GUI

Complete the following steps:

1. The recovery key must be acquired by each Security Administrator. The Security Administrators must log in with their own credentials, copy and paste the key into the Verify Key field, and then activate it by clicking **Activate Recovery Key**, as shown in Figure 4-45.



The screenshot shows a window titled "Acquire Recovery Key". Inside, there are two text input fields. The first is labeled "Recovery Key:" and contains the text "8D53A2B2C3AC90E3D082832A46A33FCD38B7843F7F4BF33A4F04EE3AF4EE0014". Below this field is a small italicized note: "This is your recovery key. Please copy and keep it in a safe place." The second field is labeled "Verify Key:" and has a red asterisk icon to its left. It also contains the same alphanumeric string. At the bottom of the window, there are two buttons: "Activate Recovery Key" and "Cancel".

Figure 4-45 Activate Recovery Key

2. An information window is shown indicating that, after verification, you cannot acquire the key again. Save the key in a text file and keep it in a secured place that is physically separate from both the XIV system and the key servers. Click **Continue** to proceed, as shown in Figure 4-46.

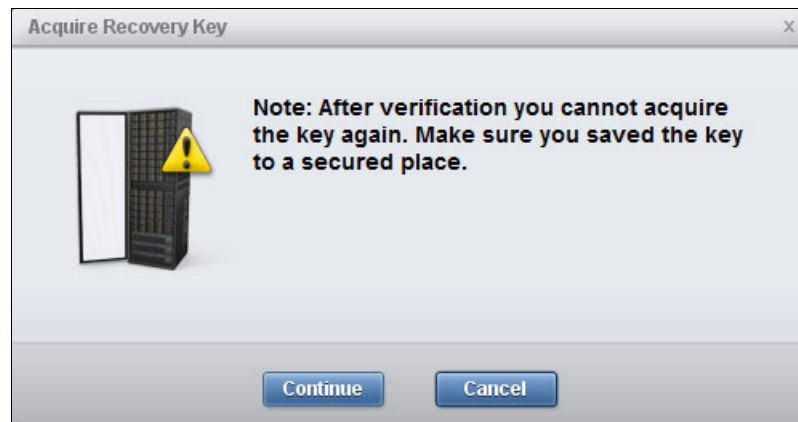


Figure 4-46 Acquire Recovery Key note

If you look at the systems lists through the XIV GUI, the Recovery Key column changes to “1 of 2 acquired”.

Now, the next Security Administrator must log in to the XIV GUI with correct credentials, and then repeat the Activate Recovery Key procedure.

**Tip:** If multiple IBM XIV systems are defined in the XIV GUI, you can decrease the loading time after login by right-clicking the XIV system that you want to access, clicking **Modify IP addresses** without changing any entry, and clicking **Update**.

#### 4.4.5 Recovery key verification by using the XCLI

Complete the following steps:

1. You can use the XCLI to verify the recovery keys that are defined by using the **encrypt\_recovery\_key\_generate** command that is shown in Table 4-3 on page 49.

Table 4-5 The *encrypt\_recovery\_key\_verify* command

Category	Command	Description
System	<b>encrypt_recovery_key_verify</b>	Confirms that the current user has correctly copied the recovery key share that is presented by the <b>encrypt_recovery_key_get</b> command.

Here is an example of this command:

```
XIV_ITS0>>encrypt_recovery_key_verify
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
Command executed successfully.
recovery_status=Key accepted, 1 of 2 fragments have been verified
remaining_fragments=1
```

2. The state of verification can be checked by running the **encrypt\_recovery\_key\_status** command that is shown in Table 4-4 on page 50.

Table 4-6 The *encrypt\_recovery\_key\_status* command

Category	Command	Description
System	<b>encrypt_recovery_key_status</b>	Shows the status of the recovery keys

Here is an example of this command:

```
XIV_ITS0>>encrypt_recovery_key_status
Date Created      User              Status
2016-11-24 11:55:15  itsosecadmin1  Verified
2016-11-24 11:55:15  itsosecadmin2  Unverified
```

You can run **encrypt\_recovery\_key\_list** to show the number of shares that have recovery keys and how many of them are required for recovery.

3. The next Security Administrator must log in to the XCLI with the correct credentials and repeat the Activate Recovery Key procedure.
4. Save the key in a text file and keep it in a secure place that is physically separate from both the XIV system and the IBM Security Key Lifecycle Manager servers.

After all defined Security Administrators have collected and verified their keys, the XIV data-at-rest encryption can be activated. For more information, see 4.5.1, “Activating data-at-rest XIV encryption” on page 55.

## 4.4.6 Recovery key rekey

*Rekeying* is the process of changing cryptographic values in the chain between the key server, recovery key, and DAKs so that the previous value no longer enables access to the system.

The rekey and Verify Recovery Key functions can be performed any time while the recovery key is configured and a key server is available. A key server is required to enable the XIV system to verify that it is in the correct environment.

Only when the key server can decrypt the data key can the XIV system be sure that it is in the same environment. Only then, it generates a new recovery key. For example, on an XIV system that was stolen and put in a separate environment, rekeying the recovery key is not possible.

During a rekeying operation, the following actions are performed:

1. The XIV system sends the ESK to the key server and requests a rekey validation.
2. The key server verifies the identity of the XIV system by using its certificates.
3. The key server signals the XIV system that it can proceed to generate a recovery key.
4. The XIV system generates a recovery key.

Changing the recovery key does not erase the data.

An Unconfigure function of the recovery key is not available after data-at-rest encryption is activated, but you can use the Regenerate Recovery Key function to change your keys.

The recovery key can also be rekeyed to replace the current recovery key with a new one. All defined Security Administrators must collect and verify the new recovery key.

To regenerate the recovery key, you can use either of the following approaches:

- In the XIV GUI, log in as a Security Administrator, and select **All Systems** → **List**, select your system, and select **Re-Generate Recovery Key**, as shown in Figure 4-47.



Figure 4-47 Re-Generate Recovery Key

- The recovery key can also be rekeyed in the XCLI by using the **encrypt\_recovery\_key\_rekey** command that is shown in Table 4-7.

Table 4-7 The *encrypt\_recovery\_key\_rekey* command

Category	Command	Description
System	<b>encrypt_recovery_key_rekey</b>	Restarts the recovery key generation process that is described in <b>encrypt_recovery_key_generate</b> .

Here is an example of this command:

```
XIV 1310092 Cona>>encrypt_recovery_key_rekey
Command executed successfully.
```

#### 4.4.7 Using a recovery key to unlock an XIV system

On an XIV system with a recovery key configured, an option exists to let a Security Administrator enter the recovery key.

After a power-off followed by a power-on action, if the XIV system cannot get the required data key from the master key server, it attempts to contact all other configured key servers to obtain the required key. If that is not successful, the Security Administrator provides the recovery key. The XIV system uses the recovery key to unwrap the XMK that unlocks the drives. After access to data is restored, the XIV system is available to serve host I/O again.

Each Security Administrator enters their individual parts of the recovery key until the number of defined minimum required Security Administrators is reached and it is again possible to unlock the disks. The XCLI command to do so is **encrypt\_recovery\_key\_enter**, as shown in Table 4-8.

Table 4-8 *encrypt\_recovery\_key\_enter*

Category	Command	Description
System	<b>encrypt_recovery_key_enter</b>	Unlocks encrypted disks when the system reboots and cannot access any of the defined key servers if the recovery keys were defined

This example shows the command:

```
encrypt_recovery_key_enter
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
```

As soon as the last one of the minimum number of defined Security Administrators has logged in with credentials and entered a recovery key, the XIV system unlocks and activates the data-at-rest encryption again, as shown in Figure 4-48.

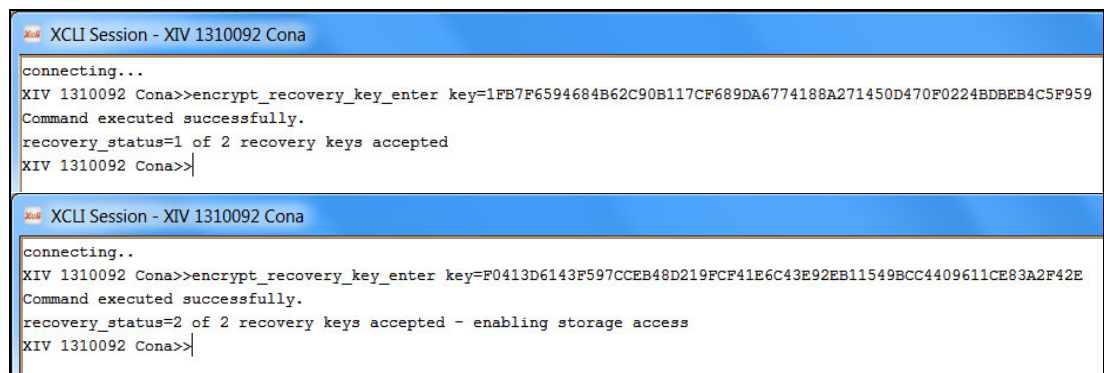


Figure 4-48 *The encrypt\_recovery\_key\_enter command*

**Important:** If your XIV system has a microcode level that is early then 11.5, the **encrypt\_recovery\_finish** command is not available and an IBM SSR must access the system with *technician* authority and change the state of the XIV system from maintenance to on by running a **state\_change target\_state=on** command. A Storage Administrator does not have the authority to run that command.

Now, you can verify by running the `state_list` command again that the `system_state` has changed to `on`, as shown in Figure 4-49. The I/O operation has resumed and the XIV system is serving data to its hosts again.

XIV_ITS0>> <b>state_list</b>	
Category	Value
<b>system_state</b>	<b>on</b>
target_state	on
safe_mode	no
shutdown_reason	No Shutdown
data_protection_status	Fully Protected
encryption	Enabled
data_reduction_state	Online

Figure 4-49 The `state_list` command

## 4.5 Activating or deactivating encryption

Now that the implementation and configuration of the XIV system and its corresponding key server are finished, you can enable (activate) the data-at-rest encryption in the XIV system.

### 4.5.1 Activating data-at-rest XIV encryption

- For data-at-rest encryption to complete successfully, all of these prerequisites must be fulfilled:
- ▶ The current encryption state must be `DISABLED` (displayed as `Supported` by `state_list`).
  - ▶ One master key server must be configured successfully, and recovery keys must be generated and verified by at least two separate Security Administrators, unless a `recovery_keys=no` parameter was passed. This action can be handled either in the XIV GUI or through the XCLI.

In the XIV GUI, log in as Security Administrator, and click **Systems** → **System Settings** → **Activate Encryption**, as shown in Figure 4-50.

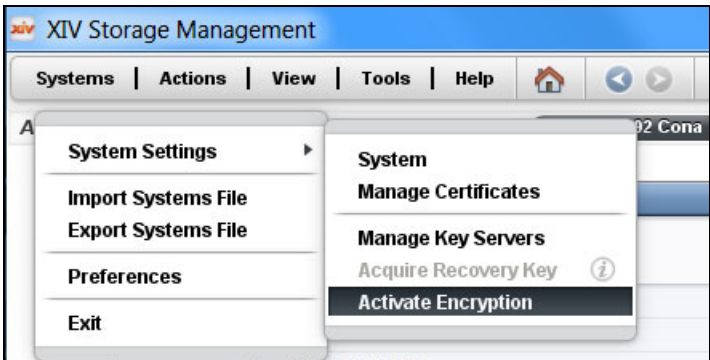


Figure 4-50 Activate data-at-rest XIV encryption

This command is entered by a Security Administrator to enable the data protection feature. Optionally, the data-at-rest encryption can be activated from the XCLI by using the command that is shown in Table 4-9.

Table 4-9 The `encrypt_enable` command

Category	Command	Description
System	<code>encrypt_enable</code>	Enables the data protection feature

This example shows the command:

```
XIV 1310092 Cona>>encrypt_enable
Warning: ARE_YOU_SURE_YOU_WANT_TO_ENABLE_ENCRYPTION y/n: y
Command executed successfully.
```

## 4.5.2 Deactivating XIV data-at-rest encryption

You can disable the data protection feature. A prerequisite is that no volumes are defined on the system. In addition to disabling the data protection, a cryptographic erase is performed on all protected bands to ensure that all existing user data is no longer accessible. After the command completes successfully, all bands are left in an unlocked state. Disabling encryption when the encryption state is other than ACTIVE is an error (`state_list` needs to show it as “Enabled”).

In the XIV GUI, log in as Security Administrator, and click **Systems** → **System Settings** → **Deactivate Encryption**, as shown in Figure 4-51.

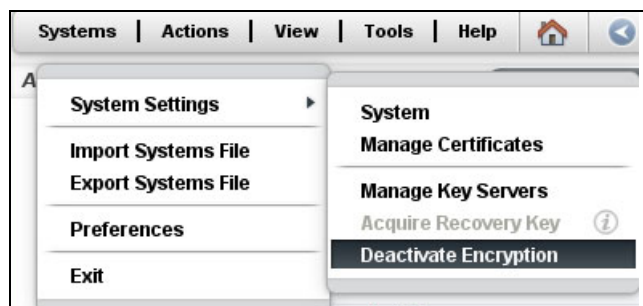


Figure 4-51 Deactivate Encryption

As Figure 4-52 on page 57 shows, the system prompts you to verify that you want to deactivate encryption on the XIV system.



Figure 4-52 Deactivate Encryption verification

## 4.6 Verifying encryption state

These actions enable users to verify the encryption state of the system:

- The encryption state column in the output of the XCLI **state\_list** command shows any of these states: Supported (encryption is disabled), Enabling/Activating, Partial, Enabled/Activated, Enabling on Boot, or Disabling. It is shown as Enabled in Figure 4-53.

XIV_ITS0>> <b>state_list</b>	
Category	Value
system_state	on
target_state	on
safe_mode	no
shutdown_reason	No Shutdown
data_protection_status	Fully Protected
<b>encryption</b>	<b>Enabled</b>
data_reduction_state	Online

Figure 4-53 Encryption state

- The key server state can be checked with the **encrypt\_keyserver\_list** command, as shown in Figure 4-54. The status is checked once every hour or if something has changed or was updated.

XIV_ITS0>> <b>encrypt_keyserver_list</b>										
Module	Name	App/Key	Status	Last_time_checked	Master	Port	Address	Keyserver	Type	
1	sklml	ACTIVE		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM		
2	sklml	ACTIVE		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM		
3	sklml	ACTIVE		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM		

Figure 4-54 Key server state

- Starting with XIV software Version 11.3, in the new encryption state column in the output of the XCLI **disk\_list** command, states can be *Banded* or *Enrolled*.
- The **ssd\_list** command shows the encryption state of the solid-state drives (SSDs) that are used as flash cache in the system. Encryption-related columns are `encryption_state` and `secure_erase_status`.

More data-at-rest encryption-related XCLI commands are listed in IBM Knowledge Center:

<http://www.ibm.com/support/knowledgecenter/api/redirect/ibmxiv/r2/index.jsp>

The *Command-Line Interface (CLI) Reference Guide* includes an “Encryption enablement and support commands” section, and can be downloaded from IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/STJTAG/com.ibm.help.xivgen3.doc/xiv\\_pubsrelatedinfoic.html](http://www.ibm.com/support/knowledgecenter/en/STJTAG/com.ibm.help.xivgen3.doc/xiv_pubsrelatedinfoic.html)



# Implementing IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption

This chapter describes how to configure and implement data-at-rest encryption for the IBM FlashSystem A9000 and IBM FlashSystem A9000R systems.

It covers these topics:

- ▶ 5.1, “Encryption process overview” on page 60
- ▶ 5.2, “IBM Security Key Lifecycle Manager installation” on page 61
- ▶ 5.3, “IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption configuration” on page 62
- ▶ 5.4, “Recovery key use and maintenance” on page 78
- ▶ 5.5, “Activating or deactivating encryption” on page 85
- ▶ 5.6, “Verifying the encryption state” on page 86

**Note:** Several illustrations in this chapter are based on Version 2.6 of the IBM Security Key Lifecycle Manager GUI.

## 5.1 Encryption process overview

The IBM FlashSystem A9000 or IBM FlashSystem A9000R data-at-rest encryption initial configuration starts with installing and configuring the external key server. In our testing, we used IBM Security Key Lifecycle Manager Version 2.6.

After the key server is installed and configured, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and the key server must be able to connect to one another. They establish a trusted connection by exchanging their certificates, as explained in 5.3.1, “Overview of configuration steps” on page 62. Then, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system generates a random master key (XMK), which is used to create Access Keys for MicroLatency modules and vaulting SSDs devices. Next, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system requests and receives the externally stored key (ESK) from the key server. The ESK is used to wrap (encrypt) the XMK that is stored in the IBM FlashSystem A9000 or IBM FlashSystem A9000R system. Figure 5-1 illustrates the process.

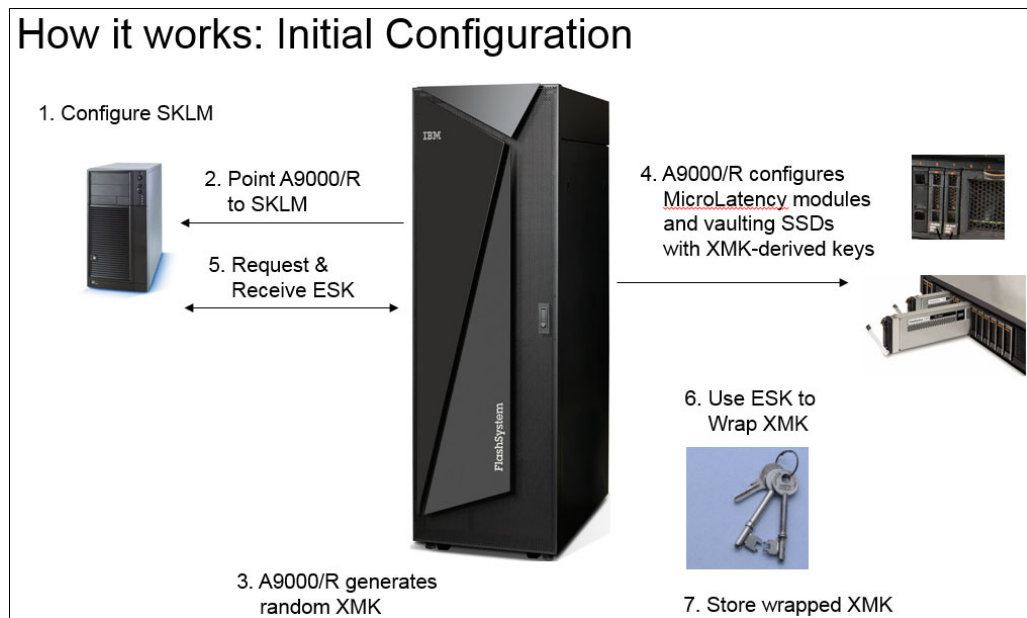


Figure 5-1 Initial configuration

After IBM FlashSystem A9000 or IBM FlashSystem A9000R data-at-rest encryption is activated and upon booting (after power maintenance, for example), the main encryption start sequence is as shown in Figure 5-2 on page 61. The MicroLatency modules and vaulting SSDs are locked during a restart. Therefore, you need a valid connection to the key server.

When the IBM FlashSystem A9000 or IBM FlashSystem A9000R system is booting, it establishes a Secure Sockets Layer (SSL) tunnel based on the Key Management Interoperability Protocol (KMIP) if the certificates on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and on the key server match. The IBM FlashSystem A9000 or IBM FlashSystem A9000R system then requests the ESK, which the key server provides. It is used to unwrap the XMK to derive the Access Keys that unlock the MicroLatency modules.

**Note:** If there is no valid key server available at power-on, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system boots into maintenance mode with no host I/O possible, and all MicroLatency modules are locked. A simple IBM FlashSystem A9000 or IBM FlashSystem A9000R restart also locks the modules.

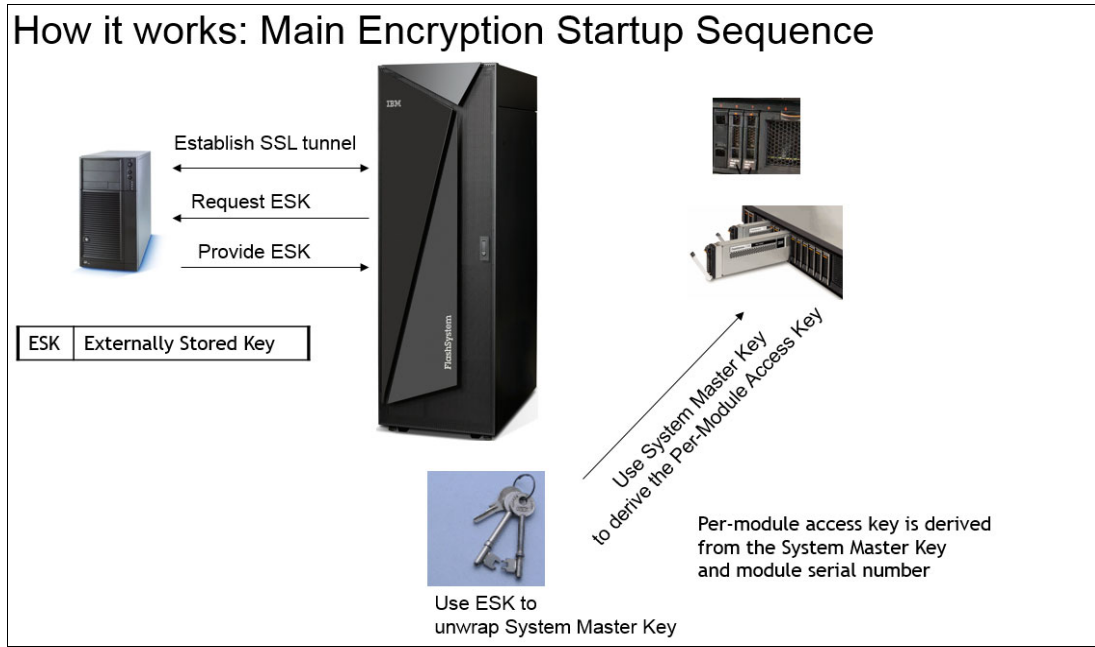


Figure 5-2 Encryption startup sequence

## 5.2 IBM Security Key Lifecycle Manager installation

Starting with IBM Security Key Lifecycle Manager Version 2.6, the supported operating systems are SUSE Linux, Red Hat Linux, AIX, and Windows.

For specific supported operating system requirements, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/install\\_guide/cpt/cpt\\_ic\\_release\\_oview\\_sw.html?view=embed](http://www.ibm.com/support/knowledgecenter/SSWPVP_2.6.0/com.ibm.sk1m.doc/install_guide/cpt/cpt_ic_release_oview_sw.html?view=embed)

Use the REST interfaces as an alternative to the command-line interface (CLI), which might become deprecated in future versions of IBM Security Key Lifecycle Manager.

All references to the alias property of cryptographic keys and certificates in the graphical user interface, command-line interface, and REST interface will be deprecated in the later versions of IBM Security Key Lifecycle Manager.

You can find installation instructions online in the IBM Knowledge Center and chose your desired IBM Security Key Lifecycle Manager version:

[https://www.ibm.com/support/knowledgecenter/en/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/install\\_guide/top/landing-install.html/](https://www.ibm.com/support/knowledgecenter/en/SSWPVP_2.6.0/com.ibm.sk1m.doc/install_guide/top/landing-install.html/)

A collection of IBM Security Key Lifecycle Manager Version 2.6 pdf documents can be found at the following website:

<http://www.ibm.com/support/docview.wss?uid=swg27046975>

## 5.3 IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption configuration

This section describes the steps that are required to prepare IBM Security Key Lifecycle Manager to serve an encryption-enabled IBM FlashSystem A9000 or IBM FlashSystem A9000R system. It is based on the assumption that IBM Security Key Lifecycle Manager server is installed and ready to be configured.

### 5.3.1 Overview of configuration steps

These steps are required to configure IBM Security Key Lifecycle Manager for an IBM FlashSystem A9000 or IBM FlashSystem A9000R system:

1. Define the IBM Security Key Lifecycle Manager master key store.
2. Verify the date and time of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and key server.
3. Create securityadmin role users.
4. Manage certificates:
  - a. Copy the IBM FlashSystem A9000 or IBM FlashSystem A9000R device-specific certificate from the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and add the IBM Security Key Lifecycle Manager (import the certificate).
  - b. Keep pending client device communication, certificates XIV Machine affinity, and hold new device requests pending approval.
  - c. Create an IBM Security Key Lifecycle Manager self-signed certificate on the IBM Security Key Lifecycle Manager GUI (add the KMIP-based SSL server certificate).
  - d. Export the IBM Security Key Lifecycle Manager server certificate.
5. Define the IBM Security Key Lifecycle Manager key server on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system (import the cert.pem IBM Security Key Lifecycle Manager Certificate).
6. Perform recovery key use and maintenance.
7. Activate or deactivate encryption.
8. Verify the encryption state.

### 5.3.2 Detailed configuration steps

This section describes the steps to configure and implement the IBM FlashSystem A9000 or IBM FlashSystem A9000R system with IBM Security Key Lifecycle Manager.

The IBM Security Key Lifecycle Manager solution provides simple-to-use installation options and a management console.

## Step 1: IBM Security Key Lifecycle Manager master key store

Version 2.6 of the IBM Security Key Lifecycle Manager generates the AES 256-bit XMK automatically for data encryption after a successful installation of IBM Security Key Lifecycle Manager. To conform with the HIPAA and PCI-DSS standards and for increased data security, a 256-bit length SMK is used for encrypting IBM Security Key Lifecycle Manager sensitive data, such as key material. This key store holds all keys and certificates that are managed by IBM Security Key Lifecycle Manager. On older key server versions, you might have to manually create the master key store.

## Step 2: Verifying the date and time of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and the key server

When you implement encryption by using keys and certificates, you must have a matching date and time to avoid validity issues, especially when the key server and the system to be encrypted are in different time zones. A mismatch might lead to certificates becoming valid at a future time in the other time zone.

Complete the following steps:

1. Log in to the IBM FlashSystem A9000 or IBM FlashSystem A9000R XCLI by using an administrator role-based user and run `time_list`, as shown in Figure 5-3.

A9000_ITS0>> <code>time_list</code>			
Time	Date	Time Zone	Daylight Saving Time
16:07:38	2016-10-17	UTC	no

Figure 5-3 The `time_list` command

2. You can match times between the key server and the IBM FlashSystem A9000 or IBM FlashSystem A9000R system by adjusting the time zone or setting the time manually. If you choose to adjust the time zone, make sure that you use a valid time zone, or your IBM FlashSystem A9000 or IBM FlashSystem A9000R system might disappear from the HSM UI. If that happens, deleting the IBM FlashSystem A9000 or IBM FlashSystem A9000R system entry in the HSM UI and adding it again brings it back. Valid time zones can be shown by running the `timezone_list` command, as shown in Figure 5-4.

A9000_ITS0>> <code>timezone_list</code>	
Timezone	
Universal	
Egypt	
Africa/Nairobi	
Africa/Mbabane	
Africa/Niamey	
Africa/Dakar	
Africa/Luanda	
...	
Europe/Ulyanovsk	
NZ-CHAT	
EET	
GB	
GMT	

Figure 5-4 Excerpt of the `timezone_list` command

3. Run **timezone\_set** to adjust the time zone of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system to match one of the valid time zones that are produced by the command in Figure 5-4 on page 63, as shown in Figure 5-5.

```
A9000_ITS0>>timezone_set timezone=UTC
Command executed successfully.
```

Figure 5-5 The *timezone\_set* command

4. If the date and time do not match between the key server and the IBM FlashSystem A9000 or IBM FlashSystem A9000R system, you can set it on the A9000/R by running the **time\_set** command, as shown in Figure 5-6.

```
A9000_ITS0>>time_set time=2016-10-18.12:02:00
Command executed successfully.
```

Figure 5-6 The *time\_set* command

5. Also, check whether your IBM FlashSystem A9000 or IBM FlashSystem A9000R system shows the encryption support in the output of the **state\_list** command, as shown in Figure 5-7.

```
A9000_ITS0>>state_list
Category                Value
system_state            on
target_state            on
safe_mode               no
shutdown_reason         No Shutdown
data_protection_status  Fully Protected
encryption            Supported
data_reduction_state    Online
```

Figure 5-7 Show encryption support by running the *state\_list* command

If the encryption category does not state a supported value, contact IBM Support.

### Step 3: Creating securityadmin role users

Managing and configuring encryption with an IBM FlashSystem A9000 or IBM FlashSystem A9000R system requires a minimum of two securityadmin role users. To create them, either use the Hyper-Scale Manager User Interface (HSM UI) or the XCLI.

#### ***Creating the securityadmin role by using the Hyper-Scale Manager User Interface***

To use the HSM UI, complete the following steps:

1. Open a web browser, and specify the IP address and IP port of the HSM by using the following web address format:

```
https://<HSM_IP>:<HSM_PORT>/
```

For example:

```
https://9.123.123.123:8443/
```

2. Enter your IBM FlashSystem A9000 or IBM FlashSystem A9000R system administrator role user ID and password, and then click **Login**.

3. The HSM UI Dashboard indicates that you are successfully logged in to the IBM Hyperscale Manager. If you manage multiple IBM FlashSystem A9000 or IBM FlashSystem A9000R systems, choose the correct one from the list in the Dashboard, as shown in Figure 5-8.

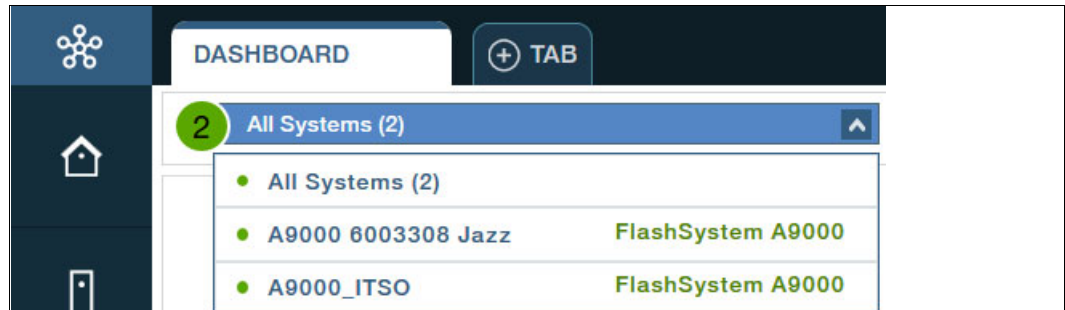


Figure 5-8 Choose your IBM FlashSystem A9000 or IBM FlashSystem A9000R system from the dashboard

4. On the Dashboard's left side, click the **Access Views** icon and then click **Users** to open the Users menu, as shown in Figure 5-9.

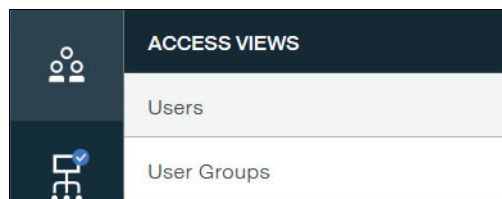


Figure 5-9 Users access view

5. In the Users access view, click the + icon to create an object, as shown in Figure 5-10.

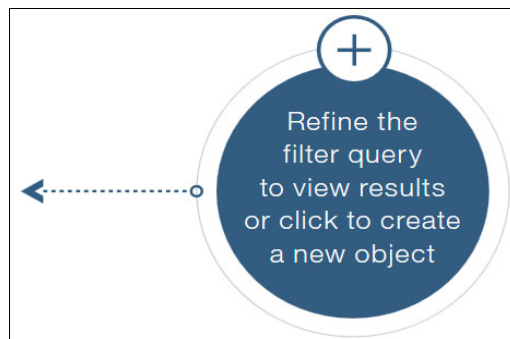


Figure 5-10 Create a user object

6. Enter the name for your security admin user, choose a system and password, change the category to “Security Administrator”, and click **Create**. Phone and Email are optional fields. You cannot choose a domain because Security Admin is associated to the Global Space only. See Figure 5-11.

The screenshot shows a web form for creating a security administrator user. The form is divided into two main sections. The left section contains fields for Name, System, Password, and Phone. The right section contains fields for Category, User Group, Retype Password, and Email. The Name field is filled with 'secadmin1'. The System field is a dropdown menu with a red asterisk icon, showing two options: 'A9000 6003308 Jazz' and 'A9000\_ITSO'. The Password and Retype Password fields are masked with dots. The Category field is a dropdown menu with 'Security Administrator' selected. The User Group field is empty. Below the User Group field, there is a section titled 'USER GROUP DOMAINS'. At the bottom right of the form are 'Cancel' and 'Create' buttons.

Figure 5-11 Create a secadmin user

7. Repeat these steps for further users. The minimum securityadmin users that are required is two.

### ***Creating securityadmin role users by using the XCLI***

You can also use the XCLI to add the securityadmin role users by running the **user\_define** command, as shown in Figure 5-12.

```
user_define user=secadmin1 category=securityadmin password=passw0rd
password_verify=passw0rd

user_define user=secadmin2 category=securityadmin password=passw0rd
password_verify=passw0rd
```

Figure 5-12 Define secadmin users in XCLI

Run the **user\_list** command to verify that the creation of the secadmin users is successful, as shown in Figure 5-13 on page 67.

```
A9000_ITS0>>user_list
Name          Category    Group    Active
xiv_development  xiv_development  yes
xiv_maintenance  xiv_maintenance  yes
admin           storageadmin      yes
technician       technician        yes
xiv_hostprofiler xiv_hostprofiler  yes
manager_server_user storageadmin      yes
secadmin1        securityadmin     yes
secadmin2        securityadmin     yes
```

Figure 5-13 The user\_list command example

You can change certain user's parameters by running the **user\_update** command, as shown in Figure 5-14.

```
A9000_ITS0>>user_update
user=          password=          password_verify=  email_address=
number=        area_code=          exclusive=
```

Figure 5-14 The user\_update command example

**Tip:** After a Security Administrator role base user owns a recovery key, the user name cannot be changed anymore, as shown in Figure 5-15.

```
A9000_ITS0>>user_rename user=secadmin1 new_name=secadmin1b
Error:  USER_OWNS_RECOVERY_KEY
Details: User owns recovery key and therefore cannot be deleted or renamed
```

Figure 5-15 User owns the recovery key

## Step 4: Managing certificates

To manage and configure the IBM Security Key Lifecycle Manager certificates, log in to the XCLI. You must log on to your IBM FlashSystem A9000 or IBM FlashSystem A9000R system as Security Administrator to complete the following tasks:

1. Copy the IBM FlashSystem A9000 or IBM FlashSystem A9000R device-specific certificate from the IBM FlashSystem A9000 or IBM FlashSystem A9000R system by exporting the certificate, and add it to the IBM Security Key Lifecycle Manager. To get the certificate name that is required in step 2 on page 68, run the **pki\_list** command, which shows the IBM FlashSystem A9000 or IBM FlashSystem A9000R default device-specific certificate that was installed during manufacturing, as shown in Figure 5-16.

```
A9000_ITS0>>pki_list
Name Fingerprint          Has signed certificate Services
XIV  2c4cd4f951e48d664ebd978357e5a16b yes
                                           XCLI,CIM,IPSEC,KMIP
```

Figure 5-16 The pki\_list command

2. Using the name that you get from the output (this example says *<default certificate>*), run `pki_show_certificate name=<default_certificate>`, as shown in Figure 5-17.

```
A9000_ITS0>>pki_show_certificate name=XIV
Certificate:
...
-----BEGIN CERTIFICATE-----
MIIDVjCCAj6gAwIBAgIEAJiaaDANBgkqhkiG9w0BAQsFADAiMQswCQYDVQQGEwJV
UzETMBEGA1UEChMKaWJtWE1WRG1zazAeFw0xNjEwMTcyMzIzMDVaFw00MTEwMTEy
MzIzMDVaMDkxOzAjbG9uYVYTA1VTMRMwEQYDVQKEwppYm1YSVZEaXNrMRUwEwYD
VQDEw5ODM1LTlYwMDMzMTAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQCEJEFedWePwcucU1wvAdRZ1eKLXhD1Hg4fKdQ+BspI4NOX7Rpq3Pr0oi2JpRqRW
G4z1rFecRfOqlaE4U1imKOzC4Cx5YpXsEONG2L98aMLP/qGpDQfRhMABOAXTEKum
p8FryHe2MAOrfXZ5EnnDq9YPmvueBbMBHbuxboW3/XKTLfKA3PGNssA+nS2WucfSI
VcgM7kmzpXWpbqDxU0Vigcx+BAOHZNha/9Lur8MR7DSmba9mFdNkVpemHg7Scru5
1Z19Xi+7upY9XZaH1eSC13m2pUmv2ZvNdaDoAPF8W5TwgWP9W09rAX0jdAFcZE3B
8bNuI3qfQWQ2w08kuwdzgNznAgMBAAGjFTB7MEoGA1UdIwRDMGAF0ZrcCLEx/xs
trBUvc+gKcAUMIZroSakJDAiMQswCQYDVQQGEwJVUzETMBEGA1UEChMKaWJtWE1W
RG1za4IBADA0BgNVHQ8BAf8EBAMCBaAwHQYDVRO0BBYEFL7sJFutxiki1eR3wSyd
SWnn/FMxMA0GCSqGSIb3DQEBCwUAA4IBAQAHC1yYYzEkcmo23UWYNQEa7Z1+7WpF
steIgYx9eWHAXiHceo+ItOVT4bovmjH+JgAjWb44Hj7FVkJG14hokDZPmD+bMy+m
y1ahZCcGzOGExaBjj33CT1UixnK2ZyEOEY8SYrDn1fiFGMTAWBCWtTJzeV00cOp5
aBcZn89DohhoURXMqvXvKlQ+Q2LWJ7fV/zrNxgFQHIEbUdpu2PJwRFWqzbbc3Bm
Jou1rZ2F1mYSwrFbremhA54VtjogP35UC/Vb9wAPTYgNxF822vF1mWE4jVYr2xys
UzWUZM8/HdMz1sWycpcedhR6xELTZ2HMrq13XJRHSPk9W/R9XY4UHbEb
-----END CERTIFICATE-----
```

Figure 5-17 The `pki_show_certificate` excerpt

3. Copy the portion that includes `-----BEGIN CERTIFICATE-----` through `-----END CERTIFICATE-----`, paste it into a blank text editor, and save the file as *<filename>.pem*.  
As you can see, this manufacturing default certificate is readable. After the data-at-rest encryption is activated, the public key is wrapped, and it is encrypted.
4. Select **Keep pending client device communication certificates**, which can be set in the Key Serving Parameters tab of the Configuration tab, as shown in Figure 5-18 on page 69.  
When the setting is disabled, client device communication certificates must be manually imported in the wsadmin CLI and do not show up in the Welcome window automatically.

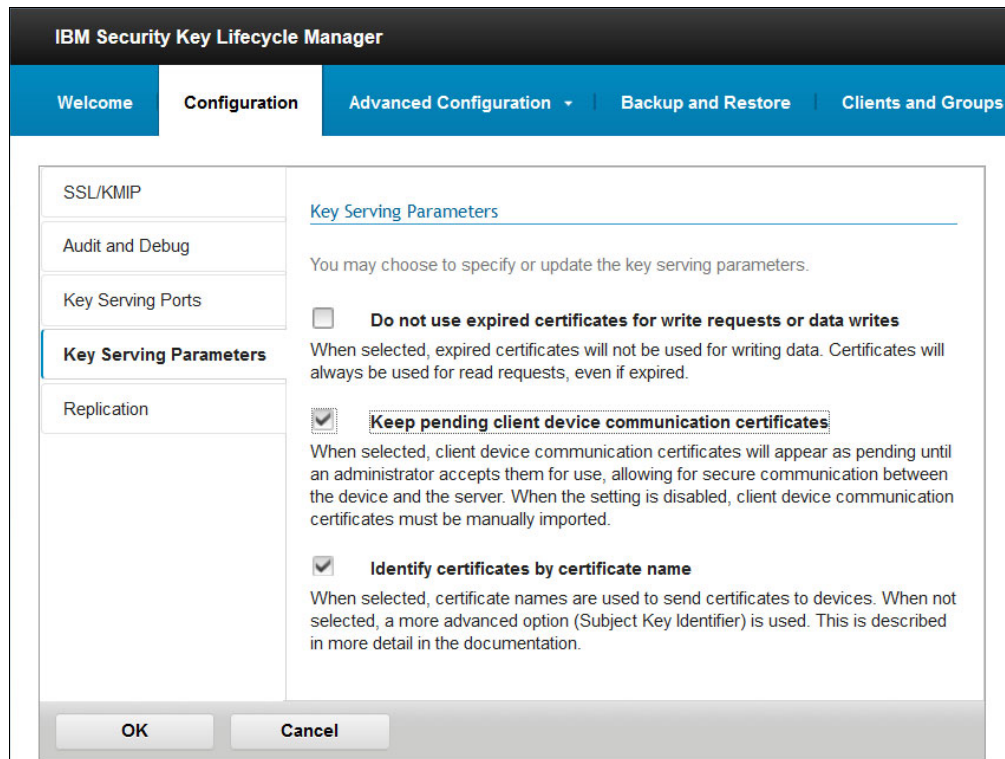


Figure 5-18 Keep pending client device communication certificates

5. In the Welcome window of the IBM Security Key Lifecycle Manager GUI, select the IBM FlashSystem A9000 or IBM FlashSystem A9000R system Device Group and then select **Go to → Manage keys and devices**, as shown in Figure 5-19.

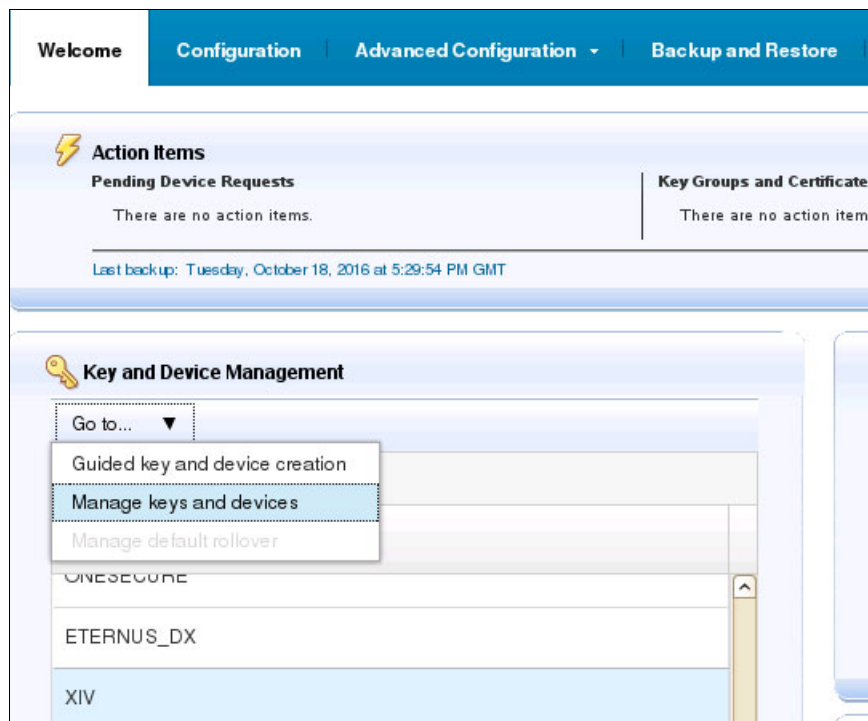


Figure 5-19 Manage keys and devices window

6. Select **Machine affinity** and select **Hold new device requests pending my approval**, as shown in Figure 5-20. This action ensures that when you generate the recovery key or add the key server in the IBM FlashSystem A9000 or IBM FlashSystem A9000R system that it shows as Pending in the Welcome window.

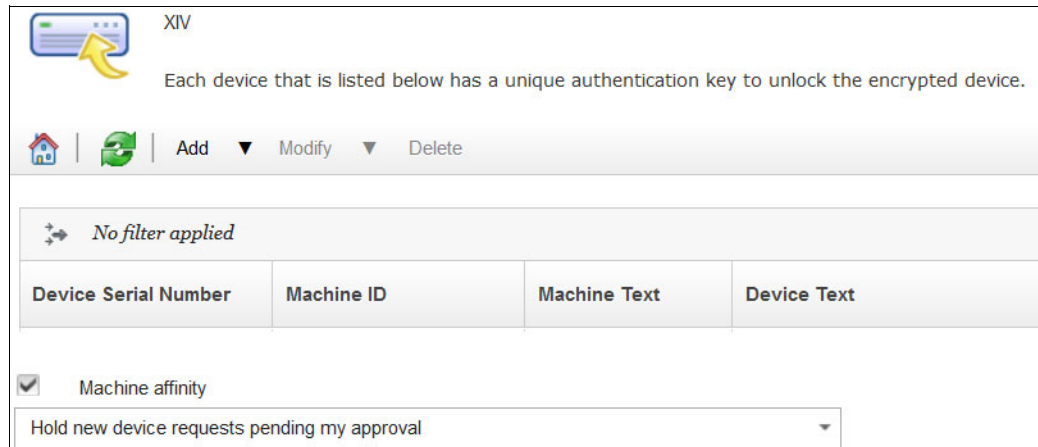


Figure 5-20 Machine affinity and Hold new device requests pending approval

7. Import the newly created SSL certificate as “trusted” in the IBM Security Key Lifecycle Manager web GUI. Click **Advanced Configuration** → **Client Certificates**, and click **Import** (under the SSL/KMIP Certificate for Clients section), as shown in Figure 5-21 and in Figure 5-22 on page 71.

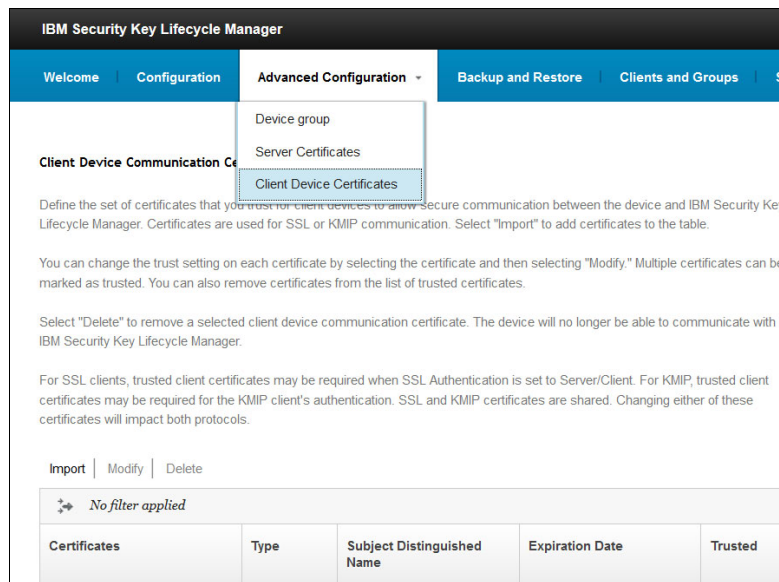
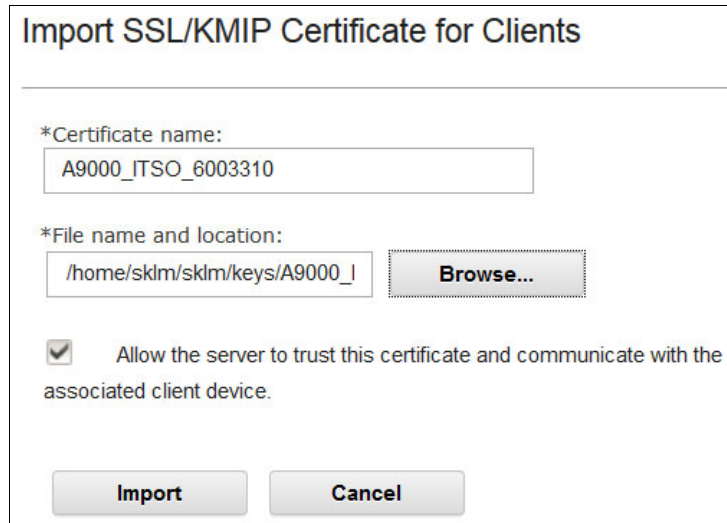


Figure 5-21 Client Device Communication Certificates display



**Import SSL/KMIP Certificate for Clients**

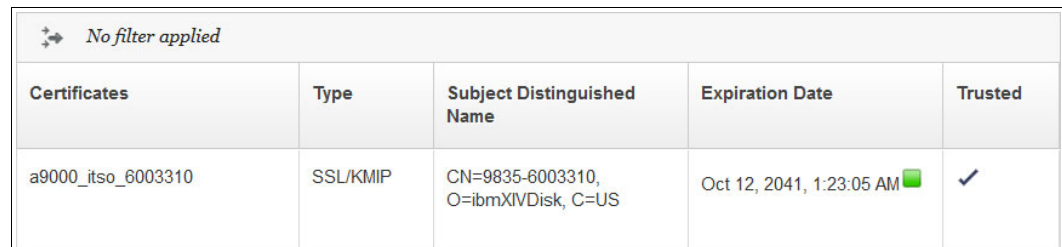
\*Certificate name:

\*File name and location:

☒ Allow the server to trust this certificate and communicate with the associated client device.

Figure 5-22 Import SSL/KMIP Certificate for Clients

The certificate shows as trusted with a type of SSL/KMIP and its name and expiration date are shown, as shown in Figure 5-23.





No filter applied				
Certificates	Type	Subject Distinguished Name	Expiration Date	Trusted
a9000_itso_6003310	SSL/KMIP	CN=9835-6003310, O=ibmXIVDisk, C=US	Oct 12, 2041, 1:23:05 AM 	

Figure 5-23 Show the trusted device-specific certificate

- To define the key server in the IBM FlashSystem A9000 or IBM FlashSystem A9000R system, you must create and export the key server certificate on the IBM Security Key Lifecycle Manager and add it to the IBM FlashSystem A9000 or IBM FlashSystem A9000R system afterward.

Create an IBM Security Key Lifecycle Manager self-signed certificate in the IBM Security Key Lifecycle Manager GUI (add the SSL/KMIP certificate for key serving).

If this is a new key server installation, you must have an Action Item in the IBM Security Key Lifecycle Manager Welcome window to create the server certificate, as shown in Figure 5-24.

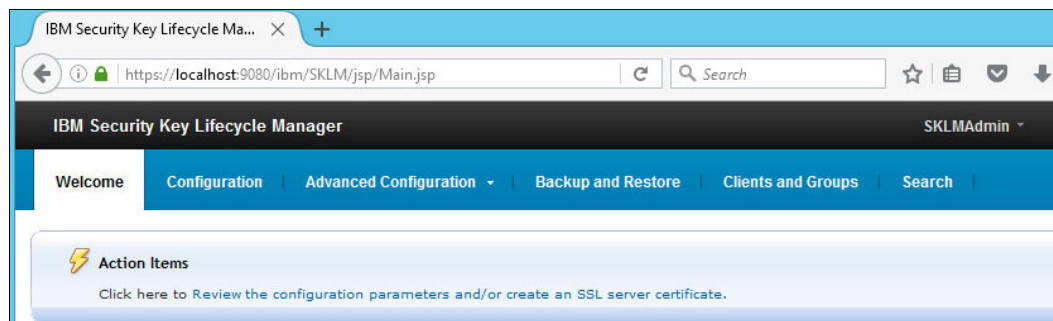


Figure 5-24 IBM Security Key Lifecycle Manager Welcome with initial configuration Action Item

If the Action Item in the Welcome window is not available anymore, the server certificate can be created by completing the following steps:

- a. In the window that is shown in Figure 5-24 on page 71, under Advanced Configuration, click **Create self-signed certification**.
- b. When the window that is shown in Figure 5-25 opens, create the certificate that is used to encrypt data for secure communication over SSL.

IBM Security Key Lifecycle Manager

Welcome Configuration **Advanced Configuration** Backup and Restore Clients and Groups Search

Administer Server Certificates

For secure communication, the server must have a certificate.

To create a certificate, click the **Add** button.

Add Modify Delete

No filter applied

Certificates

Total: 0 Selected: 0

### Add SSL/KMIP Certificate

☒ **Create self-signed certificate**  
Use a self-signed certificate in a known environment. A client cannot verify a self-signed certificate before accepting a connection.

☐ **Request certificate from a third-party provider**  
Use a certificate generated by a third-party provider as a more secure means of communication. A client can verify a certificate from a third-party provider before accepting a connection.

[Self-signed Certificate](#)

\*Certificate label in keystore:  
SKLM\_ITSO\_6003310

\*Certificate description (common name):  
SKLM\_ITSO\_6003310

\*Validity period of new certificate (in days; for example, 3 years is 365 x 3 = 1095 days):  
3650 The interval in days ranges from 1 to 9000

\*Algorithm:  
RSA

Organizational unit name:

Add Certificate Cancel

Figure 5-25 Add SSL/KMIP Server Certificate

**Tip:** Do not confuse this certificate with the *device* certificate that is associated with the IBM FlashSystem A9000 or IBM FlashSystem A9000R system.

- c. Select **Create self-signed certificate**. Third-party signed certificates are also supported.

**Caution:** Although using an existing certificate from the key store is possible, using the same certificate to encrypt disk data and protect the communication protocol is *not* recommended.

- d. Choose a descriptive label and a certificate expiration validity in days in accordance with your security guidelines. You also may enter certificate parameters.
- e. Click **Add Certificate**.

As indicated in the Warning notice that is shown in Figure 5-26 on page 73, the SSL/KMIP certificate is updated. For this change to take effect, you must restart the server. Stopping and starting the key server is described in 6.2, “Starting and stopping an IBM Security Key Lifecycle Manager server” on page 93. Also, create a backup to ensure that you can restore this data.

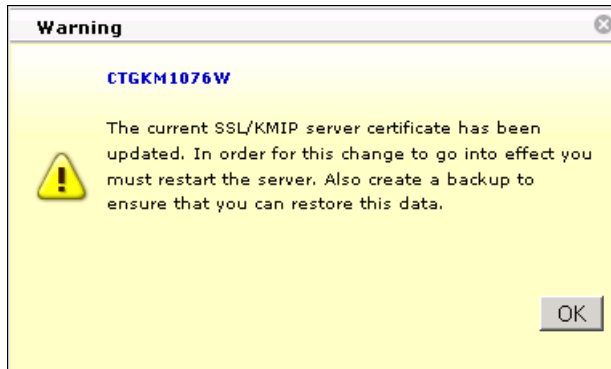


Figure 5-26 Reminder to restart the server

9. In the left pane of the IBM Security Key Lifecycle Manager GUI, click **Welcome** to return to the Welcome window, as shown in Figure 5-27.

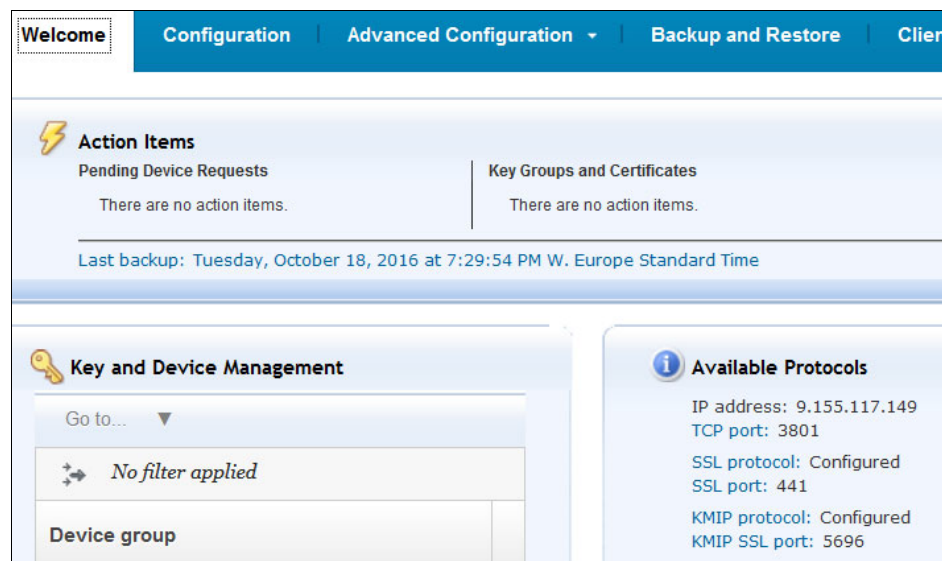


Figure 5-27 Welcome window

You have now created the IBM Security Key Lifecycle Manager master key store and the SSL certificate. As a result, the Available protocols section now has both *SSL protocols* and *KMIP protocols* configured.

After it is created, you must export the certificate. In previous versions earlier than IBM Security Key Lifecycle Manager V2.6, you can do this only by running **wsadmin**.

10. Exporting IBM Security Key Lifecycle Manager server certificates can be done in the IBM Security Key Lifecycle Manager GUI starting with Version 2.6 by completing the following steps:

- a. Log in to the IBM Security Key Lifecycle Manager server after if restarts and click **Advanced Configuration** → **Server Certificates**, as shown in Figure 5-28.

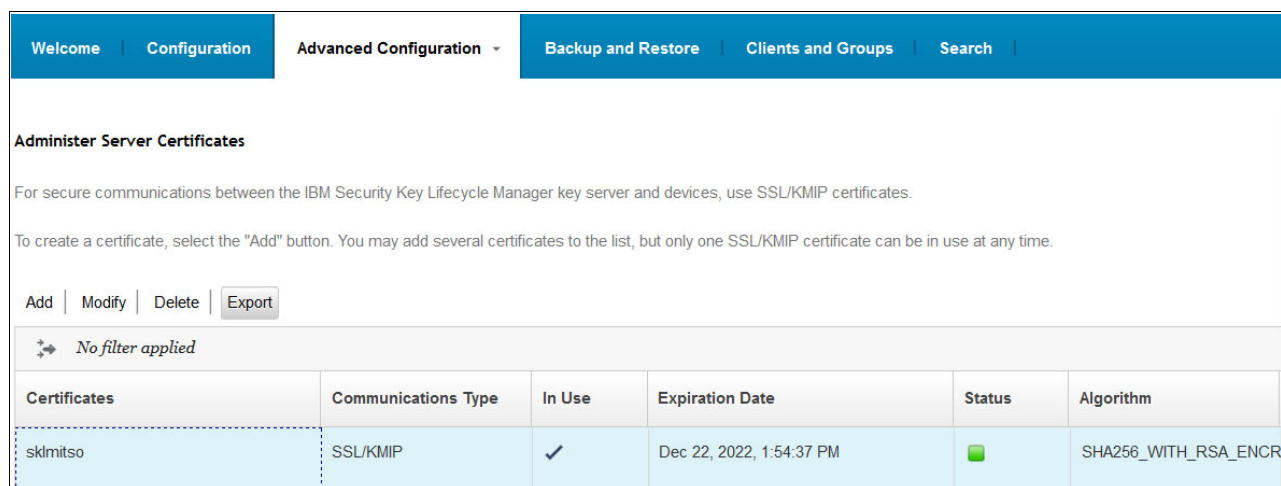


Figure 5-28 Export the server certificate

- b. Select your desired server certificate file name, including the .pem file ending, and browse to the file location where you want it to be saved. Choose certificate type base64 because the DER format is not supported in an XIV system, as shown in Figure 5-29.

The screenshot shows the 'Export Certificate' dialog box. It has a title bar 'Export Certificate'. Inside, there are three main sections: 'File name:' with a text input field containing 'sklmitso.pem'; 'File location:' with a text input field containing '/home/ftp' and a 'Browse...' button to its right; and 'Certificate type:' with two radio buttons, 'base64' (which is selected) and 'DER'. At the bottom of the dialog are two buttons: 'Export Certificate' and 'Cancel'.

Figure 5-29 Export certificate type base64

11. For IBM Security Key Lifecycle Manager Version 2.5 and earlier, run **wsadmin** to export the server certificate:

- For a Microsoft Windows operating system, open a DOS prompt with Administrator privileges. Run the following **wsadmin** command:  

```
cd <WAS_HOME> wsadmin -username skladmin -password <skladmin password> -lang jython
```

<WAS\_HOME> is, for example, C:\Program Files (x86)\IBM\WebSphere\AppServer.
- For a Linux operating system, open a UNIX terminal session, and run the following command:  

```
cd <WAS_HOME>/bin>rm -f ./tmp/cert.der  
./wsadmin.sh -username SKLAdmin -password <skladmin password> -lang jython
```

<WAS\_HOME> is, for example, /opt/IBM/WebSphere/AppServer/bin/.

12. To view all existing certificates, run the **print AdminTask.tklmCertList()** command that is shown in Figure 5-30.

```
wsadmin>print AdminTask.tklmCertList('[-alias a9000_itso_6003310]')
CTGKM0001I Command succeeded.

uuid = CERTIFICATE-23882e94-73b3-4282-b253-185569a76a4d
alias = a9000_itso_6003310
key store name = defaultKeyStore
key state = ACTIVE
issuer name = O=ibmXIVDisk,C=US
subject name = CN=9835-6003310,O=ibmXIVDisk,C=US
creation date = 10/18/16 6:22:45 PM GMT+00:00
expiration date = 10/12/41 1:23:05 AM GMT+00:00
serial number = 10001000
```

Figure 5-30 List all certificates command in wsadmin

13. Print the certificate that is created in step 8 on page 71 by running the **print** command:

```
print AdminTask.tklmCertList('[-alias <label provided in Step 2>]')
```

Figure 5-31 shows an example of the result.

```
wsadmin>print AdminTask.tklmCertList('[-alias a9000_itso_6003310]')
CTGKM0001I Command succeeded.

uuid = CERTIFICATE-23882e94-73b3-4282-b253-185569a76a4d
alias = a9000_itso_6003310
key store name = defaultKeyStore
key state = ACTIVE
issuer name = O=ibmXIVDisk,C=US
subject name = CN=9835-6003310,O=ibmXIVDisk,C=US
creation date = 10/18/16 6:22:45 PM GMT+00:00
expiration date = 10/12/41 1:23:05 AM GMT+00:00
serial number = 10001000
```

Figure 5-31 List a specific certificate in wsadmin

14. Take the Universally Unique Identifier (UUID) information from the output of step 12 on page 75 and use it to export the certification file. You might want to change the **-fileName** option to something other than `/tmp/cert.der` if you want to save it in a different folder.

The specified folder and file name are relative. Therefore, if you specify `/tmp/cert.der`, it is saved in a subdirectory of your IBM Security Key Lifecycle Manager installation directory. On a Windows server, you can find it in this path:

```
C:\ibm\tivoli\tpk\lmV2\products\tklm\tmp\cert.der
```

15. Export the certification file by running the following command:

```
print AdminTask.tklmCertExport('[-uuid
CERTIFICATE-a44aba79-6bcc-47dd-94c0-23ddb5db102c -format base64 -fileName
/tmp/cert.pem ]')
```

This is a successful output response:

```
CTGKM0001I Command succeeded /tmp/cert.pem
```

This `.pem` file is the certificate that is passed by a parameter in the XCLI **encrypt\_keyserver\_define** command (as described in “Step 5: Defining the key server on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system”).

## Step 5: Defining the key server on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system

You can define a key server on an IBM FlashSystem A9000 or IBM FlashSystem A9000R system by adding the IBM Security Key Lifecycle Manager certificate that you just exported to the key server. To do so, run the XCLI **encrypt\_keyserver\_define** command.

Each operating system has its own way of creating line breaks in a file. Those breaks are not always visible, and if the server certificate is specified to be added as a file name, it can lead to a bad `cert` status after the key server is added.

One way to eliminate this error is to edit the server certificate in a text editor and create the whole **encrypt\_keyserver\_define** command as one line, including the certificate, and add the `“*”` (asterisk) character after each line break that you remove, as shown in Figure 5-32.

```
A9000_ITS0>>encrypt_keyserver_define name=sklm1 ipv4=9.123.123.123 master=yes
certificate="-----BEGIN
CERTIFICATE-----*MIICyTCCAbGgAwIBAgIGKSiYd1FPMAOGCSqGSIb3DQEBCwUAMBQxEjAQBgNVBA
MTCXNrbG1pdHNv*MzAeFw0xNjEwMjEwOTIxMzlaFw0yMjEwMjEwOTIxMzlaMBQxEjAQBgNVBAMTCXNr
bG1pdHNvMzCC*ASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIGeknLONw39+wYw4ZTtUWYjtN
u00eUJ1we0*UgG6pdfZR1N5xj5ijEYENiWMHFxkK06Su871QLXTEvL3QsQ1xrAmOKMuit0QxNugguEs
LaCim1Nhawv1Da*Rtd1eD1AKeD1AfERReiRaZhaiVnUw8uvURzbLlofgg+/iKfnPeFLBeBqrnkI17Fq
rdVN0T0bbFxp0V6*DfM49pjG81M5kAR93R05UysS1z6N0w9srK6eGCfNJGWqzpsHYK8gBmfDHco/j
/hD4+1TjLbnxna*w8gdQYTFRDkWR4oteFKDB1LM7szIHLZ3VS2CjkyW/VZHXGcqzzALy3ZP5uvWFFKW
T5TH03I80tJe*xpkCAwEAAAMhMB8wHQYDVR00BBYEFDUo+63Y6PQCGt9qM8P60j+cZfC7MAOGCSqGSI
b3DQEBCwUA*A4IBAQBwx7WrZGN9E77LcFQmjKwty90y44cPHuf5B7d064x4q3pZ0qFAuY10BPAExw0g
hDgW6IeY*nkaWWGfIfTie/NA+U/sP2toVVtR/xuNQSp2WKnMnd8ddNosHo2q9Xprx4d2CJS1H7oUj+
Maf70L*nap0baHKyaa3veLH0fxyN/HWFQTf0Kf0qYwq1Cypx3nLw3XF+a3PKAAZA1hSnMfdNG59RJY+
xoa4*fzrmAlEvX6iUqZV3jy3Eg2mf3Pam/qP1Pbelmz14SdJBC+XMfJ2dquidQKedVYymSVy977NF4T
ws*erD6HgQHskfR3FEM+b7EfOUPFIbrys8rKtLRbWvovebq*-----END CERTIFICATE-----"
```

Figure 5-32 The `encrypt_keyserver_define` command that uses the server certificate

You can verify that the key server was added successfully by running the **encrypt\_keyserver\_list** command that is shown in Figure 5-33 on page 77.

A9000_ITS0>>encrypt_keyserver_list									
Module	Name	App/Key	Status	Last_time_checked	Master	Port	Address	Keyserver	Type
1	sk1m1	NOKEY		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM	
2	sk1m1	NOKEY		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM	
3	sk1m1	NOKEY		2016/10/26 13:48:45	yes	5696	9.123.123.123	TKLM	

Figure 5-33 The encrypt\_keyserver\_list command

The NOKEY status shows that the IBM FlashSystem A9000 or IBM FlashSystem A9000R system has not been accepted on the key server. You see a Pending Devices link in the Welcome window of the key server GUI, as shown in Figure 5-34.

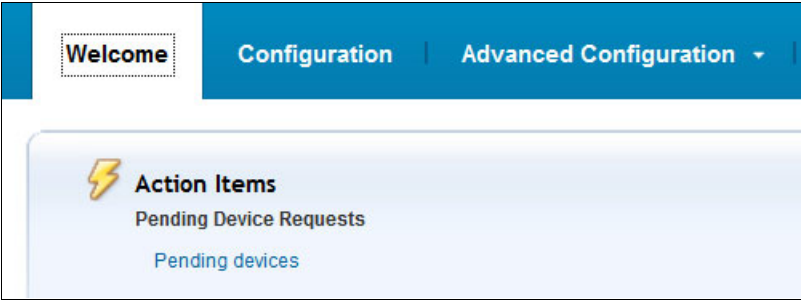


Figure 5-34 Pending Devices in the Welcome window

Complete the following steps:

1. Click the **Pending Devices** action item and the IBM FlashSystem A9000 or IBM FlashSystem A9000R system that you added appears, as shown in Figure 5-35.

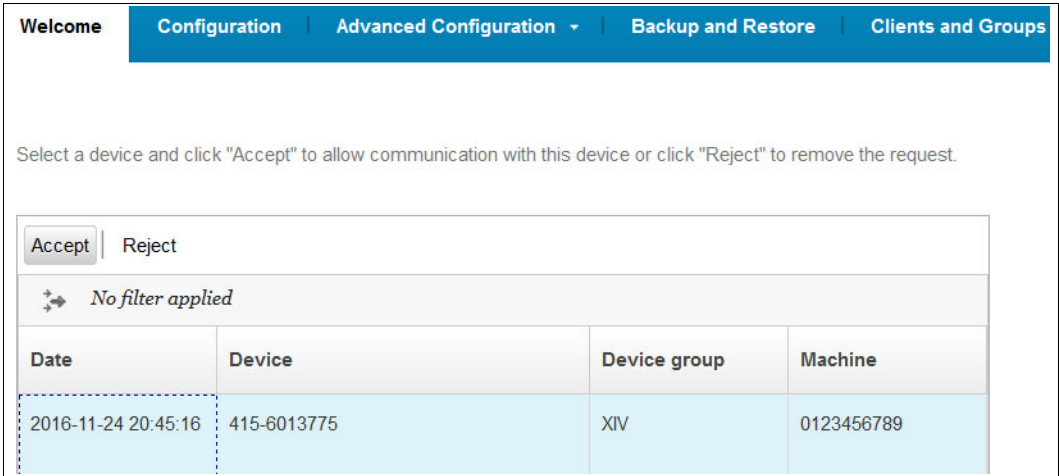


Figure 5-35 Accept the pending IBM FlashSystem A9000 or IBM FlashSystem A9000R system device

- Highlight the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and click **Accept** to add it to the key server. If this device served keys in the past from that key server, a warning message, as shown in Figure 5-36, appears and advises you to take a backup before accepting the pending device. This situation can happen during migrations or if encryption was enabled and disabled before. It is a preferred practice to create a backup now to be able to revert to the current state of configuration of the key server.

### Accept Device Request

**Warning: Keys have been generated for this device. Perform a backup before accepting this request.**

Click Accept to add the device to the system.

Click Modify and Accept to modify the device properties before adding the device to the system.

Device Serial Number: 415-6013775  
Device group: XIV  
Device Text:  
Machine ID: 0123456789  
Machine Text: A9000 pod itso lab tucson

Accept

Modify and Accept

Cancel

Figure 5-36 Accept Device Request

After the Device Request is accepted, it shows as a Client Device Communication Certificate in the Advanced Configuration section, as shown in Figure 5-37.

Import   Modify   Delete				
No filter applied				
Certificates	Type	Subject Distinguished Name	Expiration Date	Trusted
a9000_itso_6013775	SSL/KMIP	CN=2810-6013775, O=ibmXIVDisk, C=US	Mar 16, 2041, 4:26:42 PM	✓
a9000_itso_6003310	SSL/KMIP	CN=9835-6003310, O=ibmXIVDisk, C=US	Oct 12, 2041, 1:23:05 AM	✓

Figure 5-37 Client Device Communication Certificates

## 5.4 Recovery key use and maintenance

To protect against the possibility (following a disaster, for example) that all IBM Security Key Lifecycle Managers become unusable and unrecoverable, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system enables you to create a *recovery key*, as shown in Figure 5-38 on page 79. With a recovery key, Security Administrators can unlock an IBM FlashSystem A9000 or IBM FlashSystem A9000R system without the involvement of an IBM Security Key Lifecycle Manager server.

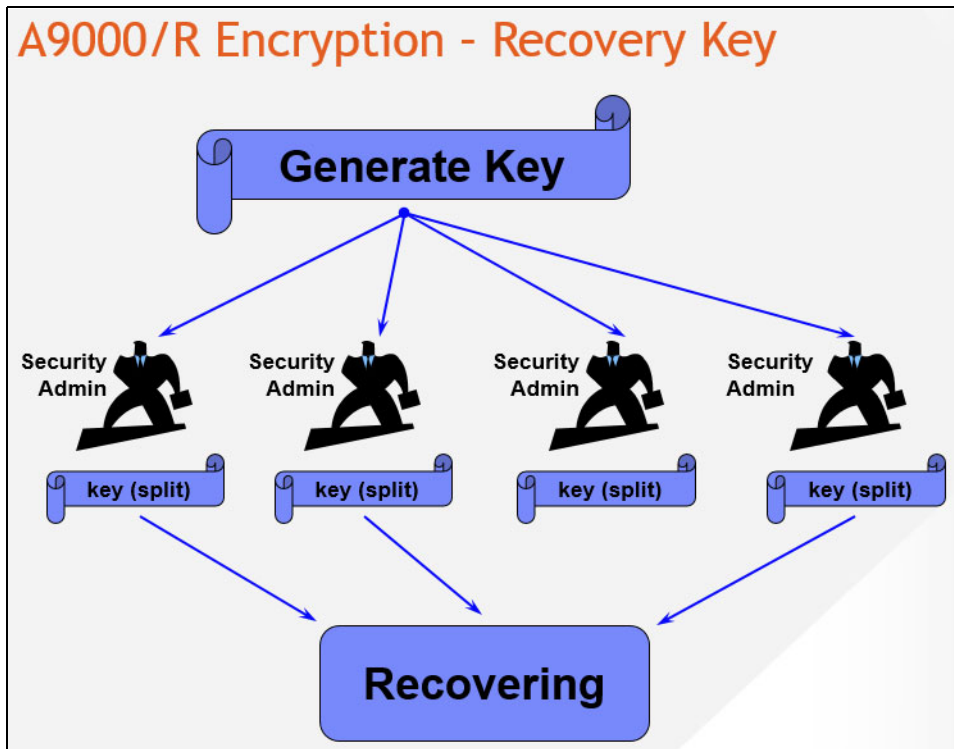


Figure 5-38 Recovery key

Encryption can be activated in the XCLI. The option to activate data-at-rest encryption without recovery keys is possible, but only through the XCLI by running the **encrypt\_enable** command with the **recovery\_keys=no** flag. The recovery keys are split according to the number of defined Security Administrators and created separately for each Security Administrator.

The recovery key is used to unwrap the XMK, which unlocks the drives.

**Important:** A recovery key can be created only if data-at-rest encryption is not yet enabled. You cannot create a recovery key when IBM FlashSystem A9000 or IBM FlashSystem A9000R encryption is already activated.

Managing the recovery key requires at least two Security Administrators. They maintain the recovery key and keep it safe.

**Client responsibility:** Although an IBM FlashSystem A9000 or IBM FlashSystem A9000R system supports two roles, Storage Administrator and Security Administrator, only the Security Administrator is allowed to use the recovery key. The client is responsible for assigning at least two *separate* individuals as Security Administrators to prevent data access by a single person.

### 5.4.1 Process for recovery keys

The recovery keys can be generated only if data-at-rest encryption is deactivated on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system. Make sure that at least two XIV Security Administrators are defined on the system. Recovery key creation requires communication with the key server.

The following steps are required to configure recovery keys:

1. Generate recovery keys for each Security Administrator.
2. Get keys for each Security Administrator (make a note of them for later).
3. Verify keys for each Security Administrator to make the keys usable.

The IBM FlashSystem A9000 or IBM FlashSystem A9000R system generates a random recovery key and a related wrapping key. The recovery key can also be rekeyed, which generates a new recovery key. That new key must be acquired and verified again by each defined Security Administrator.

## 5.4.2 Recovery key generation with XCLI

If you prefer, you can generate the recovery key through the XCLI by completing the following steps:

1. Start with the **encrypt\_recovery\_key\_generate** command that is shown in Table 5-1.

Table 5-1 The *encrypt\_recovery\_key\_generate* command

Category	Command	Description
System	<b>encrypt_recovery_key_generate</b>	Specifies which Security Administrators receive recovery key shares and the minimum number of recovery key shares that must be entered

Here is an example of this command:

```
A9000_ITS0>>encrypt_recovery_key_generate min_req=2
users=itsosecadmin1,itsosecadmin2
Command executed successfully.
```

2. Each defined Security Administrator must collect and verify the keys that are generated individually by using their credentials to log in to the XCLI, as shown in Table 5-2.

Table 5-2 The *encrypt\_recovery\_key\_get* command

Category	Command	Description
System	<b>encrypt_recovery_key_get</b>	Retrieves the recovery key share that is generated for the current user

Here is an example of this command:

```
A9000_ITS0>>encrypt_recovery_key_get
Command executed successfully.
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
```

3. All defined Security Administrators must collect their keys.

### 5.4.3 Recovery key verification

Complete the following steps:

1. All Security Administrators users that are defined by the **encrypt\_recovery\_key\_generate** command must verify their keys, as shown in Table 5-3.

Table 5-3 The *encrypt\_recovery\_key\_verify* command

Category	Command	Description
System	<b>encrypt_recovery_key_verify</b>	Confirm s that the current user correctly copied the recovery key share that is presented by the <b>encrypt_recovery_key_get</b> command

Here is an example of this command:

```
A9000_ITS0>>encrypt_recovery_key_verify
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
Command executed successfully.
recovery_status=Key accepted, 1 of 2 fragments have been verified
remaining_fragments=1
```

2. The state of verification can be checked by running the **encrypt\_recovery\_key\_status** command that is shown in Table 5-4.

Table 5-4 The *encrypt\_recovery\_key\_status* command

Category	Command	Description
System	<b>encrypt_recovery_key_status</b>	Shows the status of the recovery keys

Here is an example of this command:

```
A9000_ITS0>>encrypt_recovery_key_status
Date Created      User              Status
2016-11-24 11:55:15  itsosecadmin1  Verified
2016-11-24 11:55:15  itsosecadmin2  Unverified
```

You can run **encrypt\_recovery\_key\_list** to show the number of shares that have recovery keys and how many of them are required for recovery.

3. The next Security Administrator must log in to the XCLI with the correct credentials and repeat the Activate Recovery Key procedure.
4. Save the key in a text file and keep it in a secure place that is physically separate from both the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and the IBM Security Key Lifecycle Manager servers.

After all defined Security Administrators have collected and verified their keys, the IBM FlashSystem A9000 or IBM FlashSystem A9000R data-at-rest encryption can be activated. For more information, see 5.5.1, “Activating data-at-rest encryption” on page 85.

## 5.4.4 Recovery key rekey

*Rekeying* is the process of changing cryptographic values in the chain between the key server, recovery key, and DAKs so that the previous value no longer enables access to the system.

The rekey and Verify Recovery Key functions can be performed any time while the recovery key is configured and an IBM Security Key Lifecycle Manager server is available. An IBM Security Key Lifecycle Manager server is required to enable the IBM FlashSystem A9000 or IBM FlashSystem A9000R system to verify that it is in the correct environment.

Only when the IBM Security Key Lifecycle Manager can decrypt the data key can the IBM FlashSystem A9000 or IBM FlashSystem A9000R system be sure that it is in the same environment. Only then, it generates a new recovery key. For example, on an IBM FlashSystem A9000 or IBM FlashSystem A9000R system that was stolen and put in a separate environment, rekeying the recovery key is not possible.

During a rekeying operation, the following actions are performed:

1. The IBM FlashSystem A9000 or IBM FlashSystem A9000R system sends the ESK to the IBM Security Key Lifecycle Manager and requests a rekey validation.
2. The IBM Security Key Lifecycle Manager verifies the identity of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system by using its certificates.
3. The IBM Security Key Lifecycle Manager signals the IBM FlashSystem A9000 or IBM FlashSystem A9000R system that it can proceed to generate a recovery key.
4. The IBM FlashSystem A9000 or IBM FlashSystem A9000R system generates a recovery key.

Changing the recovery key does not erase the data.

An Unconfigure function of the recovery key is not available after data-at-rest encryption is activated, but you can use the Regenerate Recovery Key function to change your keys.

The recovery key can also be rekeyed to replace the current recovery key with a new one. All defined Security Administrators must collect and verify the new recovery key.

The recovery key can be rekeyed in the XCLI by running the **encrypt\_recovery\_key\_rekey** command that shown in Table 5-5.

Table 5-5 *encrypt\_recovery\_key\_rekey*

Category	Command	Description
System	<b>encrypt_recovery_key_rekey</b>	Restarts the recovery key generation process that is described in <b>encrypt_recovery_key_generate</b>

Figure 5-39 shows the command.

```
A9000 ITS0>>encrypt_recovery_key_rekey
Command executed successfully.
```

Figure 5-39 The `encrypt_recovery_key_rekey` command

After the recovery key is replaced with a new one, the same recovery key verification procedure that is shown in 5.4.3, “Recovery key verification” on page 81 must be performed again by the members that were originally defined by the `encrypt_recovery_key_generate` command.

### 5.4.5 Using a recovery key to unlock an IBM FlashSystem A9000 or IBM FlashSystem A9000R system

On an IBM FlashSystem A9000 or IBM FlashSystem A9000R system with a recovery key that is configured, an option exists to let a Security Administrator enter the recovery key.

After a power-off followed by a power-on action, or after a simple reboot, if the IBM FlashSystem A9000 or IBM FlashSystem A9000R system cannot get the required data key from the SMK server, it attempts to contact all other configured IBM Security Key Lifecycle Manager servers to obtain the required key. In some key server versions, a secondary key server must be promoted to master first. If that is not successful, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system enters `maintenance_mode` status, which can be verified by running `state_list`, as shown in Figure 5-40.

```
A9000 ITS0>>state_list
Category                                Value
-----
data_protection_status                  N/A
data_reduction_state                    Online
encryption                             Enabled
safe_mode                              no
shutdown_reason                         No Shutdown
system_state                           maintenance
target_state                           on
```

Figure 5-40 `state_list` command

In this case, the Security Administrators must provide their recovery keys. The IBM FlashSystem A9000 or IBM FlashSystem A9000R system uses the recovery key to unwrap the XIV SMK that unlocks the IBM FlashSystem A9000 or IBM FlashSystem A9000R system. After access to data is restored, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system is available to serve host I/O again.

Each Security Administrator enters their individual parts of the recovery key until the number of defined minimum required Security Administrators is reached and it is again possible to unlock the IBM FlashSystem A9000 or IBM FlashSystem A9000R system. The XCLI command to do so is **encrypt\_recovery\_key\_enter**, as shown in Table 5-6.

Table 5-6 *encrypt\_recovery\_key\_enter*

Category	Command	Description
System	<b>encrypt_recovery_key_enter</b>	Unlocks encrypted disks when the system reboots and cannot access any of the defined key servers if the recovery keys were defined

This example shows the command:

```
encrypt_recovery_key_enter
key=62807CB1902AM074EDLA4EV8F0C574E40A1564F55570CDEEBED37BC3876789
```

As soon as the last one of the minimum number of defined Security Administrators has logged in with credentials and entered a recovery key, the IBM FlashSystem A9000 or IBM FlashSystem A9000R system unlocks and activates the data-at-rest encryption again, as shown in Figure 5-41.

```
secadmin1 user:
A9000_ITS0>>encrypt_recovery_key_enter
key=717708CB7DE9A0BCB40A619B17B82ADD8390C032965EED89B76401C6C20A3F71
Command executed successfully.
    recovery_status=1 of 2 recovery keys accepted
secadmin2 user:
A9000_ITS0>>encrypt_recovery_key_enter
key=E042B767755D599F726071DB095016BDBA20F638B0659D320E16D3E2BA3BD5AE
Command executed successfully.
    recovery_status=2 of 2 recovery keys accepted - enabling storage access
```

Figure 5-41 *The encrypt\_recovery\_key\_enter command*

After the minimum required number of keys are entered, a Storage Administrator or a Security Administrator user must change the state of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system from maintenance to on by running the **encrypt\_recovery\_key\_finish** command, as shown in Figure 5-42.

```
A9000_ITS0>>encrypt_recovery_finish
Command executed successfully.
```

Figure 5-42 *The encrypt\_recovery\_finish command*

Now, you can verify by running the **state\_list** command again that the system\_state changed to on, as shown in Figure 5-43 on page 85.

```

A9000_ITS0>>state_list
Category                Value
system_state            on
target_state            on
safe_mode               no
shutdown_reason         No Shutdown
data_protection_status  Fully Protected
encryption              Enabled
data_reduction_state    Online

```

Figure 5-43 The `state_list` command

## 5.5 Activating or deactivating encryption

Now that the implementation and configuration of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system and its corresponding IBM Security Key Lifecycle Manager server are finished, you can enable (activate) the data-at-rest encryption in the IBM FlashSystem A9000 or IBM FlashSystem A9000R system.

### 5.5.1 Activating data-at-rest encryption

For data-at-rest encryption to complete successfully, all of these prerequisites must be fulfilled:

- ▶ The current encryption state must be DISABLED (displayed as Supported in the `state_list` output).
- ▶ One master key server must be configured successfully, and recovery keys must be generated and verified by at least two separate Security Administrators, unless a `recovery_keys=no` parameter was passed.

This command is entered by a Security Administrator to enable the data protection feature.

The data-at-rest encryption can be activated from the XCLI by using the command that is shown in Table 5-7.

Table 5-7 The `encrypt_enable` command

Category	Command	Description
System	<code>encrypt_enable</code>	Enables the data protection feature

Figure 5-44 shows the output of the command.

```

ITS0 A9000>>encrypt_enable
Warning:  ARE_YOU_SURE_YOU_WANT_TO_ENABLE_ENCRYPTION y/n: y
Command executed successfully.

```

Figure 5-44 Output of the `encrypt_enable` command

It can take a couple of minutes to enroll the MicroLatency modules and the vaulting SSDs. After the procedure finishes, a successful encryption of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system can be seen by running the **state\_list** command, as shown in Figure 5-45.

A9000_ITS0>> <b>state_list</b>	
Category	Value
system_state	on
target_state	on
safe_mode	no
shutdown_reason	No Shutdown
data_protection_status	Fully Protected
<b>encryption</b>	<b>Enabled</b>
data_reduction_state	Online

Figure 5-45 The *state\_list* command to verify that encryption is enabled

## 5.5.2 Deactivating data-at-rest encryption

This **encrypt\_disable** command disables the data protection feature. A prerequisite for this action is that no volumes and no pools are defined on the system. In addition to disabling the data protection, a cryptographic erase is performed on all protected storage to ensure that all existing user data is no longer accessible. After the command completes successfully, all MicroLatency modules and vault SSDs are left in an unlocked state. Disabling encryption when the encryption state is other than ACTIVE causes an error (the **state\_list** command must show it as Enabled).

In XCLI, run **encrypt\_disable** and the system prompts you to verify that you want to deactivate encryption on the IBM FlashSystem A9000 or IBM FlashSystem A9000R system, as shown in Figure 5-46.

A9000_ITS0>> <b>encrypt_disable</b>	
Warning:	Are you sure you want to disable encryption on this system? y/n: y
Command executed successfully.	

Figure 5-46 The *encrypt\_disable* command

## 5.6 Verifying the encryption state

These actions enable users to verify the encryption state of the system:

- ▶ With IBM FlashSystem A9000 or IBM FlashSystem A9000R software Version 12.0.1 or higher, the Encryption column in the output of the **state\_list** command shows whether the system is encrypted, as shown in Figure 5-47 on page 87. The **state\_list** command can show any of these states: *Supported* (encryption is disabled), *Enabling/Activating*, *Partial*, *Enabled/Activated*, *Enabling on Boot*, or *Disabling*.

```
A9000_ITS0>>state_list
Category          Value
system_state      on
target_state      on
safe_mode         no
shutdown_reason   No Shutdown
data_protection_status Fully Protected
encryption      Enabled
data_reduction_state Online
```

Figure 5-47 The `state_list` command

- The key server state can be checked by running the `encrypt_keyserver_list` command, as shown in Figure 5-48. The status is checked once every hour or if something has changed or was updated.

```
A9000_ITS0>>encrypt_keyserver_list
Module Name App/Key Status Last_time_checked Master Port Address Keyserver Type
1 sklm1 ACTIVE 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
2 sklm1 ACTIVE 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
3 sklm1 ACTIVE 2016/10/26 13:48:45 yes 5696 9.123.123.123 TKLM
```

Figure 5-48 The `encrypt_keyserver_list` command

- To force an update of the `encrypt_keyserver_list` state, you can run the `encrypt_keyserver_check_status` command, as shown in Figure 5-49.

```
A9000_ITS0>>encrypt_keyserver_check_status
Command executed successfully.
```

Figure 5-49 The `encrypt_keyserver_check_status` command

- The **flash\_enclosure\_list** command gives you various information about its state, as shown in Figure 5-50. One is the encrypted value that shows *yes* if the whole the IBM FlashSystem A9000 or IBM FlashSystem A9000R system is encrypted and no if it is not. The **key\_needed** value shows you whether the flash enclosure and its MicroLatency modules are locked. If they are locked, the value is *yes*.

```
A9000_ITS0>>flash_enclosure_list -x
<XCLIRETURN STATUS="SUCCESS" COMMAND_LINE="flash_enclosure_list -x">
  <OUTPUT>
    <flash_enclosure id="4d90e000000">
      <component_id value="1:Flash_Enclosure:1"/>
      <status value="OK"/>
      <currently_functioning value="yes"/>
      <control_path_status value="OK"/>
      <fru_part_number value="00DH521"/>
      <fru_identity value="11S00DH412YS16BG66N03Z"/>
      <temperature_state value="cool"/>
      <fw_level value="1.4.4.1-199.102"/>
      <required_service value=""/>
      <service_reason value=""/>
      <enabled value="yes"/>
      <cluster_ip value="14.10.204.35"/>
      <array_status value="OK"/>
      <redundancy_state value="online"/>
      <has_spare value="yes"/>
      <fw_upgrade_progress value="0"/>
      <fw_upgrade_status value="invalid"/>
      <fw_upgrade_committed value="no"/>
      <target_fw_version value=""/>
      <fw_file_name value=""/>
      <utility_file_name value=""/>
      <cluster_id value="000002006d2ab064"/>
      <encrypted value="yes"/>
      <key_needed value="no"/>
      <base_guid value="5005076061D24B80"/>
      <charging value="no"/>
      <flash_status value="ok"/>
    </flash_enclosure>
  </OUTPUT>
</XCLIRETURN>
```

Figure 5-50 The *flash\_enclosure\_list -x* command

- The **vault\_device\_list** command shows you the state of the SSDs that are used as vaulting devices in the IBM FlashSystem A9000 or IBM FlashSystem A9000R system. The syntax of the command is **vault\_device\_list [ module=ModuleNumber | vault\_device=ComponentId ]**. If you do not specify the vault device, it shows all the existing SSDs. After the encryption of the IBM FlashSystem A9000 or IBM FlashSystem A9000R system is enabled, the **encryption\_state** shows the value **Enrolled**, and **sw\_encryption\_active** shows the value **yes**, as shown in Figure 5-51.

```
A9000_ITS0>>vault_device_list vault_device=1:Vault_Device:1:1 -x
<XCLIRETURN STATUS="SUCCESS" COMMAND_LINE="vault_device_list
vault_device=1:Vault_Device:1:1 -x">
  <OUTPUT>
    <vault_device id="51313b00004">
      <component_id value="1:Vault_Device:1:1"/>
      <status value="OK"/>
      <currently_functioning value="yes"/>
      <capacity_in_bytes value="250059350016"/>
      <capacity value="250GB"/>
      <target_status value=""/>
      <model value="HUSMR1625ASS20E"/>
      <vendor value="LENOVO-X"/>
      <serial value="OPVKLBMA"/>
      <part_number value="00NA685"/>
      <requires_service value=""/>
      <service_reason value=""/>
      <temperature value="24"/>
      <firmware value="P4C9"/>
      <original_firmware value=""/>
      <revision value=""/>
      <drive_pn value=""/>
      <encryption_state value="Enrolled"/>
      <security_state value="Unchecked"/>
      ...etc...
      <accessible_modules id="14" value="no"/>
      <desc>
        <disk_id value="1"/>
        <read_fail value="no"/>
        <smart_code value="NO ADDITIONAL SENSE INFORMATION"/>
        <smart_fail value="no"/>
        <power_on_hours value="0"/>
        <power_on_minutes value="0"/>
        <last_sample_time value="0"/>
        <last_sample_serial value="OPVKLBMA"/>
        <last_time_pom_was_mod value="0"/>
        <temperature_status>
          .....
          <power_is_on value="no"/>
          <bgd_scan value="0"/>
          <sw_encryption_active value="yes"/>
        </desc>
      </vault_device>
    </OUTPUT>
  </XCLIRETURN>
```

Figure 5-51 The **vault\_device\_list -x** command

More data-at-rest encryption-related XCLI commands are listed in IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/STJKN5\\_12.0.2/c\\_category\\_\\_Encryption\\_enablement\\_and\\_support\\_commands.html](http://www.ibm.com/support/knowledgecenter/en/STJKN5_12.0.2/c_category__Encryption_enablement_and_support_commands.html)



# Maintaining

This chapter explains maintenance tasks that are related to data-at-rest encryption for the IBM XIV and the IBM FlashSystem A9000 and IBM FlashSystem A9000R systems.

It covers these topics:

- ▶ 6.1, “Automated replication” on page 92
- ▶ 6.2, “Starting and stopping an IBM Security Key Lifecycle Manager server” on page 93
- ▶ 6.3, “Server rekey” on page 95
- ▶ 6.4, “Encryption deadlock” on page 97
- ▶ 6.5, “Disk and module replacement” on page 99

## 6.1 Automated replication

Automated replication provides a way of cloning a key server automatically. This function avoids the need for manual or scripted backup and restore operations.

**Supported:** IBM Security Key Lifecycle Manager Version 2.6 or later supports automated replication functions through the GUI.

The replication capabilities depend on your version of IBM Security Key Lifecycle Manager:

► IBM Security Key Lifecycle Manager Version 2.6

IBM Security Key Lifecycle Manager Version 2.6 provides an operating system-independent GUI for automated replication configuration. You can configure the replication program to replicate IBM Security Key Lifecycle Manager critical data across clone servers when new keys are added to the master server.

The automated replication process enables cloning of IBM Security Key Lifecycle Manager environments to multiple servers in a manner that is independent of operating systems and the 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. You can clone a master IBM Security Key Lifecycle Manager server with up to 20 copies. For more information, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/en/SSWPVP\\_2.6.0/com.ibm.sk1m.doc/admin/cpt/cpt\\_ic\\_admin\\_replication\\_clone\\_master.html](http://www.ibm.com/support/knowledgecenter/en/SSWPVP_2.6.0/com.ibm.sk1m.doc/admin/cpt/cpt_ic_admin_replication_clone_master.html)

► IBM Security Key Lifecycle Manager Version 2.5

IBM Security Key Lifecycle Manager version 2.5 automated clone replication uses a program to clone a master IBM Security Key Lifecycle Manager with up to five copies. You can configure the program to replicate the keys and also other configuration information, such as when new keys are rolled over. This program automates the replication of everything that is needed. 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.

You can replicate the following data:

- Tables in the IBM Security Key Lifecycle Manager database
- All keys materials in the IBM Security Key Lifecycle Manager database
- IBM Security Key Lifecycle Manager configuration files apart from the replication configuration file

**Note:** This data is taken as part of an IBM Security Key Lifecycle Manager backup. During a replication, the replication configuration file is not backed up and passed to the clone.

You can configure IBM Security Key Lifecycle Manager replication with the `ReplicationSKLMgrConfig.properties` configuration file. You must specify the replication configuration file on all systems that are involved in the replication process. Each instance of IBM Security Key Lifecycle Manager is defined as either the master (the system that is to be cloned) or a clone (the system that the data is being replicated on). The master and clone systems must be identical. The operating system, directory structures, and IBM DB2® admin user must be same on all systems, or there might be unpredictable results.

For more information, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/SSWPVP\\_2.5.0.3/com.ibm.sk1m.doc\\_2.5.0.3/overview/cpt/cpt\\_ic\\_oview\\_featur\\_tk1m\\_replication.html](http://www.ibm.com/support/knowledgecenter/SSWPVP_2.5.0.3/com.ibm.sk1m.doc_2.5.0.3/overview/cpt/cpt_ic_oview_featur_tk1m_replication.html)

There is “how to set up” documentation in the IBM ISM Library (IBM ID required):

<https://www.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10TK02>

## 6.2 Starting and stopping an IBM Security Key Lifecycle Manager server

You might have to use the **startServer** or **stopServer** command to start or stop the IBM Security Key Lifecycle Manager server. Restarting that server, for example, is required after the completion of a restore task.

Versions 2.5 and 2.6 or later of IBM Security Key Lifecycle Manager server automatically restart after a backup file is restored when the `autoRestartAfterRestore` property value is true (default value) in the `SKLMConfig.properties` file.

### Starting and stopping the server by using scripts

Scripts to start and stop the IBM Security Key Lifecycle Manager server are in the `WAS_HOME/bin` directory for Linux and IBM AIX platforms. In the commands, the **server1** parameter is the default name of the configured IBM Security Key Lifecycle Manager server instance.

#### *Starting the IBM Security Key Lifecycle Manager server*

To start the server, run the command for your system:

- ▶ Microsoft Windows systems:

```
StartServer.bat server1
```

- ▶ Linux and IBM AIX systems:

```
./startServer.sh server1
```

#### *Stopping the IBM Security Key Lifecycle Manager server*

To stop the server, use the **stopServer** script for your system:

- ▶ Windows systems:

```
StopServer.bat server1 -username wasadmin -password Password
```

- ▶ Linux and AIX systems:

```
./stopServer.sh server1 -username wasadmin -password Password
```

When global security is enabled (do not disable global security when you use IBM Security Key Lifecycle Manager), enter the user ID and password of the IBM Security Key Lifecycle Manager console administrator as parameters for the **stopServer** script. The script prompts for these parameters if they are omitted, but you can specify them on the command line.

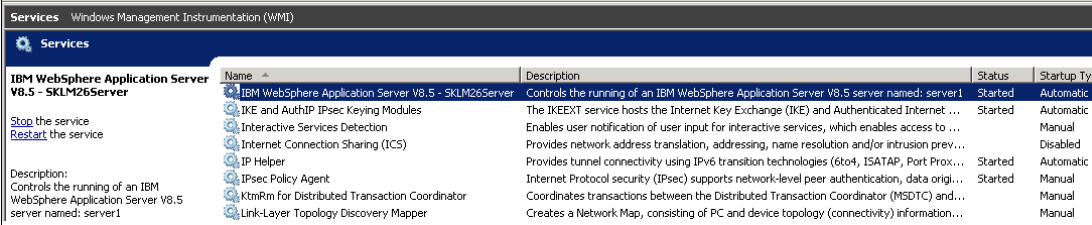
## Determining status

If you want to determine whether the IBM Security Key Lifecycle Manager server is running, try to log in to the IBM Security Key Lifecycle Manager web console. If the login is successful, the IBM Security Key Lifecycle Manager service is running. Otherwise, you can issue the **serverStatus** command (in the WAS\_HOME/bin) with the server instance, user name, and password parameters, as illustrated in Example 6-1.

### Example 6-1 Check server status

```
SKLM26SLES11:~ # /opt/IBM/WebSphere/AppServer/bin/serverStatus.sh server1
-username wasadmin -password Password
ADMU0116I: Tool information is being logged in file
/opt/IBM/WebSphere/AppServer/profiles/KLMProfile/logs/server1/serverStatus.log
ADMU0128I: Starting tool with the KLMProfile profile
ADMU0500I: Retrieving server status for server1
ADMU0508I: The Application Server "server1" is STARTED
```

For Windows systems, you can also check in the Services window to verify that the service is running, as shown in Figure 6-1.



Name	Description	Status	Startup Type
IBM WebSphere Application Server V8.5 - SKLM26Server	Controls the running of an IBM WebSphere Application Server V8.5 server named: server1	Started	Automatic
IKE and AuthIP IPsec Keying Modules	The IKEEXT service hosts the Internet Key Exchange (IKE) and Authenticated Internet ...	Started	Automatic
Interactive Services Detection	Enables user notification of user input for interactive services, which enables access to ...	Manual	Manual
Internet Connection Sharing (ICS)	Provides network address translation, addressing, name resolution and/or intrusion prev...	Disabled	Disabled
IP Helper	Provides tunnel connectivity using IPv6 transition technologies (6to4, ISATAP, Port Prox...	Started	Automatic
IPsec Policy Agent	Internet Protocol security (IPsec) supports network-level peer authentication, data origi...	Started	Manual
KtmRm for Distributed Transaction Coordinator	Coordinates transactions between the Distributed Transaction Coordinator (MSDTC) and...	Manual	Manual
Link-Layer Topology Discovery Mapper	Creates a Network Map, consisting of PC and device topology (connectivity) information...	Manual	Manual

Figure 6-1 Windows server service state check

In Linux, run the **ps** command, as shown in Example 6-2.

### Example 6-2 Linux server state check

```
SKLM26SLES11:~ # ps -ef |grep server1|grep -v grep
root      18905      1  0 Oct27 ?        00:12:17
/opt/IBM/WebSphere/AppServer/java_1.7.1_32/bin/java -Declipse.security
-Dwas.status.socket=34282 -Dosgi.install.area=/opt/IBM/WebSphere/AppServer
...etc...
/opt/IBM/WebSphere/AppServer/profiles/KLMProfile/config SKLMCell SKLMNode server1
```

Run the **tklmKeyExport** command with the **-alias**, **-fileName**, **-keyStoreName**, and **-type** parameters to export secret or private keys. The **-alias** parameter is the name from the IBM Security Key Lifecycle Manager server in the XIV or in the IBM FlashSystem A9000 and IBM FlashSystem A9000R system. The **-keyStoreName** is the master keystore name for the IBM Security Key Lifecycle Manager server. See Example 6-3.

### Example 6-3 Export the key store

```
wsadmin>print AdminTask.tklmKeyExport('[-alias 1310092 -fileName TKLM_XIV
-keyStoreName "defaultKeyStore_xiv" -type privatekey -password xxxxxxxx]')
CTGKM0001I Command succeeded.
```

The TKLM\_XIV file was created in this /opt/IBM/tivoli/tip/products/tklm.

Copy and archive the exported key to the new IBM Security Key Lifecycle Manager server. The exported keys are regular files on the file system. The way that they are transferred depends on the operating systems.

For more information, see IBM Knowledge Center:

[http://www.ibm.com/support/knowledgecenter/api/redirect/tivihelp/v2r1/index.jsp?to pic=%2Fcom.ibm.tklm.doc\\_2.0.1%2Fref%2Fref\\_ic\\_cli.html](http://www.ibm.com/support/knowledgecenter/api/redirect/tivihelp/v2r1/index.jsp?to pic=%2Fcom.ibm.tklm.doc_2.0.1%2Fref%2Fref_ic_cli.html)

## 6.3 Server rekey

The server rekey option is available on the XIV and on IBM FlashSystem A9000 and IBM FlashSystem A9000R systems. This option enables a user in the Security Administrator role to rekey against the master key server. As a preferred security practice, use this function to periodically change the keys. With an XIV system, you can use either the XIV GUI or the XIV XCLI. With IBM FlashSystem A9000 and IBM FlashSystem A9000R, you must use the XCLI.

### 6.3.1 XIV system rekey by using the XIV GUI

The following procedure describes how to rekey the server:

1. In the XIV GUI, click **All systems** → **List**, select your XIV system, right-click it, and choose the **Server Re-Key** menu entry, as shown in Figure 6-2.

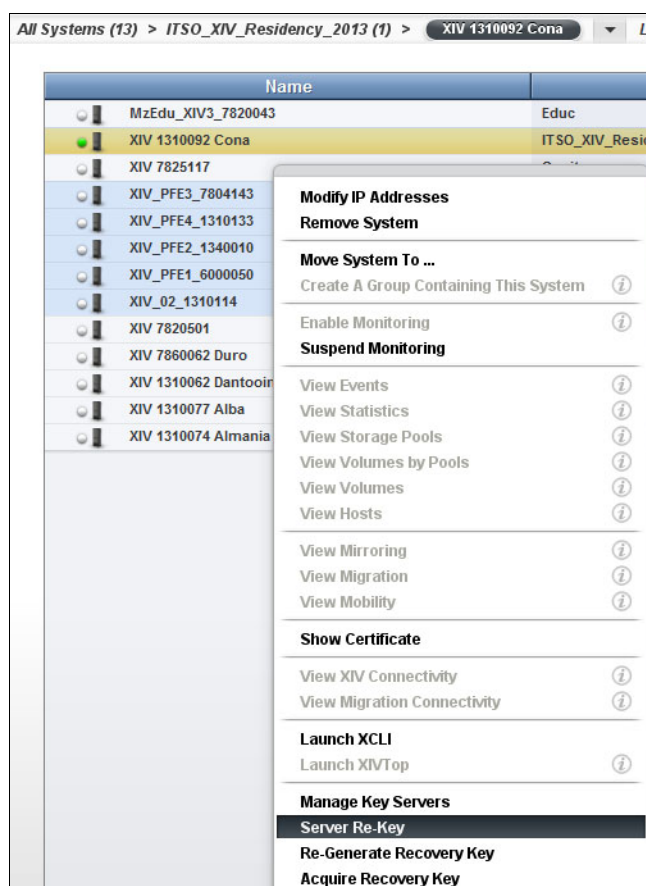


Figure 6-2 Server Re-Key

2. When the Server Re-key window that is shown in Figure 6-3 opens, select the XIV system from the drop-down menu and click **Start**.



Figure 6-3 Server Re-Key start

The key server must be available to process the rekey request. If it is not available, the error message that is shown in Figure 6-4 is displayed.



Figure 6-4 Server rekey error message

If the key server is available, it creates a key. The Completed Successfully confirmation message looks like the example in Figure 6-5.

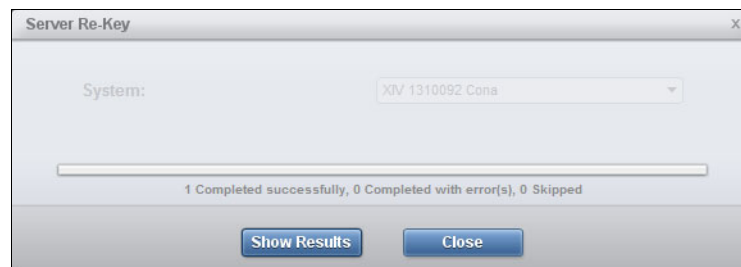


Figure 6-5 Server Rekey results

3. You can click **Show Results** to see the Completed Successfully message, as shown in Figure 6-6.

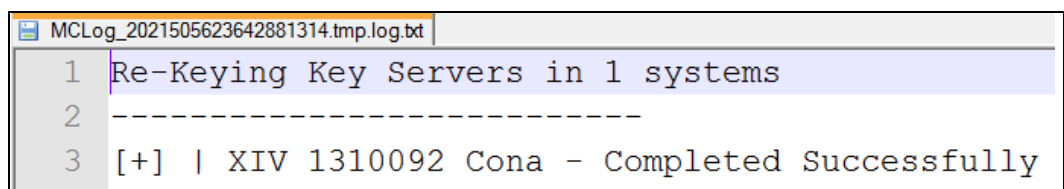


Figure 6-6 Rekey completed successfully

### 6.3.2 XIV and FlashSystem A9000 or A9000R server rekey by using XCLI

For XIV and IBM FlashSystem A9000 and IBM FlashSystem A9000R systems, the server key can be rekeyed by using the **encrypt\_keyserver\_rekey** XCLI command, as shown in Table 6-1.

Table 6-1 The *encrypt\_keyserver\_rekey* command

Category	Name	Description
System	<b>encrypt_keyserver_rekey</b>	Initiates a rekey against the master key server

## 6.4 Encryption deadlock

The key server platform provides the operating environment in which the key server application runs, accesses its keystore on persistent storage, and interfaces with client storage devices, such as the XIV or IBM FlashSystem A9000 and IBM FlashSystem A9000R systems, that require key server services.

The keystore data is accessed by the key server application through a password that is specified by the customer. As such, the keystore data is encrypted at rest, independently from 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 has a dependency 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 plus any other data that is required by that operating system and its associated software stack to run the key server application. This action is necessary to allow the key server to access its key store and to allow the key server to communicate with its storage device clients. Similarly, any backups of the key store must not be stored on a storage device that has a dependency on a key server to access data.

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

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

The encryption deadlock can be temporary or permanent:

<b>Temporary encryption deadlock</b>	The temporary encryption deadlock indicates a situation where the XIV or FlashSystem A9000 or A9000R system cannot access its disk devices because IBM Security Key Lifecycle Manager servers are not online, the network is down, or there are other temporary hardware-related errors. This temporary failure can be fixed at the client site.
<b>Permanent encryption deadlock</b>	This permanent encryption deadlock is the worst case scenario. Here, all IBM Security Key Lifecycle Manager servers that manage some set of data cannot be made operational either because they have a dependency on inaccessible encrypted storage or because all encrypted online and offline data that is managed by the set of IBM Security Key Lifecycle Managers is, in effect, cryptographically erased. <i>For all practical purposes, that data is permanently lost.</i>

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

- ▶ As the availability of encryption-capable devices becomes more pervasive, more data will be 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, several layers of virtualization in the I/O stack hierarchy can cause difficulties for the client in maintaining awareness of where all of the files that are 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. The volume manager communicates with a storage subsystem that provisions logical volumes with capacity that is obtained from other subordinate storage arrays. The data that is required by the key server might end up provisioned over various storage devices, each of which can be independently encryption-capable or encryption-enabled.
- ▶ Consolidation of servers and storage tends to drive data migration and tends to move increasingly more data under a generalized shared storage environment. This storage environment becomes encryption-capable as time goes by.
- ▶ All IBM server platforms support fabric-attached boot devices and storage. Some servers do not support internal boot devices. Therefore, boot devices are commonly present within the generalized storage environment. These storage devices are accessible to generalized storage management tools that support data management and relocation.

To mitigate the risk of an encryption deadlock, a stand-alone IBM Security Key Lifecycle Manager server is mandatory, and the client must be directly involved in managing the encryption environment. For more information about this topic, see Chapter 3, “Planning” on page 15, Chapter 4, “Implementing IBM XIV encryption” on page 25, and Chapter 5, “Implementing IBM FlashSystem A9000 and IBM FlashSystem A9000R encryption” on page 59.

## 6.5 Disk and module replacement

The following scenarios are considered:

- ▶ XIV disk and module replacement

If a disk or multiple disks must be replaced, the new disks must be unlocked upon power-on and contacting the key server to get the Disk Access Key (DAK). If a disk drive that does not support encryption is added to an encryption-enabled XIV system, it fails the component test and cannot be included in the running XIV configuration. This action ensures that no unencrypted data is on any disk drives inside the XIV system.

- ▶ FlashSystem A9000 and A9000R MicroLatency module and SSD replacement

If MicroLatency modules or vaulting SSDs must be replaced, the new parts must be encryption-capable, just like the original parts being replaced.



# 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 additional information about the topic in this document. Some publications that are referenced in this list might be available in softcopy only.

- ▶ *IBM FlashSystem A9000 and IBM FlashSystem A9000R Architecture, Implementation, and Usage*, SG24-8345
- ▶ *IBM XIV Storage System Architecture and Implementation*, SG24-7659

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](http://ibm.com/redbooks)

## Other publications and online resources

These publications are also relevant as further information sources:

- ▶ IBM Fix Central:  
<http://www.ibm.com/support/fixcentral/>
- ▶ IBM FlashSystem A9000 on IBM Knowledge Center:  
<http://www.ibm.com/support/knowledgecenter/STJKN5>

The following publications are at this website:

- *IBM FlashSystem A9000 Command-Line Interface (CLI) Reference Guide*, SC27-8559
- *IBM FlashSystem A9000 Models 9836-415 and 9838-415 Deployment Guide*, GC27-8564
- *IBM FlashSystem A9000 Product Overview*, GC27-8583-00
- *Hyper-Scale Manager 5.0 REST API Specification*, SC27-6440-01
- *Hyper-Scale Manager 5.0 User Guide*, SC27-8560

- ▶ IBM FlashSystem A9000R on IBM Knowledge Center:  
<http://www.ibm.com/support/knowledgecenter/STJKMM>

The following publications are at this website:

- *IBM FlashSystem A9000R Command-Line Interface (CLI) Reference Guide*, SC27-8711
- *IBM FlashSystem A9000R Model 415 Deployment Guide*, GC27-8565
- *IBM FlashSystem A9000R Product Overview*, GC27-8558-00

- ▶ IBM FlashSystem A9000 product page:  
<http://www.ibm.com/systems/storage/flash/a9000>
- ▶ IBM FlashSystem A9000R product page:  
<http://www.ibm.com/systems/storage/flash/a9000R>
- ▶ IBM Offering Information page (announcement letters and sales manuals):  
[http://www.ibm.com/common/ssi/index.wss?request\\_locale=en](http://www.ibm.com/common/ssi/index.wss?request_locale=en)  
On this page, enter A9000, select the information type, and click **Search**. On the next page, narrow your search results by geography and language.
- ▶ IBM System Storage Interoperation Center (SSIC) website:  
<http://www.ibm.com/systems/support/storage/ssic/interoperability.wss>
- ▶ *IBM XIV Storage System Planning Guide*, GC27-3913
- ▶ *IBM XIV Storage System: Product Overview*, GC27-3912
- ▶ *IBM XIV Storage System User Manual*, GC27-3914
- ▶ *IBM XIV Storage System XCLI Utility User Manual*, GC27-3915

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)





REDP-5402-00

ISBN 0738455830

Printed in U.S.A.

Get connected

