IBM XIV Security with Data-at-Rest Encryption

- Data on disk encrypted with no performance impact
- Hot encryption support with no downtime
- External key manager

Bert Dufrasne
Dietmar Dausner
Roger Eriksson
Roman Fridli
Itzhack Goldberg
Markus Oscheka
Stephen Solewin

Redpaper

ibm.com/redbooks

International Technical Support Organization

IBM XIV Security with Data-at-Rest Encryption

December 2013

REDP-5047-00

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

First Edition (December 2013)

This edition applies to the IBM XIV Storage System Gen3 with the IBM XIV Storage Software Version 11.4.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

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, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	IBM®	System p®
BladeCenter®	Power Systems™	System Storage®
DB2®	Redbooks®	System x®
DS6000™	Redpaper™	Tivoli®
DS8000®	Redbooks (logo) ®	XIV®

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.

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

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

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. As with all of its features, IBM® XIV® Storage System provides data-at-rest encryption at no charge. This gives clients the opportunity to take advantage of encryption and still enjoy the lower total cost of ownership that the XIV offers.

This IBM Redpaper™ publication explains the architecture and design of the XIV encryption solution and how it must be configured and implemented. It can help clients and Storage Administrators who want to enable data encryption on XIV storage systems.

Note: The IBM Tivoli® Key Lifecycle Manager used in preparation of this paper has been renamed and replaced by IBM Security Key Lifecycle Manager, Version 2.5. Most of the information pertaining to the Tivoli Key Lifecycle Manager as presented in this paper still applies.

Authors

This paper was produced by a team of specialists from around the world while working for the IBM International Technical Support Organization at the IBM European Storage Competence Center in Mainz, Germany.

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.

Dietmar Dausner is a certified XIV Product Field Engineer in Germany for the EMEA region. and joined IBM as a manufacturing test engineer for storage devices. Later he became a Client Application Engineer for hard disk drives, supporting large OEM accounts in Europe. In 2007, he joined the European Storage Competence Center as a Product Field Engineer (PFE). Since 2008, he has supported the XIV Storage System. Dietmar holds a degree in Electrical Engineering.

Roger Eriksson is an STG Lab Services consultant, based in Stockholm, Sweden, who works for the European Storage Competence Center in Mainz, Germany. He is a Senior Accredited IBM Product Service Professional. Roger has over 20 years of experience working on IBM servers and storage, including Enterprise and Midrange disk, NAS, SAN, IBM System x®, IBM System p®, and IBM BladeCenter®. He has done consulting, proof of concepts, and education, mainly with the XIV product line, since December 2008. He has worked with both clients and various IBM teams worldwide. He holds a technical college degree in Mechanical Engineering.

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.

Itzhack Goldberg is currently an IBM Technical Advisor in the EMEA region for the XIV Storage System, based in Haifa, Israel. Itzhack worked at the IBM Austin lab from 1989 to 1997, for the development of the IBM AIX® Logical Volume Manager and File System. He won an award for the design and development of a data recovery suite for AIX. Following that assignment, Itzhack worked on the code-load design and development of the IBM DS6000™. He holds a degree in Computer Science.

Markus Oscheka is an IT Specialist for Proof of Concepts and Benchmarks in the Disk Solution Europe team in Mainz, Germany. His areas of expertise include setup and demonstration of IBM System Storage solutions in various environments, such as AIX, Linux, Microsoft Windows, VMware ESX, and Solaris. He has performed many proofs of concept and performance benchmarks for disk storage products. He has contributed to and acted as co-project lead for DS8000® and XIV Redbooks publications. He holds a degree in Electrical Engineering from the Technical University in Darmstadt, Germany.

Stephen Solewin is an XIV Corporate Solutions Architect for IBM in Tucson, Arizona, US. He has 16 years of experience working on IBM storage, including Enterprise and Midrange Disk, LTO drives and libraries, SAN, storage virtualization, and storage software. Steve has been working on the XIV product line since March of 2008. He holds a BS degree in Electrical Engineering from the University of Arizona, where he graduated with honors.

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

Eyal Abraham, Diane Benjuya, Amy Blea, Ramy Buechler, Theodore Gregg, Rony Shapiro, Yossi Siles, George Thomas, Moshe Weiss
IBM

Now you can become a published author, too!

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

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

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

- Send your comments in an email to:

  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

1

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. As with all of its features, IBM XIV Storage System provides data-at-rest encryption at no charge. This gives clients the opportunity to take advantage of encryption and still enjoy the lower total cost of ownership that XIV offers.

Tip: The XIV system provides data-at-rest encryption at no charge. However, it requires a server and license for IBM Tivoli Key Lifecycle Manager (abbreviated as TKLM in paths and other instances).

This chapter gives an overview of encryption and covers the following topics:

1.1 Introduction to encryption

The IBM XIV Storage System offers a data-at-rest encryption solution that uses self-encrypting disks (SEDs) and flexible key manager software. When encryption is enabled, the optional solid-state drive (SSD) disks used as flash cache are also encrypted by using software-based AES 256-bit encryption. The IBM XIV 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 with XIV System Software Version 11.4 on XIV Gen 3 Model 214 systems, with 1 TB, 2 TB, 3 TB, and 4 TB drive capacities, as long as those drives are SED. All 4 TB drives available in the XIV are SED, but not all 1 TB, 2 TB, and 3 TB drives are SED. Any system ordered after 8 October 2013, regardless of capacity, has SED.

The XIV encryption solution requires at least one IBM Tivoli Key Lifecycle Manager server. For best data protection, it is better to have more than one key server installed, preferably in different locations. The Tivoli Key Lifecycle Manager server does not need to be dedicated to the XIV Storage System and can be shared across multiple products in the data center.

Uniquely among IBM products, the XIV Storage System offers *hot encryption*. When encryption is enabled on the XIV system, all data that resides on it is encrypted within minutes, with no performance impact.

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 functionality on those systems. All SED drives can operate transparently in non-encrypting systems. In software Version 11.4 or higher, by enabling encryption, they can provide the level of protection that you want without disruption.

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.
- Upon hot encryption, the flash cache is emptied, and the XIV must relearn the workload.
- It offers flexibility in the business decision-making process with the option to buy today and decide to apply later, when the need for encryption arises.

Important: Encryption must be deployed with careful planning and a full understanding of the interaction among the required products.

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 disk drives leave the company's premises, which usually happens when a disk drive fails and the IBM Service Support Representative replaces it with a new drive. Sometimes, drives are replaced proactively and the data can still be accessed. IBM has a procedure to delete all data on the drive; however, this task is no longer under the control of the client. Some clients buy back the drives and destroy them themselves, but this procedure can be quite expensive. A similar concern is when clients return the whole XIV Storage System to IBM. Of course, IBM erases all data, but this step is not sufficient for some clients. IBM offers a service 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).

All of these concerns become irrelevant when data on the drives is encrypted. Without the proper decryption key, the data on the drive or even on the entire XIV system is unreadable.

The question of what to encrypt and what to leave in clear text often arises. With overall system performance not affected by encryption and the low total cost of ownership provided by an XIV system, it might make 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: The IBM XIV Storage System provides disk-based encryption for data at rest on disk. If encryption over the network is required, additional encryption services need to be investigated and deployed, as appropriate.

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.

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 the IBM XIV system by separating the storage of a wrapped data key stored on the disk from the storage of the wrap or unwrap keys within a key server (Tivoli Key Lifecycle Manager).

- Key availability

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

- Separation of responsibilities

  The IBM XIV system offers 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. This ensures that one person cannot access the data, and it also ensures separation between the Security Administrator and Storage Administrator roles. IBM XIV also enables operation without a recovery key, but this is not recommended, 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 Tivoli Key Lifecycle Manager to help organizations efficiently deploy, back up, 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 be dependent on any storage system to which it provides keys. The key server must not store any of its code nor any information about keys that it manages for storage systems on that storage system.

If this consideration is not taken into account, it becomes possible to experience *encryption deadlock,* where a key server cannot function because it is dependent upon 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.

2

IBM XIV encryption architecture

This chapter describes the IBM XIV encryption architecture.

It covers the following topics:

2.1 IBM XIV disk encryption

IBM XIV Storage System Gen3 with software Version 11.4.0 and later for machine types 2810/2812, with self-encrypting drives (SEDs), helps secure data with industry-standard encryption for data at rest, without performance impact.

The IBM XIV Gen3 supports encryption for all capacities:

- IBM XIV Gen3 systems are available with SEDs in 1 TB (stripped-down 2 TB), 2 TB, 3 TB, and 4 TB, and software-based encryption of the flash cache.
- IBM XIV data-at-rest encryption is implemented with Advanced Encryption Standard (AES) 256-bit keys.
- The XIV software V11.4 can non-disruptively hot-encrypt SED-based IBM XIV Gen3 systems in minutes.
- XIV provides Key Management Interoperability Protocol (KMIP) Version 1.0 support.

Managing keys with IBM Tivoli Key Lifecycle Manager offers production-ready key management (abbreviated as TKLM in paths and file names). This offers two advantages:

- Separated, centralized, and simplified key management
- Separation of key storage from data storage

The SED uses a symmetric data key to encrypt and decrypt data. The symmetric data key is not available in plain text when the IBM XIV system and the Tivoli Key Lifecycle Manager communicate. For details, see 2.2, “Encryption techniques used in XIV encryption” on page 11.

Security Administrator role

An IBM XIV system with software Version 11.4 or later has a new user role, called Security Administrator. This person is the only one who has authority to configure and enable encryption. The Security Administrator cannot reach any other menu items, such as Volume view or Create.

Activate encryption

You can configure an IBM XIV system with SEDs to enable encryption, and then all data stored on the IBM XIV will be encrypted. You can activate encryption concurrently with data already stored in the IBM XIV system. This capability is referred to as *hot encryption*. When hot encryption is started, data in the flash cache is erased. Therefore, after encryption is finished, the flash cache needs to start “learning” again.

> **Important:** Deactivating encryption cryptographically erases all data on the drives. Therefore, you must back up any data that must be kept, or migrate it to another system, before deactivating encryption on the XIV system.

Copy Services function considerations

If volumes on an encrypted IBM XIV system are mirrored to a non-encrypted IBM XIV, the data will not be encrypted on the target IBM XIV. Therefore, it will not be secured. Even if the target IXV system encryption is activated, the data will not be encrypted when it is transferred between the two XIV systems unless you take suitable measures to protect data in transit.

Tivoli Key Lifecycle Manager

The IBM XIV system must be configured to communicate with *at least one* IBM Tivoli Key Lifecycle Manager server to enable encryption. At least two servers are better, for redundancy.

Important: The IBM Tivoli Key Lifecycle Manager server can be installed as a virtual machine. In that case, make sure that it does not use the encrypted IBM XIV system as a storage device. Such configuration can lead to an encryption deadlock situation, where a Tivoli Key Lifecycle Manager server cannot function because it is dependent on storage that cannot release data because it needs to communicate with that same server.

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

Important: If the Tivoli Key Lifecycle Manager server is not reachable when an IBM XIV (with encryption activated) is powering up (or is rebooted), the IBM XIV will not be accessible to read or write data for the hosts. This is why it is important to have at least two Tivoli Key Lifecycle Manager servers on your IP network.

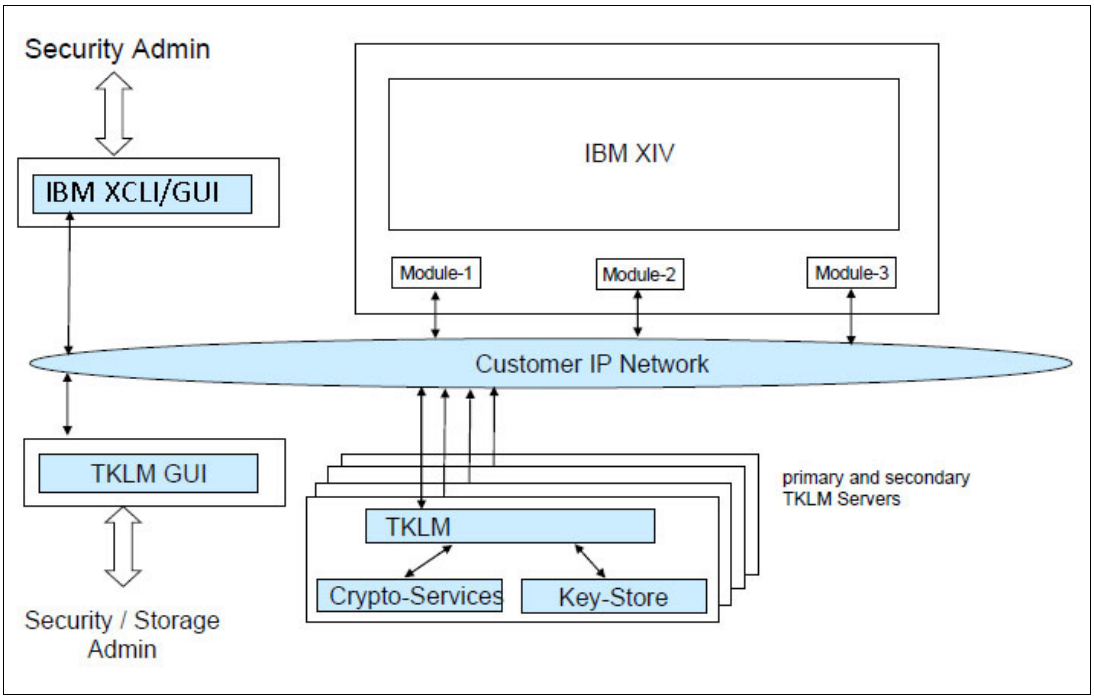

Figure 2-1 Connection between IBM XIV and Tivoli Key Lifecycle Manager (TKLM)

2.1.1 Self-encrypting drives

The IBM XIV system supports data encryption with the self-encrypting drives (SEDs). All disks in the IBM XIV 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 IBM XIVs with SEDs can also be used without encryption 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 under “Enrolling” on page 11. 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 SED in the IBM XIV system, Safe Drive Retirement 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 mentioned in 2.1, “IBM XIV disk encryption” on page 8.

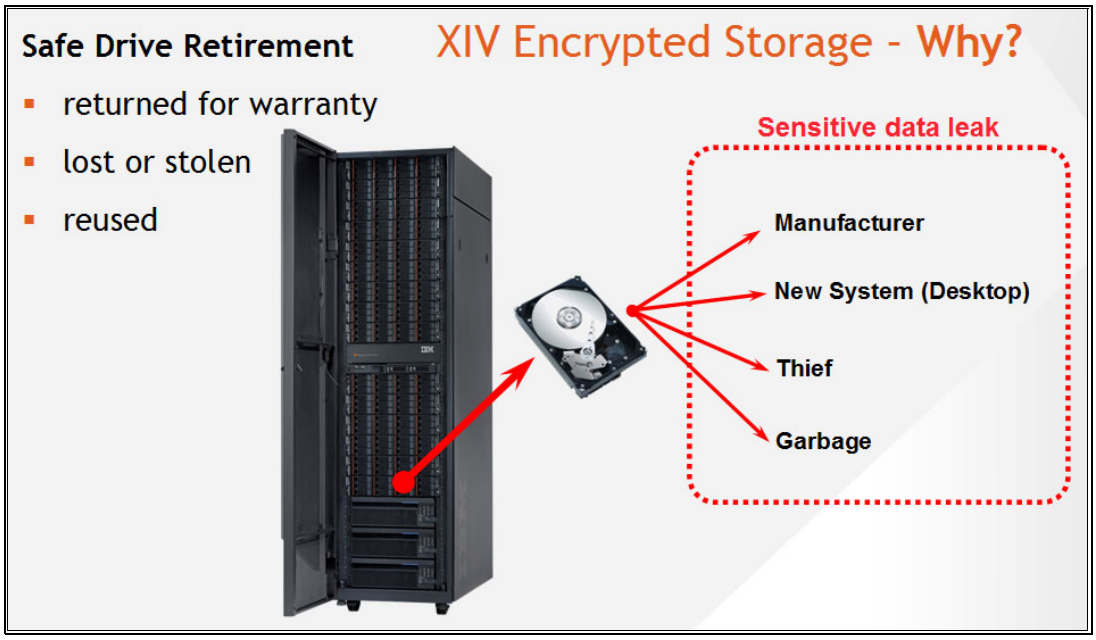

Figure 2-2 Safe Drive Retirement

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 cannot 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 XIV 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 previously stored in the band.

Enrolling

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

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

Before enrollment, the DEK is encrypted with the Manufactured Secure ID (MSID), which is a hard-coded 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.

When a disk’s band is enrolled, the band becomes unreadable, or locked, during power-up 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 IBM XIV software is responsible for providing each drive with its DAK.

Figure 2-3 illustrates the enrolling process, showing the authentication and encryption key relationship.

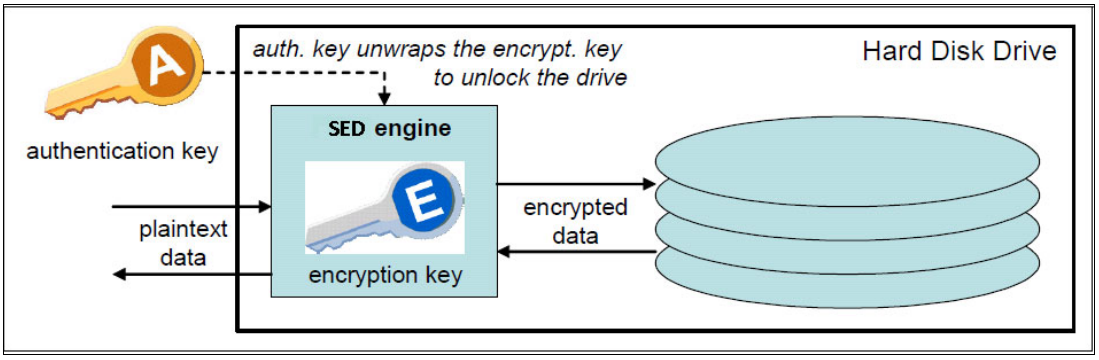

Figure 2-3 Enrolling SED

2.2 Encryption techniques used in XIV encryption

IBM XIV encryption uses 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 2-4 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 significantly faster than asymmetric encryption algorithms. This makes symmetric encryption ideal for encrypting large amounts of data.

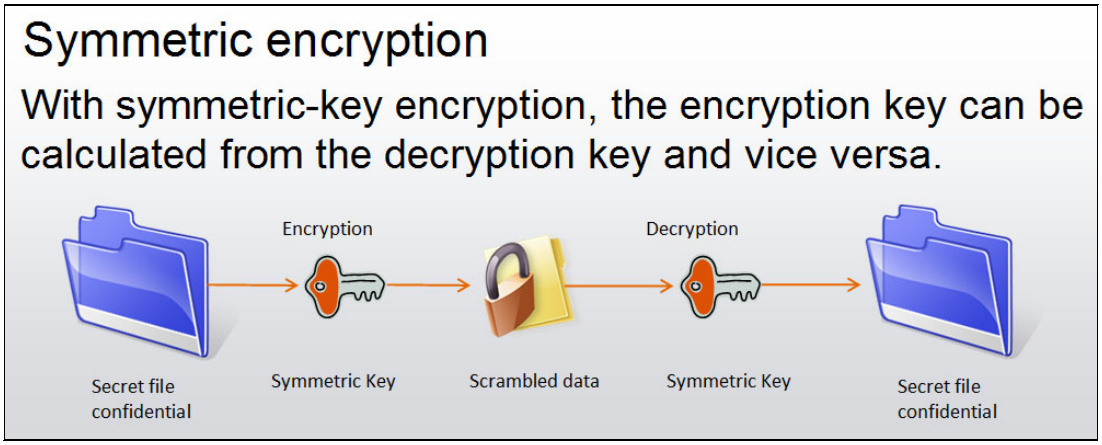

Figure 2-4 Symmetric encryption

2.2.1 Digital certificates

Digital certificates are a way to bind information with an identity. Digital certificates are exchanged between the Tivoli Key Lifecycle Manager and XIV system so that each can verify the other's identity before sending the sensitive keying information. This is to make sure that the sender can trust the receiver.

The certificates are 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 system has a unique digital certificate installed at the time of manufacture. In addition, users can install their own digital certificates if they choose.

3

Planning

This chapter explains the planning of data-at-rest encryption with the IBM XIV Storage System and Tivoli Key Lifecycle Manager.

It covers these topics:

3.1 Planning and implementation process flow

The diagram in Figure 3-1 shows the planning and implementation process for a data-at-rest encryption-capable IBM XIV system. The details for this process are described in subsequent sections of this chapter. Also, see Chapter 4, “Configuring and implementing XIV encryption” on page 19. The diagram in Figure 3-1 shows the overall decision flow and outcomes.

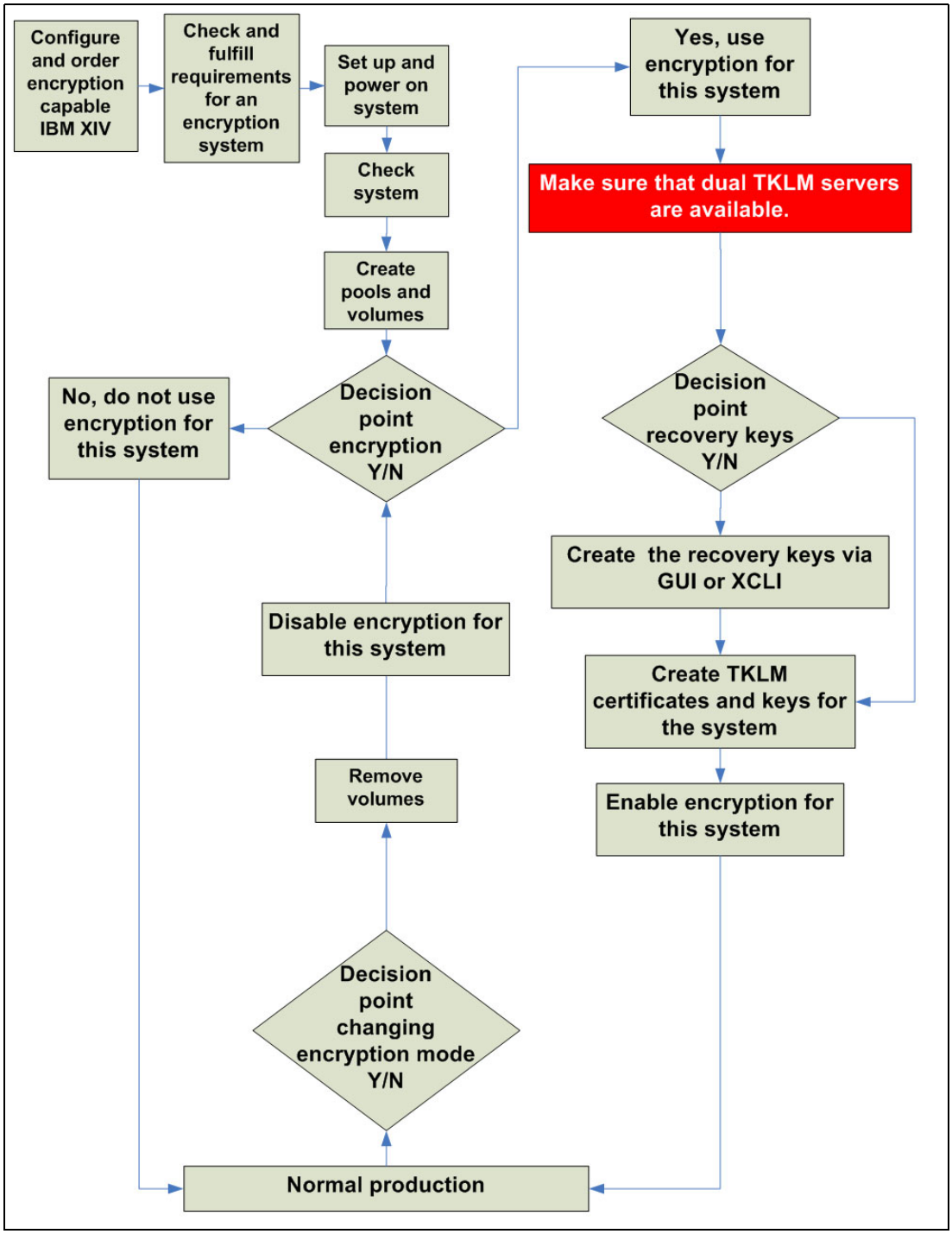

Figure 3-1 Implementation process flow

3.2 Required and optional tasks

After the Tivoli Key Lifecycle Manager installation, a few tasks are required to implement and activate encryption with IBM XIV storage.

To deploy an encryption-capable IBM XIV, the following requirements must be strictly respected:

- Configuring the recovery key is the strongly advised first step. The key is generated from the XIV GUI or XIV Storage System command-line interface (XCLI).
- Any IBM XIV that is encryption-enabled must be configured to connect to at least one key server.

  The key server can be a separately purchased hardware product that is able to support the IBM Tivoli Key Lifecycle Manager. Clients must acquire a license for use of the Tivoli Key Lifecycle Manager software that is ordered separately from the XIV system. For details, consult Chapter 2 of the *IBM Tivoli Key Lifecycle Manager V2.0.1: Installation and Configuration Guide*, SC27-2741:

  http://bit.ly/ZaWaWZ

  As previously indicated (see “Tivoli Key Lifecycle Manager” on page 9), the Tivoli Key Lifecycle Manager server can be installed as a virtual machine. In this case, make sure that it does not use the encrypted IBM XIV as the storage device.

  An encryption-enabled IBM XIV 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 Tivoli Key Lifecycle Manager supports, including other encryption-enabled IBM XIV systems or supported
  IBM tape drives.

3.3 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.

3.3.1 Security

These are the considerations and preferred practices:

- General:

  Ideally, a good 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. With a basic security policy, having a single person who handles the *storageadmin* and *secadmin* role of an IBM XIV is still possible. With LDAP, you can set up a policy in the IBM XIV that does not allow the same user ID for both roles.
- Keystore:

  During setup of the Tivoli Key Lifecycle Manager key server, a password is specified for access to the keystore. Clients must decide whether the Tivoli Key Lifecycle Manager password will be provided manually or whether a mechanism is in place to automatically provide the password to the Tivoli Key Lifecycle Manager. If a startup script that contains

the password is used at the Tivoli Key Lifecycle Manager key server, the script file must have access controls to prevent unauthorized access to the file and password.

3.3.2 Availability

These are the considerations and preferred practices:

- IBM XIV system:

  The IBM XIV must be configured with all three management modules' IPs to provide redundant access to the client's network.

- Tivoli Key Lifecycle Manager key server:

  Note the following information:

  - Configure redundant key servers to each encrypting storage device. Have independent and redundant key servers on each site.
  - To initiate the Tivoli 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.

3.3.3 Encryption administration

These are the considerations and preferred practices:

- General:

  Note the following information:

  - The change management processes at the client installation must cover any procedures necessary to ensure adherence to guidelines required to ensure correct configuration of key servers and encrypted storage.
  - At least annually, all personnel who have any of the following assignments or capabilities are required to review a client document that describes these risks and the processes adopted to mitigate them:
    - Responsibility for the implementation of Tivoli Key Lifecycle Manager key servers or encrypted storage products
    - Responsibility to manage the placement or relocation of data related to or required by any Tivoli Key Lifecycle Manager key server
    - Access authority to configure Tivoli Key Lifecycle Manager key servers or encrypted storage products
    - Responsibility to rekey the recovery key of the IBM XIV, if used
  - The client must implement automated monitoring of the availability of any equipment that is associated with management of key services and take appropriate action to keep them operational. This equipment can include but is not limited to key servers, SNMP managers, 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 good 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 IBM XIV. This key is the last resort to unlock the IBM XIV if the Tivoli Key Lifecycle Manager environment is destroyed or

totally inaccessible. The recovery key is *not* used while Tivoli Key Lifecycle Manager remains available.

- Tivoli Key Lifecycle Manager key server:

  Consider the following information:

  – 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:** IBM XIV requires at least one key server to be configured, but a good 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 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 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 good practice and reduces the probability that all key servers will experience a simultaneous power loss.

  – Clients must ensure that all key servers that a particular storage device is configured to communicate with have a consistent keystore content relative to any wrapping keys that will be 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 is dependent on a key server. See 5.1, “Backup and restore procedures” on page 50.

  – 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.

- IBM XIV

  Note the following information:

  - The IBM XIV monitors all configured Tivoli Key Lifecycle Manager key servers. Customer notification is provided through the IBM XIV 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 IBM XIV 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 IBM XIV that has encryption activated is inaccessible to attached hosts. If the IBM XIV is able to 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 IBM XIV reboot is required to make the data accessible.
    - The ability of each key server to serve data keys configured on the IBM XIV is verified at daily intervals. Loss of the ability to unwrap a configured data key is reported to the client and to IBM.

3.4 Multiple Tivoli Key Lifecycle Managers for redundancy

To ensure continuous key and certificate availability to encrypting devices, configure a primary and a replica Tivoli Key Lifecycle Manager server 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 AIX, both computers must have the required memory, speed, and available disk space to meet the workload.

Note that this is not a failover or clustered server from a Tivoli Key Lifecycle Manager point of view. The redundancy is managed by setting up multiple key manager destinations at the IBM XIV storage server.

Synchronization is achieved by backing up one server and restoring 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

3.4.1 Setting up Tivoli Key Lifecycle Manager servers

A good practice is to complete the pre-installation worksheets that are available in Appendix A of the *Tivoli Key Lifecycle Manager Installation and Configuration Guide*, SC27-2741.

For other information, see the Tivoli Key Lifecycle Manager Information Center:

http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tklm.doc_2.0.1/welcome.htm

4

Configuring and implementing XIV encryption

This chapter describes how to configure and implement data-at-rest encryption for the IBM XIV Storage System. It covers the following topics:

Note: Several illustrations in this chapter are based on Version 2.0.1 of the Tivoli Key Lifecycle Manager GUI. It was the version supported when we were writing this Redpaper.

4.1 Encryption process overview

The IBM XIV data-at-rest encryption initial configuration starts with installing and configuring the external key server. In our testing, we used the Tivoli Key Lifecycle Manager server Version 2.0.1, which was the version officially supported at the time of writing this paper.

After the key server is installed and configured, the IBM XIV and the key server must be able to connect to one another. They establish a trusted connection by exchanging their certificates, as explained further in 4.3.1, “Overview of configuration steps” on page 22. Then, the IBM XIV generates a random XIV master key (XMK), which is used to create the Disk Access Keys (DAK). Next, the IBM XIV requests and receives the externally stored key (ESK) from the key server. The ESK is used to wrap (encrypt) the XIV master key that is stored in the IBM XIV. Figure 4-1 illustrates the process.

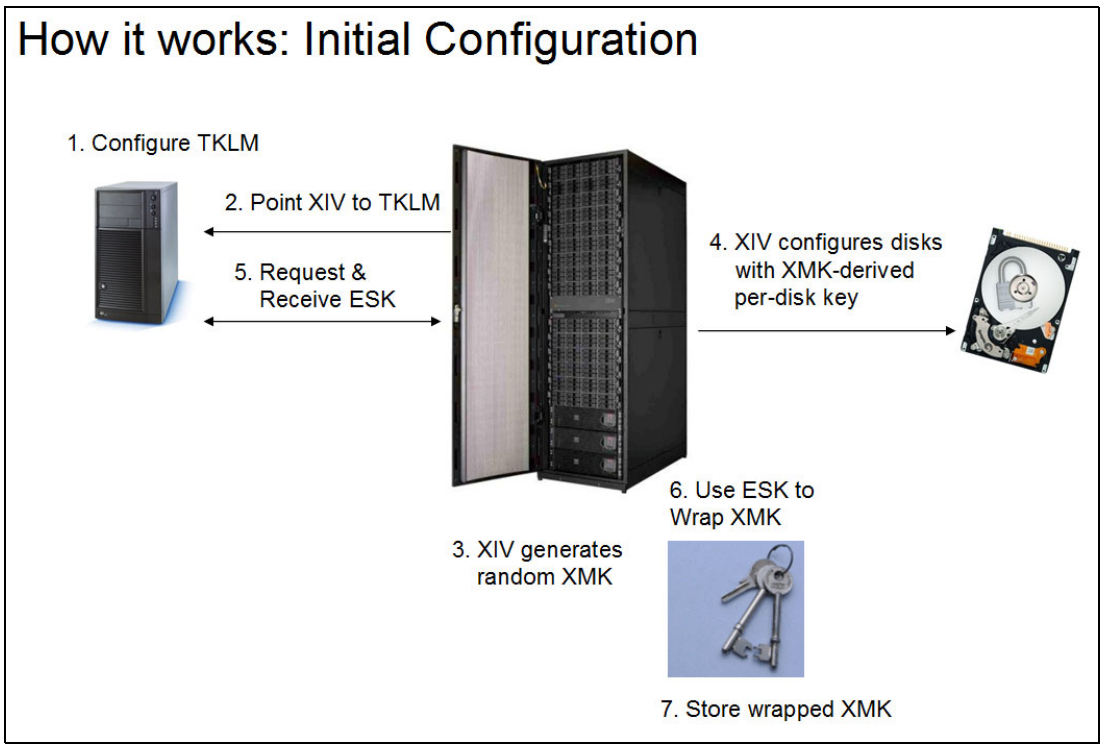

Figure 4-1 Initial configuration

After the XIV data-at-rest encryption is activated and booting (after power maintenance, for example), the main encryption startup sequence is as shown in Figure 4-2 on page 21. The self-encrypting disks (SEDs) are locked during a reboot. Therefore, you need a valid connection to the key server. When the XIV is booting, it establishes a Secure Sockets Layer (SSL) tunnel based on the Key Management Interoperability Protocol (KMIP) if the certificates on the IBM XIV and on the key server match. The XIV then requests the externally stored key, which the key server provides. It is used to unwrap the XIV master key to derive the Disk Access Keys that unlock the self-encrypted drives and the encryption-activated Flash cache.

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

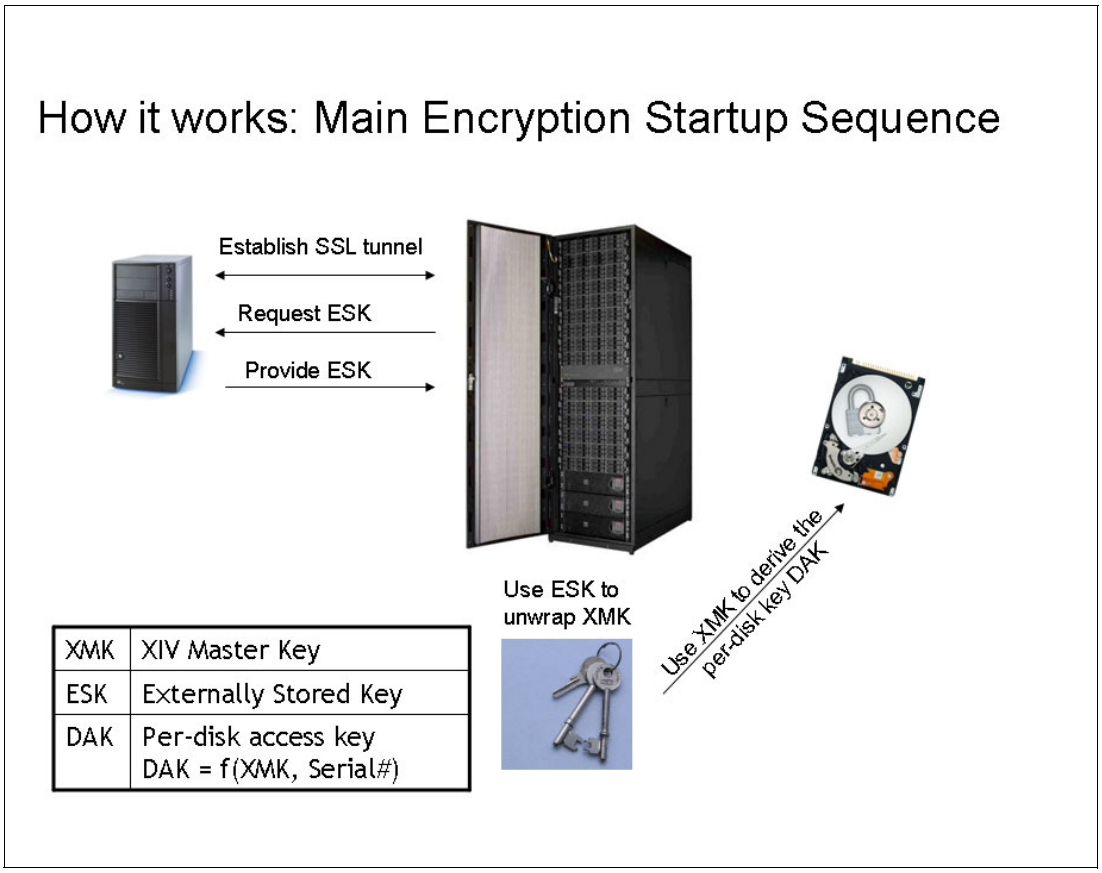

Figure 4-2 Encryption startup sequence

4.2 Tivoli Key Lifecycle Manager installation

At the time of writing this document, Tivoli Key Lifecycle Manager Version 2.0.1 was the supported key manager for use with XIV.

You can find installation instructions online in the IBM Tivoli Key Lifecycle Manager Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tklm.doc_2.0.1/welcome.htm

You can also download the IBM publication titled *IBM Tivoli Key Lifecycle Manager V2.0.1: Installation and Configuration Guide*, SC27-2741:

http://bit.ly/ZaWaWZ

4.3 IBM XIV data-at-rest encryption configuration

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

4.3.1 Overview of configuration steps

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

1. Log in to the Tivoli Key Lifecycle Manager console.
2. Create the Tivoli Key Lifecycle Manager master keystore.
3. Manage certificates:
   a. Copy the XIV device-specific certificate from XIV and add the Tivoli Key Lifecycle Manager (import certificate).
   b. Create a Tivoli Key Lifecycle Manager self-signed certificate on the Tivoli Key Lifecycle Manager GUI (add the KMIP-based SSL certificate).
   c. Export the Tivoli Key Lifecycle Manager certificate.
4. Define the Tivoli Key Lifecycle Manager key server on IBM XIV (import the `cert.pem` Tivoli Key Lifecycle Manager Certificate).
5. Create the IBM XIV device group.
6. Create the IBM XIV device in the IBM XIV device group (DS5000 group type).

Detailed instructions follow.

4.3.2 Detailed configuration steps

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

Step 1. Log in to the Tivoli Key Lifecycle Manager console

The Tivoli Key Lifecycle Manager solution incorporates the *Tivoli Integrated Portal* installation manager, which provides simple-to-use installation options and a management console. To manage and configure the Tivoli Key Lifecycle Manager, log in to the Tivoli Integrated Portal.

Open a web browser, and specify the IP address and IP port of the Tivoli Key Lifecycle Manager by using this web address format:

`https://<TKLM_IP>:<TKLM_PORT>/ibm/TKLM/login.jsp`

Figure 4-3 on page 23 shows the Tivoli Integrated Portal login window.

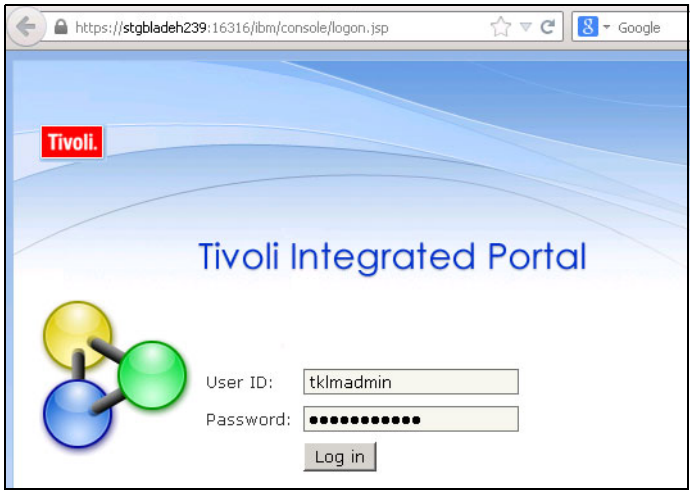

Figure 4-3 Tivoli Integrated Portal login window

Enter your Tivoli Key Lifecycle Manager admin user ID and password, and then click **Log in**.

The Welcome panel, shown in Figure 4-4, indicates that you are successfully logged in to the Tivoli Integrated Portal.

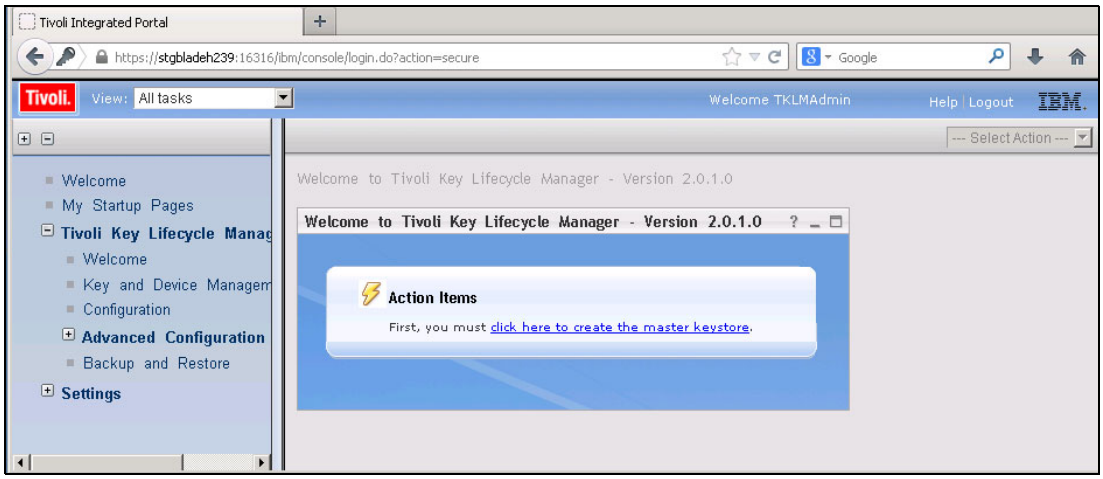

Figure 4-4 Welcome panel

Step 2. Create a Tivoli Key Lifecycle Manager master keystore

If this is a new Tivoli Key Lifecycle Manager installation, you must create the master keystore. This keystore holds all keys and certificates that are managed by Tivoli Key Lifecycle Manager.

You can use the link in the Action Items section of the Welcome window (Figure 4-4 on page 23) to create the master keystore.

Complete the following tasks:

1. When you click the **Create the master keystore** link, the Keystore window in Figure 4-5 opens. Select the **JCEKS** keystore type, which is the only choice when the Tivoli Key Lifecycle Manager server is installed on an open systems platform. The JCEKS software keystore type supports asymmetric and symmetric keys.

   Enter the values shown in these fields:

   Keystore path: `C:\ibm\tivoli\tiptklmV2\products\tklm\keystore`

   Keystore name: `default_Keystore_xiv`

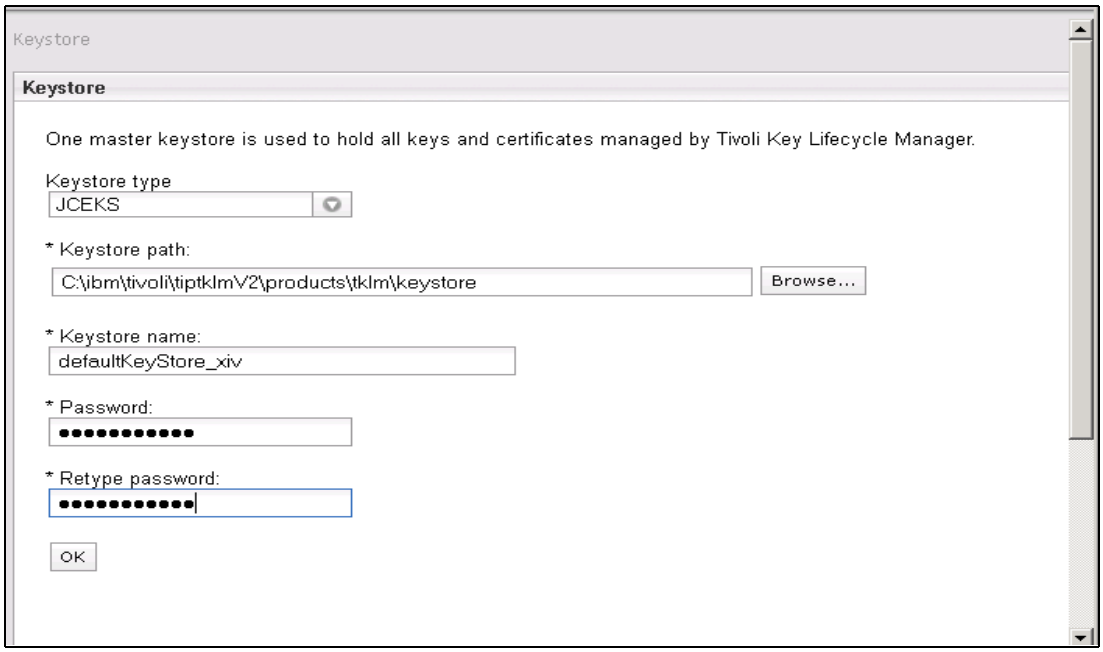

Figure 4-5 Create keystore

2. Specify a password. This password is set as the master keystore password, so it is also referred to as the Tivoli Key Lifecycle Manager *master key*. It is the key to all other keys that are maintained by the Tivoli Key Lifecycle Manager keystore.

   Lost password: Losing the password results in not being able to transfer any certificate from this Tivoli Key Lifecycle Manager server to another Tivoli Key Lifecycle Manager server.

3. After you select the keystore type and provide all other required information, click **OK**. The Welcome window that is shown in Figure 4-6 on page 25 opens and confirms successful creation of the keystore.

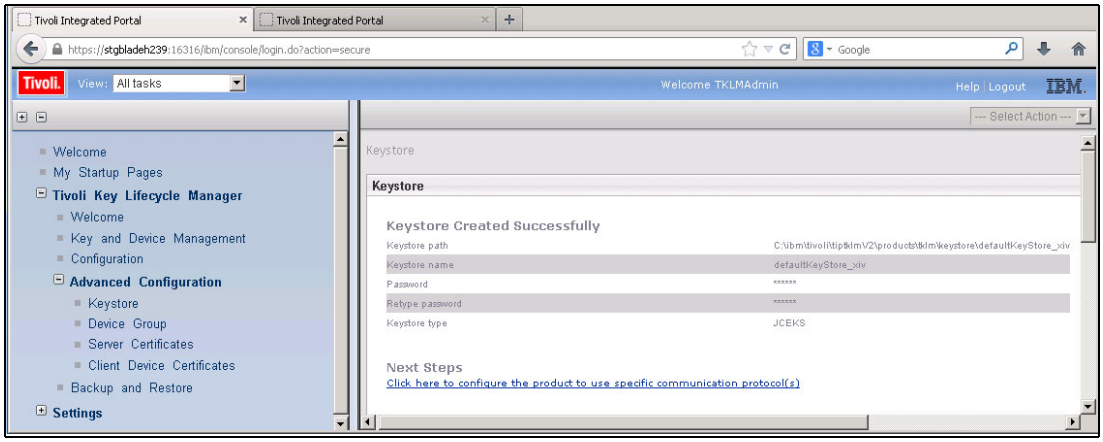

Figure 4-6 Keystore created successfully

Step 3. Manage certificates

As already mentioned, Version 11.4 of IBM XIV software introduces a new Security Administrator role. You must log on to your XIV system as Security Administrator to complete the following tasks:

1. First, copy the IBM XIV device-specific certificate from the IBM XIV, and add it to Tivoli Key Lifecycle Manager by exporting the certificate. You can do this either in the IBM XIV GUI or in the IBM XCLI. If you decide to do it in the IBM XIV GUI, log in as Security Administrator and select **Systems** → **System Settings** → **Manage Certificates**, as Figure 4-7 shows.

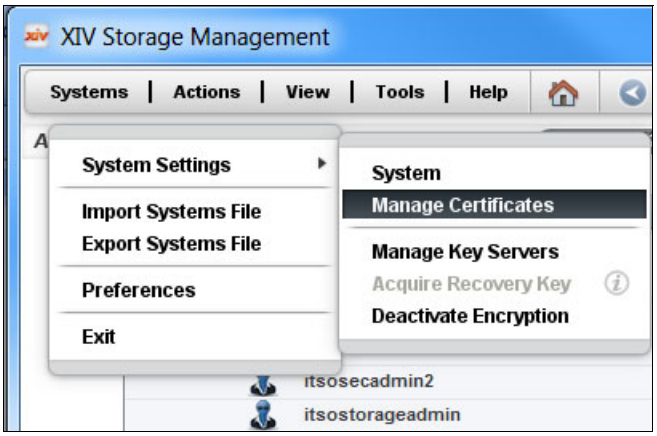

Figure 4-7 Manage Certificates

2. Select your default XIV certificate, and click the **Export Certificate** icon, as shown in Figure 4-8, and save it as your `XIV_KMIP.pem` file.

Figure 4-8 Export Certificate action

If you prefer, you can perform this action manually by using the XCLI for these steps:

Open an XCLI session as Security Administrator. Run the pki_list command, which shows the IBM XIV default device-specific certificate that was installed during manufacturing. Using the name that you get in the output where this example says <*default certificate*>, run **`pki_show_certificate name=<default_certificate>`**, as shown in Figure 4-9.

```
XIV 1310092 Cona>>pki_list
Name   Fingerprint                          Has signed certificate   Services
XIV    11581b945aad29695416fff069ecf6fa     yes                      KMIP
XIV 1310092 Cona>>pki_show_certificate name=XIV
Certificate
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 98 (0x62)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=ibmXIVDisk
        Validity
            Not Before: May 28 19:18:42 2013 GMT
            Not After : Nov 16 00:22:12 2032 GMT
        Subject: C=US, O=ibmXIVDisk, CN=2810-1310092
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d0:b0:4c:72:cd:b9:04:d1:64:48:fa:68:e2:49:
                    3b:12:84:18:fa:aa:03:cd:fe:8d:c5:03:96:f6:7d:
                    44:31:cc:20:b3:19:01:83:d0:28:b6:ee:d5:2b:17:
                    6a:c5:32:77:cf:9e:fc:cf:5c:60:6e:79:ba:79:12:
```

Figure 4-9 pki_list and pki_show_certificate

Copy and paste the portion that includes "----BEGIN CERTIFICATE----" and "----END CERTIFICATE----" into a 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.

3. The next step is to import the newly created SSL certificate as "trusted" in the Tivoli Key Lifecycle Manager web GUI. Select **Advanced Configuration** → **Client Certificates**, and click **Import** (under the section headed "SSL/KMIP Certificate for Clients") as shown in Figure 4-10 on page 27 and in Figure 4-11 on page 27.

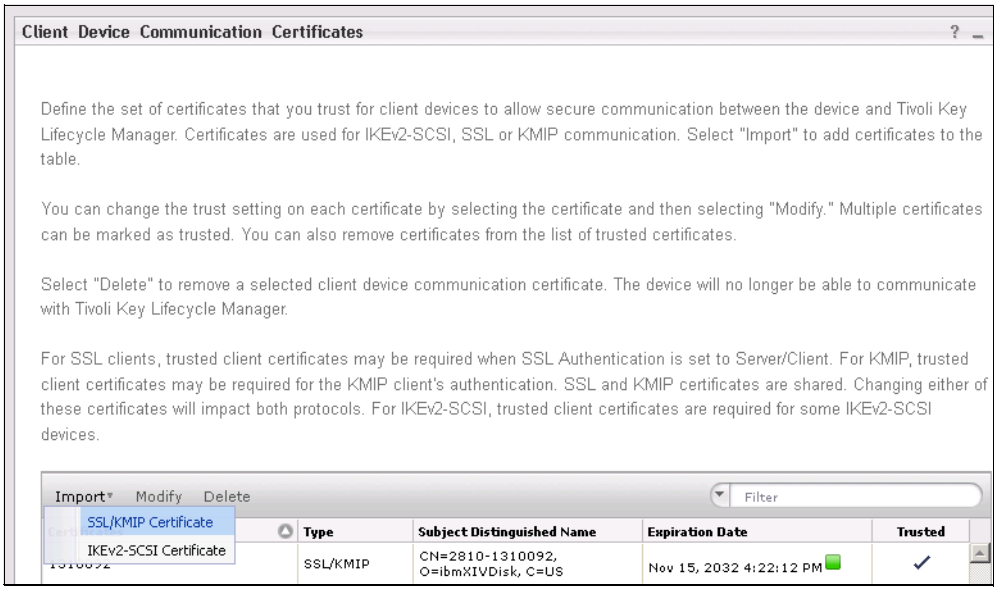

Figure 4-10 Client Device Communication Certificates display

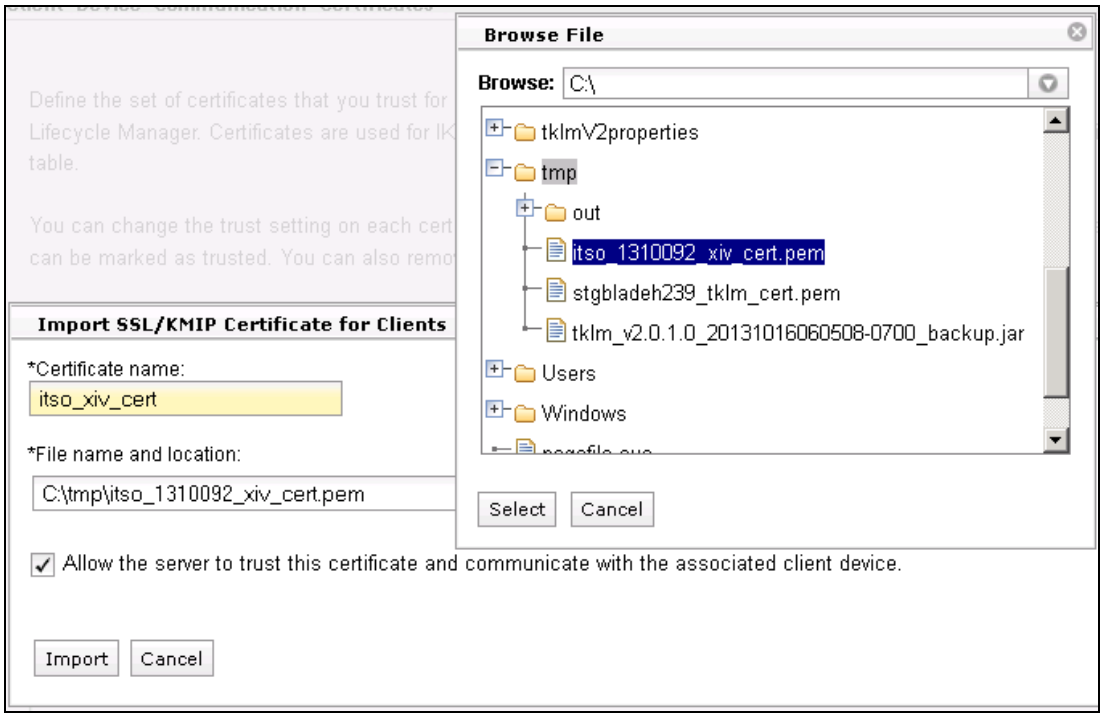

Figure 4-11 Import SSL/KMIP Certificate for Clients

4. Create a Tivoli Key Lifecycle Manager self-signed certificate in Tivoli Key Lifecycle Manager GUI (add SSL/KMIP certificate):

   a. In the window that is shown in Figure 4-6 on page 25, under Next Steps, click **Configure the product to use specific communication protocols**.

b. When the window that is shown in Figure 4-12 opens, create the certificate that is used to encrypt data for secure communication over SSL.

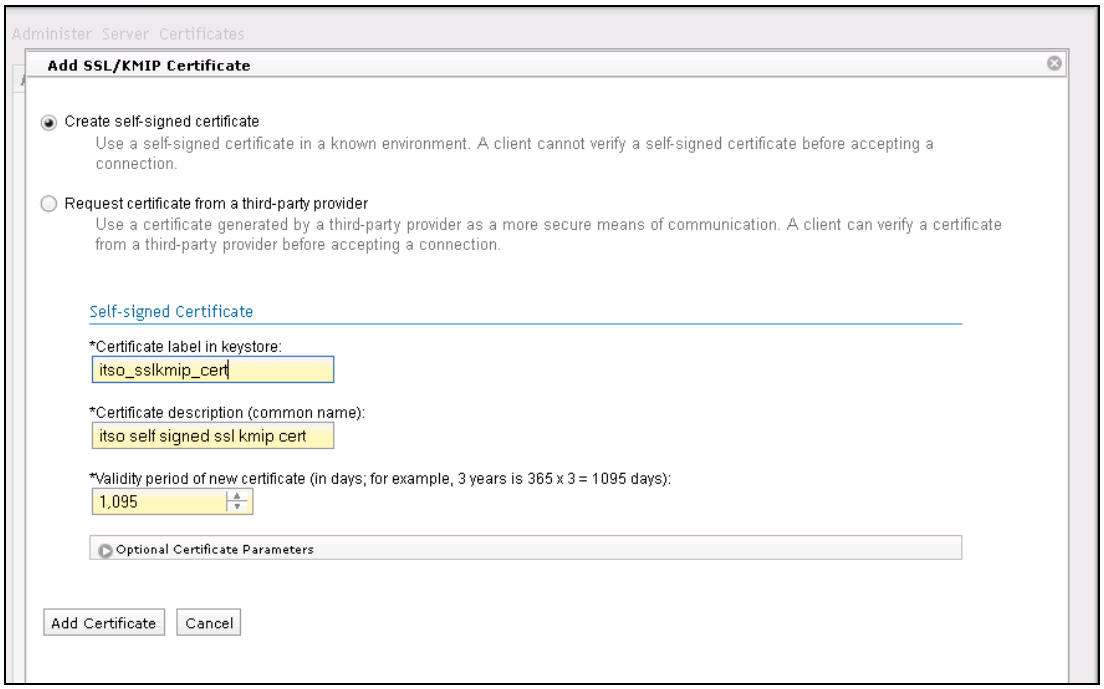

Figure 4-12 Add SSL/KMIP Certificate

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

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

Caution: Although using an existing certificate from the keystore is possible, using a certificate that is used for encrypting disk data to protect the communication protocol also is *not* a good practice.

d. Choose a descriptive label and a certificate expiration validity in days in accordance with your security guidelines. You also have the option to enter certificate parameters.

e. Click **Add Certificate**.

As indicated in the Warning notice shown in Figure 4-13, the SSL/KMIP certificate is updated. For this change to take effect, you must restart the server. Also, create a backup to ensure that you can restore this data.

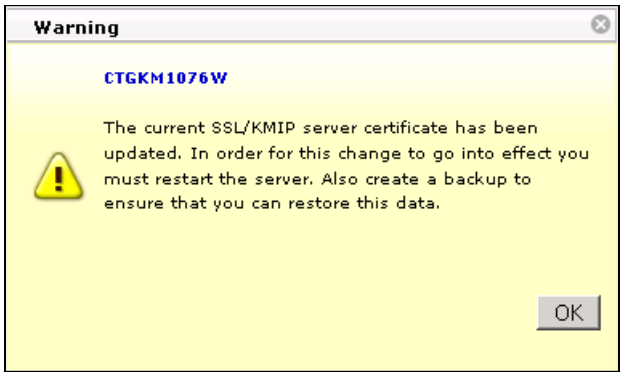

Figure 4-13 Reminder to restart the server

5. In the left pane of the Tivoli Key Lifecycle Manager web interface, click **Welcome** to return to the Welcome window shown in Figure 4-14.

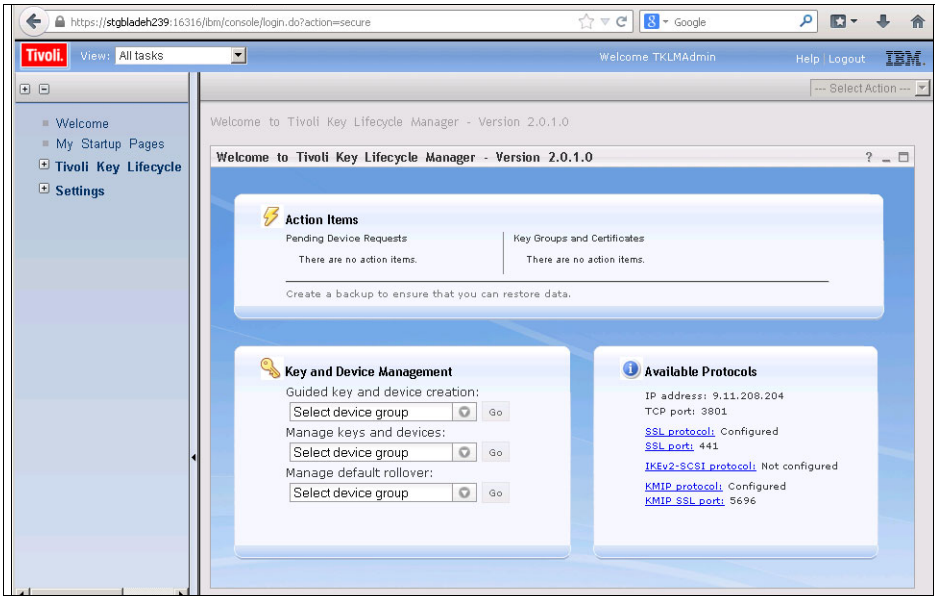

Figure 4-14 Welcome window

You have now created the Tivoli Key Lifecycle Manager master keystore 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. At the time of writing this paper, you can do this only through the **`wsadmin`** CLI.

6. Export Tivoli Key Lifecycle Manager certificate:
   - For a Microsoft Windows operating system, open a DOS prompt with Administrator privileges. Enter the following **wsadmin** command:

     cd <*TKLM PATH*> wsadmin -username tklmadmin -password <*tklmadmin password*> -lang jython

     where <*TKLM PATH*> is, for example: `C:\ibm\tivoli\tiptklmV2\bin`
   - For a Linux operating system, open UNIX terminal session, and enter this command:

     cd <TKLM PATH>rm -f /tmp/cert.der
     ./wsadmin.sh -username TKLMAdmin -password <tklmadmin password> -lang jython

     where <*TKLM PATH*> is, for example: `/opt/IBM/tivoli/tiptklmV2/bin`

7. To view all existing certificates, use the **print AdminTask.tklmCertList()** command shown in Figure 4-15.

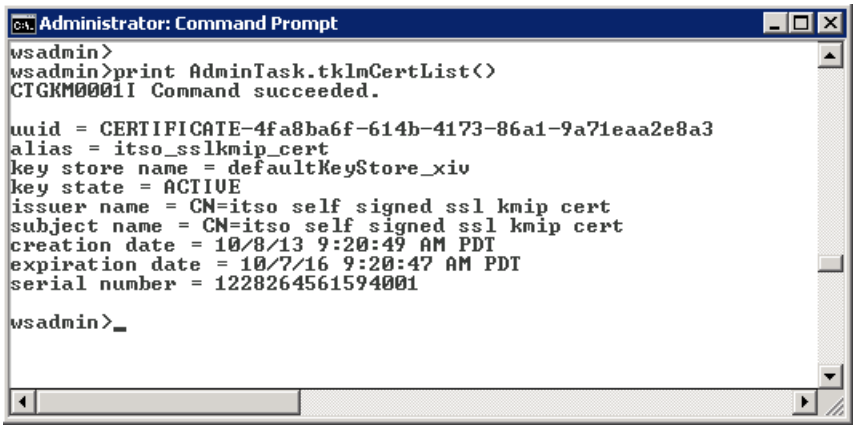

Figure 4-15 tklmCertList

8. Print the certificate created in Step 2 by typing the **print** command:

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

   Figure 4-16 shows an example of the result.

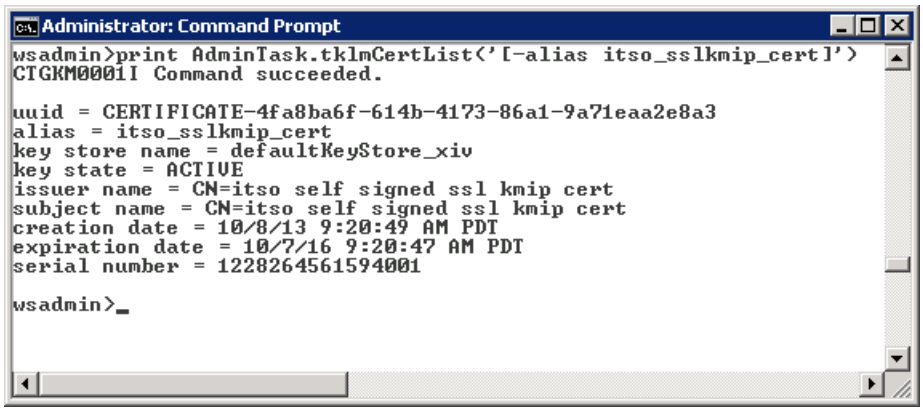

Figure 4-16 tklmCertList with specific alias

9. Take the Universally Unique Identifier (UUID) information from the output of Step 8, and use that 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 Tivoli Key Lifecycle Manager installation directory. On a Windows server, you can find it in this path:

   ```
   C:\ibm\tivoli\tiptklmV2\products\tklm\tmp\cert.der
   ```

   Export the certification file by using this 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 next in Step 4).

Step 4. Define the key server on the XIV system

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

By using the XIV GUI, log in as Security Administrator.

Select **Systems** → **System Settings** → **Manage Key Servers** and click the plus sign (**+**) icon.

Enter the name for your Tivoli Key Lifecycle Manager server, the Tivoli Key Lifecycle Manager server address by IP or DNS name, and choose the Tivoli Key Lifecycle Manager certificate that you generated previously, as shown in Figure 4-17.

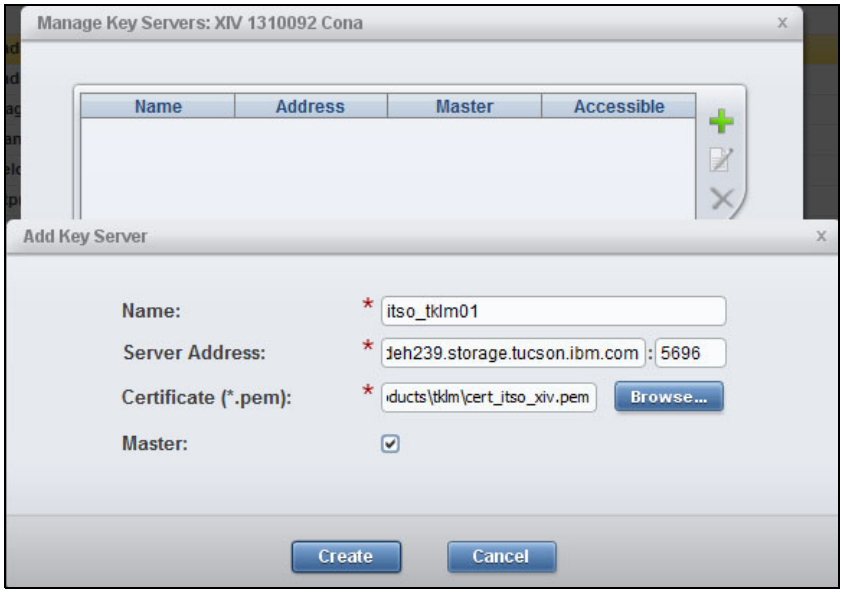

Figure 4-17 Add Key Server window

Click **Create**. The Manage Key Servers dialog in Figure 4-18 indicates that the key server is now trying to establish the connection with the Tivoli Key Lifecycle Manager server.

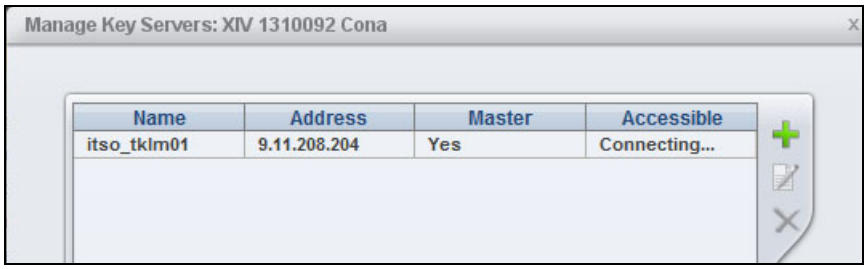

Figure 4-18 Manage Key Servers

After the connection is established, the Accessible column displays *Yes* as shown in Figure 4-19.

Figure 4-19 Added Key Server

Step 5. Create the XIV device group

XIV device groups are part of the native Tivoli Key Lifecycle Manager distribution package in newer versions of the Tivoli Key Lifecycle Manager server. Therefore, this step is not necessary with newer versions, because the XIV device group is predefined. However, with Tivoli Key Lifecycle Manager Version 2.0.1, it must be created manually, and it inherits its characteristics from the DS5000 family.

Follow these steps to create the device group if you are using Version 2.0.1:

1. Log in to the Tivoli Key Lifecycle Manager GUI as `TKLMAdmin`, and select **Tivoli Key Lifecycle Manager** → **Advanced Configuration** → **Device Group**. Click **Create**, as shown in Figure 4-20.

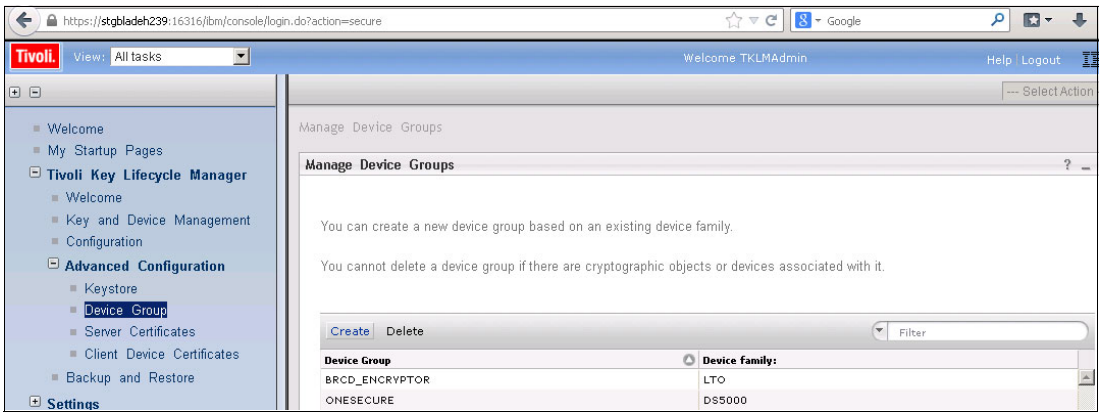

Figure 4-20 Create Device Group

2. Set “Device family” to **DS5000** and “Device group name” to XIV, and then click **Create**, as shown in Figure 4-21.

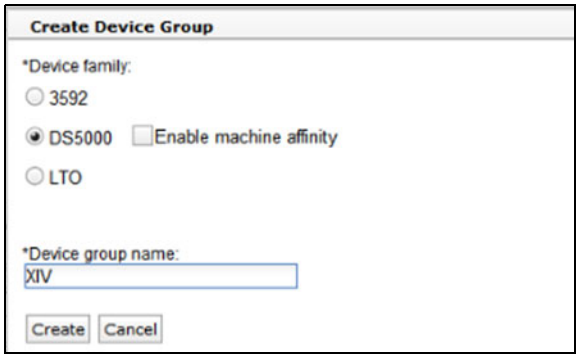

Figure 4-21 Create Device Group

By default, initial requests to communicate for devices that belong to this group are held until approved in the Tivoli Key Lifecycle Manager web GUI. As Figure 4-22 shows, a message displays this information.

Figure 4-22 Create Device Group Information

Step 6. Create the XIV device

Set the *device.AutoPendingAutoDiscovery* attribute to a value that adds incoming devices to the pending devices list.

Specify a setting such as 2 (auto pending). All incoming devices are added to a pending list, but are not automatically served keys upon request. You must accept or reject a device in the pending devices list on the Tivoli Key Lifecycle Manager server before the device is served keys upon request.

> **Note:** Do not use a setting of 1 (auto accept) for the DS5000 device family (XIV), because this setting allows generation and serving of keys to DS5000 storage servers before you can perform a backup.

Complete these tasks to create the IBM XIV device:

1. Log in to wsadmin:

```
wsadmin -username TKLMAdmin -password mypwd -lang jython
```

2. Issue the following command:

```
print AdminTask.tklmDeviceGroupAttributeUpdate ('[-name DS5000 -attributes
"{device.AutoPendingAutoDiscovery 2}"]')
```

You do not need to create the IBM XIV device manually if the “Hold pending requests until approved” check mark is selected for the IBM XIV device group. If you choose to configure a device this way, the first attempt to run **encrypt_enable**, as described in 4.5.1, “Activate data-at-rest XIV encryption” on page 43, fails. A pending request shows in the Welcome window of the Tivoli Key Lifecycle Manager web GUI, as illustrated in Figure 4-23.

Figure 4-23 Pending devices

4.4 Recovery key use and maintenance

To protect against the possibility (following a disaster, for example) that all Tivoli Key Lifecycle Managers become unusable and unrecoverable, XIV enables you to create a *recovery key* as depicted in Figure 4-24 on page 35. With a recovery key, Security Administrators can unlock an IBM XIV without involvement of a Tivoli Key Lifecycle Manager server. Encryption can be activated either in the IBM XIV GUI or through the IBM XCLI. If that action is through the IBM XIV GUI, the recovery key is mandatory. The option to activate data-at-rest encryption without recovery keys is possible but only through the IBM XCLI by using 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 XIV master key (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 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 IBM XIV 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.

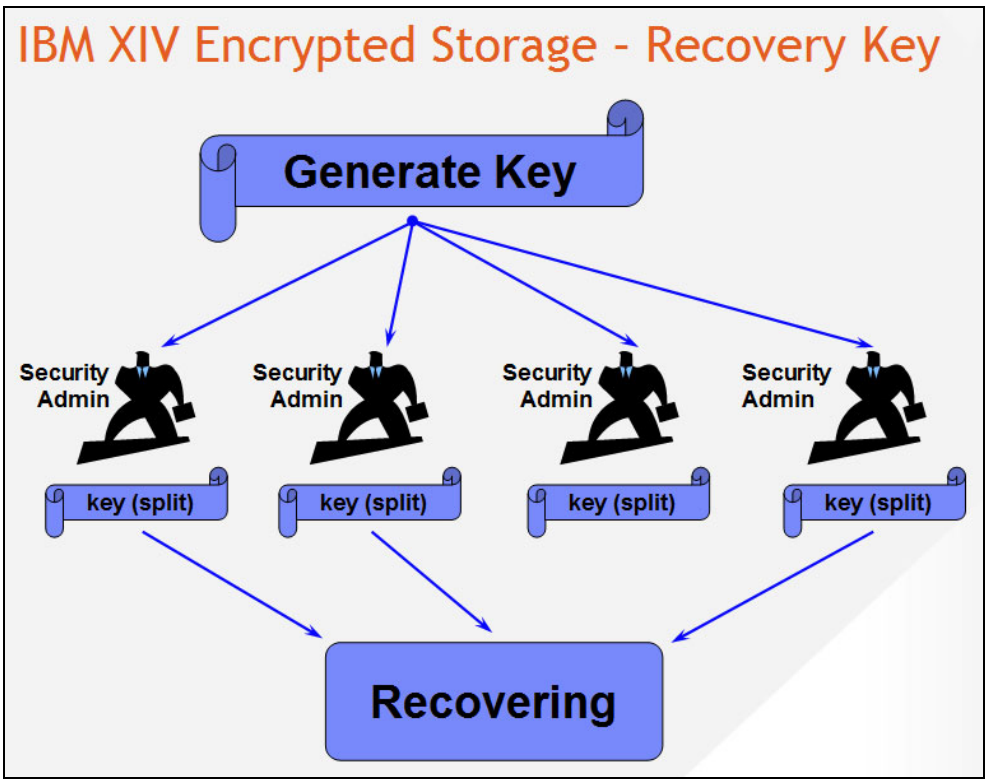

Figure 4-24 Recovery key

4.4.1 Process for recovery keys

The recovery keys can be generated only if data-at-rest encryption is deactivated on the IBM XIV system. Make sure that at least two IBM 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 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

In the IBM XIV GUI, log in as a Security Administrator and select **All Systems** → **List** → **select your system** → **Generate Recovery Key**, as shown in Figure 4-25 on page 36.

Figure 4-25 Generate Recovery Key

You must choose at least two Security Administrators and add them to the Recovery Key Owners section on the right side, and then click **Start** as shown in Figure 4-26.

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 will be 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 will be necessary to unlock the IBM XIV but three of them will be able to do so.

Figure 4-26 Generate Recovery Key pane

Now that the recovery key is generated, you can verify whether the process was successful by clicking **Show Results**, as shown in Figure 4-27 on page 37.

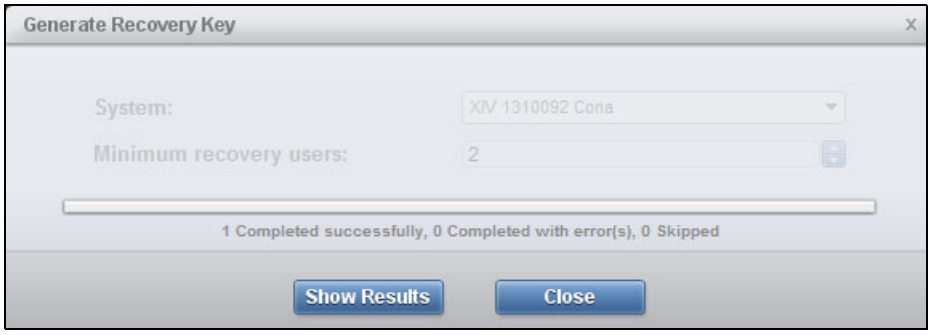

Figure 4-27 Generate recovery keys result

A default text editor opens and shows the `Completed Successfully` log message shown in Figure 4-28.

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

Figure 4-28 Recovery key generation log

When you close this window, the Generate Recovery Key window shown in Figure 4-29 informs you that each of the Security Administrators must log in to acquire the keys generated for them.

Figure 4-29 "Recovery keys generation completed successfully" message

If the IBM 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 4-30 on page 38.

```
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 4-30 Recovery key generation error log

4.4.3 Recovery key validation

Now, the recovery key must be acquired by each Security Administrator. The Security Administrators must log in with their own credentials to copy and paste the key in the Verify Key field and then activate it by clicking **Activate Recovery Key**, as shown in Figure 4-31.

Figure 4-31 Activate Recovery Key

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, physically separate from both the IBM XIV and the Tivoli Key Lifecycle Manager servers.

Click **Continue** to proceed, as shown in Figure 4-32.

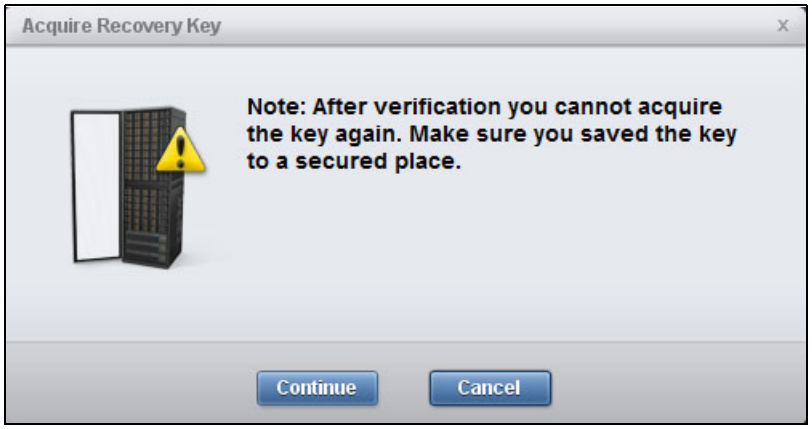

Figure 4-32 Acquire Recovery Key note

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

4.4.4 Recovery key generation with XCLI

If you prefer, you can generate the recovery key through the XIV Storage System command-line interface (XCLI).

Start with the **`encrypt_recovery_key_generate`** command shown in Table 4-1.

Table 4-1 encrypt_recovery_key_generate

Category	Command	Description
System	**`encrypt_recovery_key_generate`**	Specifies which Security Administrators receive recovery key shares and the minimum number of recovery key shares that need to be entered

This example shows the command:

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

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

Table 4-2 encrypt_recovery_key_get

Category	Command	Description
System	**`encrypt_recovery_key_get`**	Retrieve the recovery key share generated for the current user

This example shows the command:

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

All defined Security Administrators must verify their keys, as shown in Table 4-3 on page 40

Table 4-3 encrypt_recovery_key_verify

Category	Command	Description
System	**encrypt_recovery_key_verify**	Confirm that the current user has correctly copied the recovery key share presented by **encrypt_recovery_key_get**

This example shows the command:

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

The state of verification can be checked with the **encrypt_recovery_key_status** command in Table 4-4.

Table 4-4 encrypt_recovery_key_status

Category	Command	Description
System	**encrypt_recovery_key_status**	Shows status of recovery keys

This example shows the command:

```
XIV 1310092 Cona>>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 IBM XIV data-at-rest encryption can be activated. For instructions, see 4.5.1, “Activate data-at-rest XIV encryption” on page 43.

You can use **encrypt_recovery_key_list** to show the number of shares that have recovery keys and how many of them are required for recovery.

4.4.5 Recovery key rekey

Rekeying is the process of changing cryptographic values in the chain between 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 Tivoli Key Lifecycle Manager server is available. A Tivoli Key Lifecycle Manager server is required to enable the XIV system to verify that it is in the correct environment.

Only when the Tivoli Key Lifecycle Manager 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 IBM XIV 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 sends the externally stored key (ESK) to the Tivoli Key Lifecycle Manager and requests a rekey validation.
2. The Tivoli Key Lifecycle Manager verifies the identity of the XIV by using its certificates.
3. The Tivoli Key Lifecycle Manager signals the IBM XIV that it can proceed to generate a new recovery key.
4. The XIV generates a new 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.

In the IBM XIV GUI, log in as a Security Administrator, and select **All Systems** → **List** → **select your system** → **Re-Generate Recovery Key**, as shown in Figure 4-33.

Figure 4-33 Re-Generate Recovery Key

The recovery key can also be rekeyed in the IBM XCLI by using the **`encrypt_recovery_key_rekey`** command shown in Table 4-5.

Table 4-5 encrypt_recovery_key_rekey

Category	Command	Description
system	**`encrypt_recovery_key_rekey`**	Restarts the recovery key generation process described in this command: **`encrypt_recovery_key_generate`**

This example shows the command:

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

4.4.6 Using a recovery key to unlock an XIV

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 IBM XIV cannot get the required data key from the master key server, it attempts to contact all other configured Tivoli Key Lifecycle Manager 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 XIV master key that unlocks the drives. After access to data is restored, the IBM XIV 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 IBM XCLI command to do so is **`encrypt_recovery_key_enter`**, as shown in Table 4-6.

Table 4-6 encrypt_recovery_key_enter

Category	Command	Description
system	**`encrypt_recovery_key_enter`**	Unlocks encrypted disks when the system reboots and cannot access any of the defined key servers, as long as 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, XIV unlocks and activates the data-at-rest encryption again, as shown in Figure 4-34.

Figure 4-34 encrypt_recovery_key_enter

Important: After the minimum required number of keys has been entered, an IBM representative must access the system with *technician* authorithy and change the state of the XIV system from *maintenance* to *on* by issuing a **`state_change target_state=on`** command. A Storage Administrator does not have the authorithy to run that command.

4.5 Activate or deactivate encryption

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

4.5.1 Activate 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 in **`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 can be handled either in the IBM XIV GUI or through the IBM XCLI.

In the IBM XIV GUI, log in as Security Administrator, and select **Systems** → **System Settings** → **Activate Encryption**, as shown in Figure 4-35.

Figure 4-35 Activate data-at-rest IBM 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 IBM XCLI by using the command shown in Table 4-7.

Table 4-7 encrypt_enable

Category	Command	Description
System	**`encrypt_enable`**	Enable 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 Deactivate IBM XIV data-at-rest encryption

This command disables the data protection feature. A prerequisite for this 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 IBM XIV GUI, log in as Security Administrator, and select **Systems** → **System Settings** → **Deactivate Encryption**, as shown in Figure 4-36 on page 44.

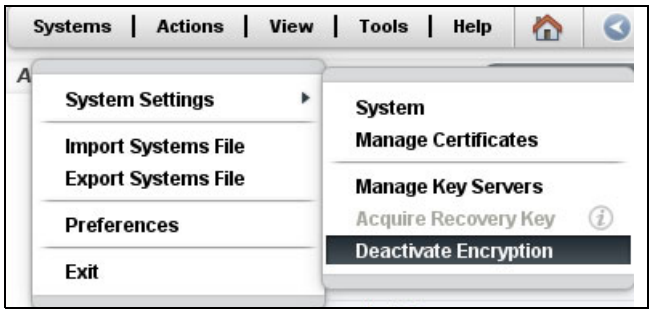

Figure 4-36 Deactivate Encryption

As Figure 4-37 shows, the system prompts you to verify that you want to deactivate encryption on the XIV system.

Figure 4-37 Deactivate Encryption verification

4.6 Verify encryption state

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

- 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 new Encryption State column in the output of the XCLI `state_list` command can show any of these states: *Supported* (encryption is disabled), *Enabling/Activating, Partial, Enabled/Activated, Enabling on Boot, or Disabling*.
- The modified `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 IBM XCLI commands are listed in the reference section of the IBM XIV Information Center:

http://publib.boulder.ibm.com/infocenter/ibmxiv/r2/index.jsp

Other data-at-rest encryption-related IBM XCLI commands include those in this list:

encrypt_enable
Enable encryption

encrypt_disable
Disable encryption

encrypt_keyserver_define
Add a key server

encrypt_keyserver_list
Show defined key servers and related parameters

encrypt_keyserver_update
Change parameters of the key server

encrypt_keyserver_delete
Delete a key server

encrypt_keyserver_rekey
Replace the previous key server key with a new one

encrypt_keyserver_rename
Rename the key server

encrypt_recovery_key_list
Show the recovery key number of shares and how many of them are required for recovery

encrypt_recovery_key_generate
Generate a recovery key

encrypt_recovery_key_verify
Verify and, therefore, activate the recovery key that is collected

encrypt_recovery_key_get
Collect the recovery key that is generated

encrypt_recovery_key_rekey
Replace the recovery key with a new one, which generates a new key that must be verified and copied by **encrypt_recovery_key_get** (works when keys are defined and can be performed by the Storage Administrator also)

The output looks like this example:

```
>> encrypt_recovery_key_rekey
command 0:
administrator:
command:
code = "SUCCESS"
status = "0"
status_str = "Command completed successfully"
aserver = "DELIVERY_SUCCESSFUL"
```

encrypt_recovery_key_enter
Enter a recovery key when the data-at-rest encryption of an IBM XIV is locked

encrypt_recovery_key_status
List all recovery key users and show the status of the recovery key

The output looks like this example:

```
>> encrypt_recovery_key_status
Date Created          User            Status
--------------------- --------------- ----------
2013-10-21 08:59:49   itsosecadmin    Verified
2013-10-21 08:59:49   itsosecadmin2   Verified
```

pki_list
Show the available IBM XIV certificates

pki_show_certificate
Show a specific IBM XIV certificate and its details

This example shows the command:

```
pki_show_certificate name=XIV
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 98 (0x62)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=US, O=ibmXIVDisk
Validity
Not Before: May 28 19:18:42 2013 GMT
Not After : Nov 16 00:22:12 2032 GMT
Subject: C=US, O=ibmXIVDisk, CN=2810-1310092
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:d0:b0:4c:72:cd:b9:04:d1:64:48:fa:68:e2:49:
3b:12:84:18:fa:aa:03:cd:fe:8d:c5:03:96:f6:7d:
....
.....
54:99:55:89:e3:f8:62:3b:36:59:d7:12:39:44:2c:
ab:85:1f:62:2f:f9:bb:aa:95:78:81:97:a3:ca:1e:
10:cf
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
8C:18:40:1F:11:6A:49:61:C6:A9:FF:76:11:D5:69:D7:A0:84:6E:D6
X509v3 Subject Alternative Name:
email:2810-1310092@storage.ibm.com, DNS:storage.ibm.com,
URI:http://www.storage.ibm.com/DS
X509v3 Private Key Usage Period:
Not Before: May 28 19:18:42 2013 GMT, Not After: May 23 19:18:42 2032 GMT
X509v3 Key Usage: critical
Digital Signature
X509v3 Extended Key Usage:
TLS Web Client Authentication, TLS Web Server Authentication
```

```
X509v3 Certificate Policies:
Policy: 1.2.3.4.5.6.7
User Notice:
Explicit Text: "NOTICE: This certificate is only valid for genuine IBM storage
products or
authorized IBM licensees."
X509v3 CRL Distribution Points:
Full Name:
URI:ldap://9.52.212.246:389/o=ibmXIVDisk,c=US
X509v3 Authority Key Identifier:
keyid:E6:6B:70:22:C4:C7:FC:6C:B6:B0:54:BD:CF:A0:29:C0:14:30:86:6B
DirName:/C=US/O=ibmXIVDisk
serial:00
Signature Algorithm: sha256WithRSAEncryption
b3:5a:1a:0b:2b:f6:c0:9f:90:2c:76:34:8e:eb:fa:a3:0d:f6:
8f:e8:20:ca:6a:28:d1:24:c5:32:8c:60:76:8f:63:d5:23:5b:
...
....
7b:24:dd:8e:59:8b:2f:c1:0d:d9:9a:0f:b4:5e:73:d8:3a:6e:
b3:f0:37:83
-----BEGIN CERTIFICATE-----
MIIEyTCCA7GgAwIBAgIBYjANBgkqhkiG9w0BAQsFADAiMQswCQYDVQQGEwJVUzET
.....
.....
LFzJ32A9vsoBZGy3XnZ1GR/fxTToN30dxwDi7Kll0DD6HbW83Q+VhHxC755Bp9Ni
MN6JQIMSj3sk3Y5Ziy/BDdmaD7Rec9g6brPwN4M=
-----END CERTIFICATE-----
```

pki_generate_csr
Generate a certificate signing request

pki_remove
Delete a PKI content

pki_set_pkcs12
Import a PKCS#12 certificate

pki_generate_private_key_and_csr
Generate a private key and certificate signing request (CSR)

pki_rename
Change a PKI symbolic name

pki_set_pem
Import a signed certificate in PEM format

pki_update
Update a PKI certificate or services

5

Maintaining

This chapter explains maintenance tasks related to data-at-rest encryption for the IBM XIV Storage System. It covers these topics:

5.1 Backup and restore procedures

IBM Tivoli Key Lifecycle Manager does not automatically synchronize between servers, but it does provide a convenient backup and restore operation that can be performed using the command line or web user interface. Synchronization involves backing up Tivoli Key Lifecycle Manager and then restoring to a separate server with the same configuration parameters. Be sure to take these factors into consideration:

- Select one server to be the *main* Tivoli Key Lifecycle Manager key server, and originate all backups from there. Make all changes on this main key server and then deploy it through a backup and restore operation to the other Tivoli Key Lifecycle Manager server.
- Both Tivoli Key Lifecycle Manager servers must be running the same OS with the same user accounts for Tivoli Key Lifecycle Manager, Tivoli Integrated Portal, and IBM DB2® database. The OS, directory structure, and DB2 admin user must be exactly the same.
- The restore task is a disruptive operation. Therefore, ensure that the other Tivoli Key Lifecycle Manager key server is active and serving keys before you perform the restore operation.

Backup and restore tasks provide protection for critical data. They require consideration of your site practices to ensure server availability and runtime capabilities. Tivoli Key Lifecycle Manager creates backup files that contain critical data for the current state of the Tivoli Key Lifecycle Manager server. If you have servers with different operating systems, you must use the export/import function.

Important: Failure to back up your keystore and other critical data properly might result in unrecoverable loss of all access to your encrypted data. Do not encrypt your backup file, and do not store a backup file on an encrypting device. Failure to back up data might also result in subsequent inconsistency of the key manager and potential data loss on the storage device.

5.2 Starting and stopping a Tivoli Key Lifecycle Manager server

You might have to use the **`startServer`** or **`stopServer`** scripts to start or stop the Tivoli Key Lifecycle Manager server. Restarting that server, for instance, is required after completion of a restore task. You can also check Tivoli Key Lifecycle Manager status.

Starting and stopping the server by using scripts

Scripts to start and stop the Tivoli Key Lifecycle Manager server are located in the `/TIP_HOME/bin` directory for Linux and IBM AIX platforms. For Microsoft Windows platforms, the directory is `C:\ibm\tivoli\tiptklmV2\bin\`. In the commands, the **`server1`** parameter is the default name of the configured Tivoli Key Lifecycle Manager server instance.

Start the Tivoli Key Lifecycle Manager server

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

- Microsoft Windows systems:

  ```
  StartServer.bat server1
  ```

- Linux and IBM AIX systems:

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

Stop the Tivoli Key Lifecycle Manager server

To stop the server, use the command for your system:

- Windows systems:

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

- Linux and AIX systems:

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

When global security is enabled (which is suggested), enter the user ID and password of the Tivoli Integrated Portal 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 Tivoli Key Lifecycle Manager server is running, try to log in to the Tivoli Integrated Portal. If the login is successful, the Tivoli Key Lifecycle Manager service is running. Otherwise, you can issue the **serverStatus** command (in the `/TIP_HOME/bin` directory for Linux and AIX and in the `C:\ibm\tivoli\tiptklmV2\bin\` directory for Windows) with the server instance, username, and password parameters, as illustrated in Example 5-1.

Example 5-1 Check server status

```
cmd> ./serverStatus server1 -username TipAdminId -password Password
ADMU0116I: Tool information is being logged in file
           /opt/IBM/tivoli/tip/profiles/TIPProfile/logs/server1/serverStatus.log
ADMU0128I: Starting tool with the TIPProfile profile
ADMU0500I: Retrieving server status for server1
ADMU0508I: The Application Server "server1" is STARTED
```

On Windows systems, you can also check in the Services window to verify that the service is running, as shown in Figure 5-1.

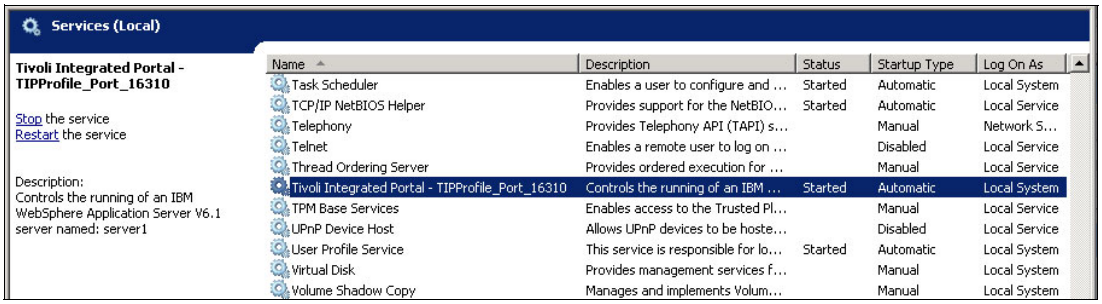

Figure 5-1 Windows server state check

In Linux, issue the **ps** command, as shown in Example 5-2.

Example 5-2 Linux server state check

```
ps -ef | grep tip | grep server1
root      3747     1 67 16:20 pts/7    00:00:55 /opt/IBM/tivoli/tip/java/bin/java
-Declipse.security -Dwas.status.socket=59520
...../opt/IBM/tivoli/tip/profiles/TIPProfile/config TIPCell TIPNode server1
```

5.3 Key exporting and importing tasks

If you have two Tivoli Key Lifecycle Manager servers running on separate operating systems, and if both key server platforms operate in *clear* key mode (export of both private and public keys is allowed), backup and restore functions are not supported. The only way to keep them synchronized is to export the certificate from one server and restore it on the other server by using the export and import functions.

Unsupported: Only the Tivoli Key Lifecycle Manager CLI mode can be used for this process, because the GUI does not support these functions. Newer versions, such as the IBM Security Key Lifecycle Manager, support these functions in the GUI.

5.3.1 Exporting keys

To export keys, complete the following steps:

1. Open a command window, navigate to `<tip installation directory>/bin` folder, and execute the **`wsadmin`** command to export keys from the primary Tivoli Key Lifecycle Manager server (server1) as illustrated in Example 5-3.

   Example 5-3 Issue the wsadmin command

   ```
   23a4088:/opt/IBM/tivoli/tip/bin/wsadmin.sh -username tipadmin -password
   tipadmin -lang jython
   WASX7209I: Connected to process "server1" on node TIPNode using SOAP connector;
   The type of process is: UnManagedProcess
   WASX7029I: For help, enter: "$Help help"
   ```

2. Use 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 Tivoli Key Lifecycle Manager server in the IBM XIV, and the **`-keyStoreName`** is the master keystore name for the Tivoli Key Lifecycle Manager server. See Example 5-4.

   Example 5-4 Exporting the keystore

   ```
   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 location: `/opt/IBM/tivoli/tip/products/tklm`.

3. Copy and archive the exported key to the new Tivoli 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 Tivoli Key Lifecycle Manager CLI information, see the “Command-line interface” topic in the information center:

http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.tklm.doc_2.0.1%2Fref%2Fref_ic_cli.html

5.3.2 Importing keys

Complete the following steps on the second Tivoli Key Lifecycle Manager server after you have exported the keys, as described previously in Step 5.3.1, “Exporting keys” on page 52, and copied them to your key server where you want to perform the import action:

1. Open a command window, go to the <*tip installation directory*>/bin folder, and use the **wsadmin** command to import keys from Tivoli Key Lifecycle Manager server1. See Example 5-5.

   Example 5-5 Issue the wsadmin command

   ```
   23a4089:/opt/IBM/tivoli/tip/bin/wsadmin.sh -username tipadmin -password xxxxxxxx -lang
   jython
   WASX7209I: Connected to process "server1" on node TIPNode using SOAP connector;  The
   type of process is: UnManagedProcess
   WASX7029I: For help, enter: "$Help help"
   ```

2. Use the **tklmKeyImport** command with the **-alias**, **-fileName**, **-password**, **-keyStoreName**, **-usage**, and **-type** parameters to import secret or private keys. The **-alias** parameter is the name from the Tivoli Key Lifecycle Manager server in the IBM XIV, and the **-password** is the key password from the IBM XIV system, the **-keyStoreName** is the master keystore name for the Tivoli Key Lifecycle Manager server, and **-usage** defines the storage type. See Example 5-6.

   Example 5-6 Import the keystore

   ```
   wsadmin>print AdminTask.tklmKeyImport('[-alias 1310092 -fileName
   /root/fromTKLM_server1/TKLM_XIV -password xxxxxxxx -keyStoreName
   "defaultKeyStore_xiv" -usage XIV -type privatekey]')
   CTGKM0001I Command succeeded.
   ```

For more Tivoli Key Lifecycle Manager CLI information, see the “Command-line interface” topic in the information center:

http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.tklm.doc_2.0.1%2Fref%2Fref_ic_cli.html

5.4 Server rekey

The server rekey option is available on IBM XIV Storage System. This option enables a user in the Storage Administrator or Security Administrator role to rekey against the master key server.

As a good security practice, use this function to periodically change the keys.

5.4.1 Server rekey by using the IBM XIV GUI

The following procedure describes how to rekey the server:

1. In the IBM XIV GUI, navigate to **All systems** → **List**, and then select your IBM XIV, right-click it, and choose the **Server ReKey** menu entry, as shown in Figure 5-2 on page 54.

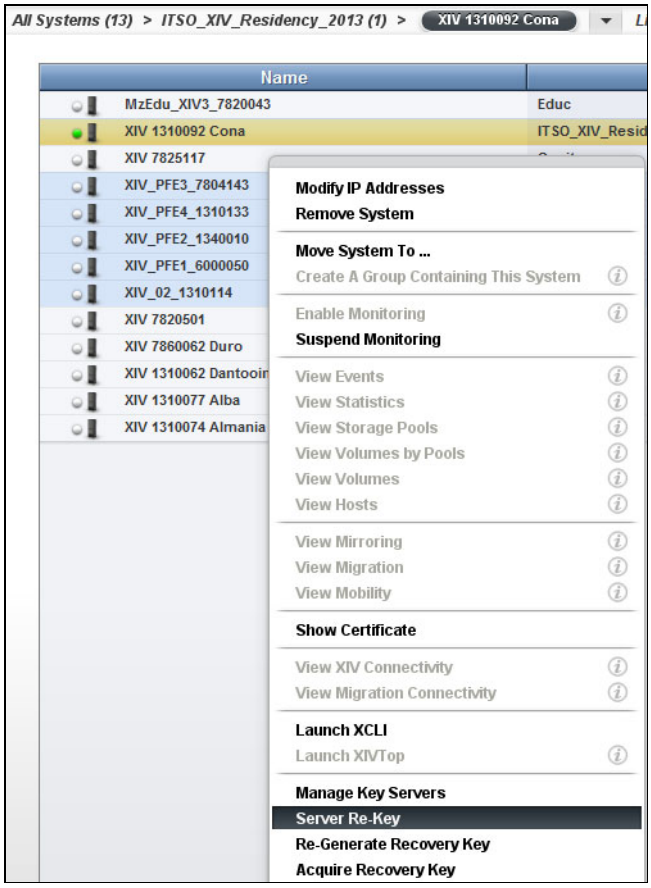

Figure 5-2 Server Re-Key

2. When the Server Re-key window shown in Figure 5-3 is displayed, you can select the XIV system from the drop-down menu, and then click **Start**.

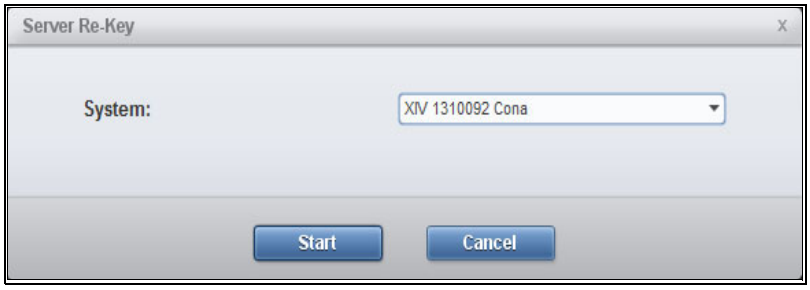

Figure 5-3 Server Re-Key Start

Obviously, the key server must be available to process the rekey request, and the XIV will display an error message, as Example 5-3 on page 52 shows, if it is not.

Figure 5-4 Server rekey error message

If the key server is available, it will create a new key. The `Completed Successfully` confirmation message looks like the example in Figure 5-5.

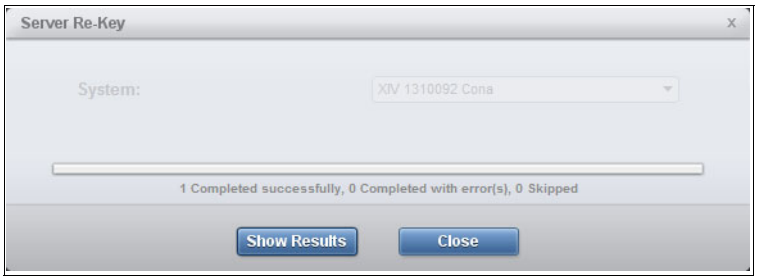

Figure 5-5 Server Rekey results

3. You can click **Show Results** to see the successfully completed message, as shown in Figure 5-6.

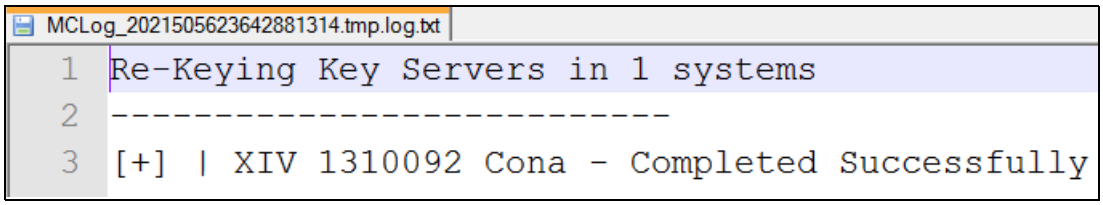

Figure 5-6 Rekey completed successfully

5.4.2 Server rekey using the XCLI

The server key can also be rekeyed in the IBM XIV Storage System command-line interface (XCLI) by using the **`encrypt_keyserver_rekey`** command, as shown in Table 5-1.

Table 5-1 encrypt_keyserver_rekey

Category	Name	Description
system	**`encrypt_keyserver_rekey`**	Initiates a rekey against the master key server

This example shows the command:

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

5.5 Encryption deadlock

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

The keystore data is accessed by the key server application through a password 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 is not able to 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 is necessary to allow the key server to access its keystore and to allow the key server to communicate with its storage device clients. Similarly, any backups of the keystore must not be stored on a storage device that has a dependency on a key server to access data.

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

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

Temporary encryption deadlock
The temporary encryption deadlock indicates a situation where the IBM XIV cannot access its disk devices because Tivoli 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.

Permanent encryption deadlock
This permanent encryption deadlock is the worst case. Here, all Tivoli 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 managed by the set of Tivoli Key Lifecycle Managers is, in effect, cryptographically erased. For all practical purposes, that data is permanently lost.

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 nonencrypted 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 required by the key server from nonencrypted to encrypted storage.
- Generally, several layers of virtualization in the I/O stack hierarchy can cause difficulties for the client to maintain an 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 obtained from other subordinate storage arrays. The data 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 Tivoli Key Lifecycle Manager server is mandatory, and the client must be directly involved in managing the encryption environment. See Chapter 3, “Planning” on page 13 and Chapter 4, “Configuring and implementing XIV encryption” on page 19.

5.6 Disk and module replacement

If a disk drive that does not support encryption is added to an encryption-enabled XIV, it will fail the component test and cannot be included in the running XIV configuration. This ensures that no unencrypted data resides on any disk drives inside the XIV system.

Related publications

The publications listed in this section are particularly suitable for a more detailed information about the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this paper. Some publications cited in this list might be available in softcopy only.

- *IBM XIV Storage System: Copy Services and Migration*, SG24-7759

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, drafts, and additional materials on the Redbooks website:

ibm.com/redbooks

Other publications

These publications are also relevant for additional information:

- *IBM XIV Storage System Planning Guide*, GC27-3913
- *IBM XIV Storage System: Product Overview*, GC27-3912
- *IBM XIV Storage System User Manual*, GC27-3914
- *IIBM XIV Storage System XCLI Utility User Manual*, GC27-3915

Online resources

These websites are also relevant for further information:

- IBM XIV Storage System Information Center:

  http://publib.boulder.ibm.com/infocenter/ibmxiv/r2/index.jsp

- IBM XIV Storage System website:

  http://www.ibm.com/systems/storage/disk/xiv/index.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM XIV Security with Data-at-Rest Encryption

Data on disk encrypted with no performance impact

Hot encryption support with no downtime

External key manager

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. As with all of its features, the IBM XIV Storage System provides data-at-rest encryption at no charge. This gives clients the opportunity to take advantage of encryption and still enjoy the lower total cost of ownership that XIV offers.

This IBM Redpaper publication explains the architecture and design of the XIV encryption solution and how it must be configured and implemented. It can help clients and Storage Administrators who want to enable data encryption on XIV storage systems.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-5047-00