

Key Management Deployment Guide

Using the IBM Enterprise Key Management Foundation

Enterprise integration for centralized
key management deployment

Complete information about
architecture and components

Deployment scenario
with hands-on details



Mike Andreasen
Carsten Dahl Frehr
W. Craig Johnston
Alina Mot
Troels Norgaard
Soren Peen
Per Snowman
Axel Buecker

ibm.com/redbooks

Redbooks



International Technical Support Organization

**Key Management Deployment Guide: Using the IBM
Enterprise Key Management Foundation**

October 2014

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

First Edition (October 2014)

This edition applies to Version 8, Release 2, Modification 1 of IBM Enterprise Key Management Foundation.

© **Copyright International Business Machines Corporation 2014. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

| | |
|--|------|
| Notices | vii |
| Trademarks | viii |
| Preface | ix |
| Authors | x |
| Now you can become a published author, too! | xiii |
| Comments welcome | xiii |
| Stay connected to IBM Redbooks | xiii |
| Part 1. Business context and solution architecture | 1 |
| Chapter 1. Business context for enterprise key management | 3 |
| 1.1 The need for encryption | 4 |
| 1.1.1 Reasons for encryption | 4 |
| 1.2 The need for enterprise key management | 5 |
| 1.3 IBM Security Framework and Blueprint | 6 |
| 1.3.1 IBM Security Framework | 7 |
| 1.3.2 IBM Security Blueprint | 8 |
| 1.4 Enterprise key management and the IBM Security Blueprint | 11 |
| 1.5 Conclusion | 12 |
| Chapter 2. Solution architecture | 13 |
| 2.1 Functional overview | 14 |
| 2.1.1 IBM Enterprise Key Management Foundation highlights | 14 |
| 2.1.2 Benefits of IBM Enterprise Key Management Foundation | 15 |
| 2.1.3 IBM Enterprise Key Management Foundation functions | 17 |
| 2.2 Logical and physical components | 21 |
| 2.2.1 Component overview | 22 |
| 2.2.2 Key Repository | 24 |
| 2.2.3 Key Management Workstation | 26 |
| 2.2.4 Browser | 29 |
| 2.2.5 Agent | 29 |
| 2.2.6 Reporter | 30 |
| 2.3 Sysplex technology | 30 |
| 2.3.1 System z logical partitioning | 31 |
| 2.3.2 Parallel Sysplex usage | 32 |
| 2.3.3 Network architecture | 32 |
| 2.4 Disaster recovery | 33 |
| 2.4.1 Key Management Workstation | 33 |

| | |
|--|-----------|
| 2.5 Smart card support | 36 |
| 2.5.1 Zone concepts | 38 |
| 2.5.2 CA smart card | 40 |
| 2.5.3 Enrolling an entity | 41 |
| 2.5.4 TKE smart cards | 41 |
| 2.5.5 Reuse of TKE smart cards between EKMF and TKE workstations | 42 |
| 2.6 Roles and responsibilities | 43 |
| 2.6.1 Basic concepts | 44 |
| 2.6.2 Access control systems | 46 |
| 2.6.3 Role concept | 55 |
| 2.7 Migration considerations | 63 |
| 2.8 Conclusion | 65 |
| Chapter 3. Deployment, administration, and maintenance | 67 |
| 3.1 Understanding deployment options | 68 |
| 3.1.1 Configurations | 68 |
| 3.1.2 Environments | 73 |
| 3.1.3 Online Key Repository access | 75 |
| 3.1.4 Online keystore access | 76 |
| 3.1.5 Designing the security organization | 78 |
| 3.2 Maintenance of the installation | 78 |
| 3.2.1 Maintenance of the workstation | 78 |
| 3.2.2 Maintenance of Agents and data tables | 80 |
| 3.3 Administering users | 81 |
| 3.4 Providing applicable logging | 81 |
| 3.5 Tracing for troubleshooting | 83 |
| 3.5.1 Other tools for troubleshooting | 84 |
| 3.6 Ensuring consistent backup and restore procedures | 86 |
| 3.7 Conclusion | 87 |
| Part 2. Use case scenario | 89 |
| Chapter 4. Overview of scenario, requirements, and approach | 91 |
| 4.1 Company overview | 92 |
| 4.1.1 Current IT infrastructure | 92 |
| 4.1.2 Key management issues in the current infrastructure | 93 |
| 4.2 Business requirements | 94 |
| 4.2.1 Compliance | 94 |
| 4.2.2 Cost-effective key management operations | 94 |
| 4.3 Functional requirements | 95 |
| 4.3.1 Centralized operations | 95 |
| 4.3.2 Basic key management requirements | 95 |
| 4.3.3 Extended key management requirements | 96 |
| 4.4 Architectural decisions | 97 |

| | | |
|---|---|------------|
| 4.5 | Solution overview | 99 |
| 4.5.1 | The design for the IT infrastructure and processes. | 99 |
| 4.5.2 | The plan for implementation phases. | 106 |
| 4.6 | Conclusion. | 107 |
| Chapter 5. Key management infrastructure setup and deployment. | | 109 |
| 5.1 | Planning for deployment | 110 |
| 5.1.1 | System z | 110 |
| 5.1.2 | Key Management Workstation | 118 |
| 5.1.3 | Keys to be managed | 121 |
| 5.1.4 | Keys to be managed for the application | 124 |
| 5.1.5 | Key label naming convention | 124 |
| 5.2 | Implementation | 126 |
| 5.2.1 | System z | 126 |
| 5.2.2 | Key Management Workstation | 135 |
| 5.3 | Managing keys | 185 |
| 5.3.1 | Adding key zones | 186 |
| 5.3.2 | Adding system application names. | 189 |
| 5.3.3 | Setting up the device configuration. | 191 |
| 5.3.4 | Importing key templates | 198 |
| 5.3.5 | Verifying the key templates | 200 |
| 5.3.6 | Generating keys | 204 |
| 5.3.7 | Leaving insecure mode | 212 |
| 5.4 | Link encryption configuration. | 216 |
| 5.4.1 | Configuring the Agents | 216 |
| 5.4.2 | Configuring RACF permissions. | 217 |
| 5.4.3 | Configuring the application | 217 |
| 5.5 | Application keys | 220 |
| 5.5.1 | Requesting key generation | 221 |
| 5.5.2 | Processing a key generation request | 224 |
| 5.6 | Key lifecycle management | 227 |
| 5.7 | Conclusion. | 232 |
| Appendix A. Troubleshooting | | 235 |
| EKMF workstation | | 236 |
| Tracing problems | | 241 |
| EKMF agents | | 242 |
| Agents on z/OS | | 242 |
| EKMF Agents on other platforms | | 243 |
| CCA Node Management Utility | | 244 |
| Appendix B. Operational procedures | | 247 |
| Smart card management using Smart Card Utility Program | | 248 |
| Initializing and personalizing a certificate authority smart card. | | 248 |

| | |
|--|------------|
| Backup CA smart card | 256 |
| Enrolling the IBM PCIe 4765 Cryptographic Coprocessor | 262 |
| Initializing and enrolling a TKE smart card | 267 |
| Personalizing a TKE smart card | 272 |
| Unlocking a TKE smart card | 278 |
| Changing the PIN of a CA smart card | 284 |
| Changing the PIN of a TKE smart card | 287 |
| SCUP logon by using split passphrase | 290 |
| SCUP group logon by using smart cards | 292 |
| IBM PCIe 4765 Cryptographic Coprocessor management using CNM | 297 |
| IBM 4765 initialization | 298 |
| Generating an IBM 4765 logon key on TKE smart card | 306 |
| Backing up a TKE smart card | 312 |
| Generating an IBM 4765 DES/PKA master key | 317 |
| Loading an IBM 4765 DES/PKA master key | 326 |
| Setting the IBM 4765 DES/PKA master keys and re-enciphering the key storage | 334 |
| Performing a CNM Utility logon by using a split passphrase | 339 |
| Performing a CNM Utility logon by using a smart card | 342 |
| Performing a CNM Utility group logon by using smart cards | 346 |
| Using the CNM Utility to create, edit, or delete a role | 350 |
| Using the CNM Utility to create a smart card profile | 352 |
| Using the CNM Utility to create a smart card group profile | 354 |
| Using the CNM Utility to create a group of groups profile | 356 |
| Using the CNM Utility to restrict the DEFAULT role | 357 |
| Managing the application | 359 |
| Application group logon | 359 |
| Related publications | 369 |
| IBM Redbooks | 369 |
| Product publications | 369 |
| Online resources | 370 |
| Help from IBM | 370 |

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. 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 IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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® | RACF® | System z® |
| DB2 Connect™ | Redbooks® | z/OS® |
| DB2® | Redbooks (logo)  ® | zEnterprise® |
| IBM® | System i® | |
| Parallel Sysplex® | System p® | |

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.

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

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

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

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

Preface

In an increasingly interconnected world, data breaches grab headlines. The security of sensitive information is vital, and new requirements and regulatory bodies such as the Payment Card Industry Data Security Standard (PCI-DSS), Health Insurance Portability and Accountability Act (HIPAA), and Sarbanes-Oxley (SOX) create challenges for enterprises that use encryption to protect their information. As encryption becomes more widely adopted, organizations also must contend with an ever-growing set of encryption keys. Effective management of these keys is essential to ensure both the availability and security of the encrypted information. Centralized management of keys and certificates is necessary to perform the complex tasks that are related to key and certificate generation, renewal, and backup and recovery.

The IBM® Enterprise Key Management Foundation (EKMF) is a flexible and highly secure key management system for the enterprise. It provides centralized key management on IBM zEnterprise® and distributed platforms for streamlined, efficient, and secure key and certificate management operations.

This IBM Redbooks® publication introduces key concepts around a centralized key management infrastructure and depicts the proper planning, implementation, and management of such a system using the IBM Enterprise Key Management Foundation solution.

A little history: The IBM Enterprise Key Management Foundation is a solution with a long standing history, although the name might appear to be new. It is based on the IBM Distributed Key Management System (DKMS), which has been the IBM strategic solution for centralized cryptographic key management since the 1990's. Since then, it has been adopted primarily by banks and other players in the payment card industry, including several major players acting worldwide, and a number of midsize companies.

In February 2013, IBM DKMS was announced as a System z® Security Solution under the name IBM Enterprise Key Management Foundation (EKMF).

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.



Mike Andreassen is an advisory IT specialist at IBM Copenhagen, Denmark. He has 15 years of experience in IT application development, security, and cryptography. Mike holds a degree in Computer Science from the Technical University of Denmark. His areas of expertise include Java development and key management. He has been a part of the EKMF development team since 2004.



Carsten Dahl Frehr is a Certified Senior IT Specialist in Copenhagen, Denmark. He has 22 years of experience in IT application development, security, and cryptography. Carsten holds a Ph.D. degree in error correcting codes from The Technical University of Denmark. His areas of expertise include cryptography and key management, in particular in relation to the payment card industry. He holds four patents in different areas of key management.



W. Craig Johnston joined IBM in 1982, and is working in the IBM System z Lab Services team specializing in mainframe security. Craig is focused on working with clients to implement enterprise-wide encryption and the System z security components. His previous positions include devising and directing the testing strategies for IBM RACF® and other security services that are provided by components and products for IBM z/OS® and other platforms. Craig has been a member of RACF Development as both a function verification and system tester. As part of the IBM Academic Initiative, Craig worked on several IBM Redbooks residencies and teaches mainframe security.



Alina Mot is an IT Security consultant and head of the Competence Center IT-Security at Empalis Consulting GmbH in Germany. Alina has 14 years of experience in cryptography on System z and distributed systems, and in the key management field. She holds a degree in Informatics from the “Vest University of Timisoara” in Romania. Her areas of expertise include architectural design and implementation of cryptographically based solutions and centralized key management with EKMF.



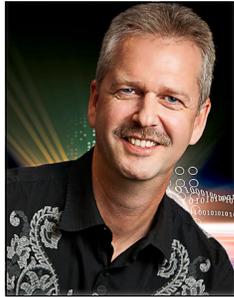
Troels Norgaard is an IT Specialist at the IBM Crypto Competence Center, Copenhagen. He has 16 years of experience working with security, encryption, and key management in the payment industry, supporting large customers across Europe and the US. He played a key role in the development of the IBM Enterprise Key Management Foundation solution and IBM Trusted Key Entry. Before joining the Crypto Competence Center, he worked as project manager on internal IBM projects. He holds Master of Science degree in Engineering.



Soren Peen is an IT security professional with six years of experience working as an application developer and team leader at the IBM Crypto Competence Center that is based in Copenhagen, Denmark. He has worked extensively with IBM solutions in the field of cryptography and enterprise key management and is an expert in the IBM Enterprise Key Management Foundation solution. He has worked for IBM for 10 years and holds a Master of Science degree in Engineering.



Per Snowman is a Senior IT Specialist at the IBM Crypto Competence Center that is based in Copenhagen, Denmark. He has 32 years of experience in IT application development, security, and cryptography. Per holds a Bachelor's degree in Engineering (Electronic Engineer, 1982) from Copenhagen University College of Engineering. His areas of expertise include development of the Danish debit card "Dankort" online payment system that is based on EFTPOS technology in the 1980s, and then he was involved in various payment, security and key management solutions, including guidance about safe configuration and organization, and assessment and compliance audits. He has been part of the IBM Enterprise Key Management Foundation development team since 1991.



Axel Buecker is a Certified Consulting Software IT Specialist at the ITSO in Austin, Texas. He writes extensively and teaches IBM classes worldwide about software security architecture and network computing technologies. He has 27 years of experience in various areas that are related to workstation and systems management, network computing, and e-business solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture. He has a degree in Computer Science from the University of Bremen in Germany.

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

Wade Wallace

International Technical Support Organization, Austin Center

Erik Pauner and Michael Anderson, IBM Crypto Competence Center Copenhagen, for their assistance with test environments and support during the IBM Redbooks residency.

Daniel Burke at Jack Henry Associates for his invaluable contribution of his user experiences with EKMF to this publication.

John Dayka and Maura Schoonmaker (IBM STG) for helping establish the IBM Redbooks residency.

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book 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>



Part 1

Business context and solution architecture

Part 1 highlights the overall business context for enterprise key management systems. It also introduces the high-level components and concepts for the design of an enterprise key management system using the IBM Enterprise Key Management Foundation. In addition, this part provides an understanding of the high-level solution architecture of the IBM Enterprise Key Management Foundation.



Business context for enterprise key management

In an increasingly interconnected world, security is a major concern. The security of sensitive information is vital, and the typical way security is achieved is through encryption. As encryption becomes more widely adopted, organizations also must contend with an ever-growing set of encryption keys. Effective management of these keys is essential to ensure both the availability and security of the encrypted information. Centralized management of keys and certificates is necessary to perform the complex tasks that are related to key and certificate generation, renewal, and backup and recovery.

Furthermore, new requirements and regulatory bodies, such as the Payment Card Industry Data Security Standard (PCI-DSS), Health Insurance Portability and Accountability Act (HIPAA), and European Union in the Data Protection Directive 95/46/EC, create challenges for enterprises that use encryption to protect their information.

This chapter explains the background and describes the business drivers for enterprise key management. This chapter contains the following sections.

- ▶ The need for encryption
- ▶ The need for enterprise key management
- ▶ IBM Security Framework and Blueprint
- ▶ Enterprise key management and the IBM Security Blueprint

1.1 The need for encryption

The security of sensitive data, such as personal information or payments data, is mandated by the payment cards industry by the PCI-DSS and PCI-PIN standards, by the European Union in the Data Protection Directive 95/46/EC, in the HiPAA, and in local legislation. Encryption is one major mechanism to help you stay compliant with these regulations. Companies in other industries might not face legislative requirements, but otherwise have good reasons to protect their assets by using encryption mechanisms.

1.1.1 Reasons for encryption

The need for encryption is dictated by the need to protect IT assets, such as sensitive data, or to secure transactions. For certain applications, such as payments and healthcare, the requirements stem directly from the standard bodies and legislation.

The *confidentiality, integrity, and availability (CIA)* model is an established way of describing IT security. Confidentiality, integrity, and availability are three basic principles in IT security (Figure 1-1). Although the CIA model covers every aspect of information security, it has important messages regarding cryptography. The end of this chapter uses the *IBM Security Framework* to put cryptography into a larger information security context.

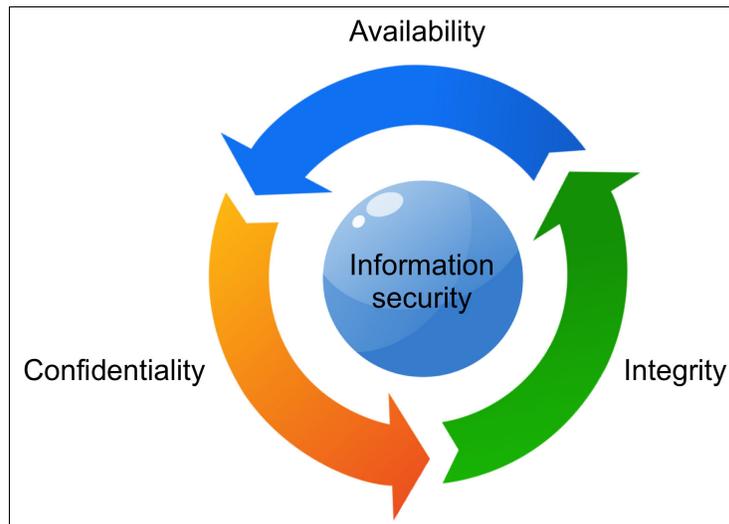


Figure 1-1 CIA security model

Where:

| | |
|------------------------|--|
| Confidentiality | Confidentiality is the safeguarding of data, such that it remains known to only those entities that have a legitimate need to access it. Encryption provides confidentiality and limits access to data to only someone that possess the encryption key. |
| Integrity | Integrity usually refers to the integrity of data, that is, the accuracy and completeness of data. Examples of encryption mechanisms that provide integrity are message authentication codes (MAC) and digital signatures. |
| Availability | The availability of systems and data is critical to businesses. With regard to the usage of encryption mechanisms, the availability of cryptography and keys is now critical. Timely provisioning of keys, backup and restore of keys to keystores, and access to these keys and keystores is required to ensure availability. |

1.2 The need for enterprise key management

Given the pervasiveness of encryption, the number of encryption keys are continuously growing. This in itself calls for effective key management at the enterprise level. In addition, regulatory requirements dictate that key management processes must be in place.

Establishing effective key management operations across an enterprise is a challenge. You must ensure that keys are managed properly over their whole lifecycle, and you must ensure that different requirements for different applications do not lead to vulnerabilities. Here are all the drivers for centralizing key management in the enterprise:

| | |
|-------------------|--|
| Compliance | Compliance is a main concern for businesses today. When you centralize the key management effort, you can streamline your key management processes, and you can assign people full time to key management. Having streamlined processes makes it easier to stay compliant with fewer trained people. |
|-------------------|--|

| | |
|-----------------------------|--|
| Effective operations | If you are in a larger organization with thousands of keys, you value that key management can be done effectively. Establishing a central key management group that owns all key management processes enables you to use the same set of basic procedures for all key management tasks, it gives you a combined overview of all your keys, and you can establish uniform reporting and handling of keys. |
| Auditability | Many customers complain that audits are taking more of their time. The underlying problem is that they are audited by more external and internal entities than before. When you centralize key management and streamline your processes, you eventually reduce the time that you spend for each audit. |

1.3 IBM Security Framework and Blueprint

Today, any business initiative inside an organization is guided by the principles of *Governance, Risk, and Compliance*, which are often seen as broad terms that typically have different meanings to different stakeholders across an organization. Each CxO is trying to manage risk for their domain, so they have different priorities and points of view when it comes to handling these risks:

| | |
|-------------|---|
| CRO | The <i>Chief Risk Officer</i> (CRO) looks at the organization's overall risk profile and where it is most vulnerable to unexpected loss. |
| CFO | The <i>Chief Financial Officer</i> (CFO) must ensure that the necessary controls are in place to have accurate financial statements. |
| CISO | The <i>Chief Information Security Officer</i> (CISO) must ensure that the IT Infrastructure supports the overall business drivers of the organization. The CISO must minimize the risk of the IT environment and assess and communicate the impact of this environment on the overall organization from a Governance, Risk, and Compliance perspective. |

Regardless of the organizational perspective of risk management, both process and IT controls must be established to get a complete picture of the organization's *risk posture*. Establishing and monitoring IT security controls, mitigating the risk observed through these controls, and reporting and communicating risk posture are critical capabilities for an IT security organization.

1.3.1 IBM Security Framework

IBM created the IBM Security Framework to help ensure that every necessary IT security aspect can be properly addressed when you use a holistic approach to business-driven security.

The IBM Security Framework is shown in Figure 1-2.

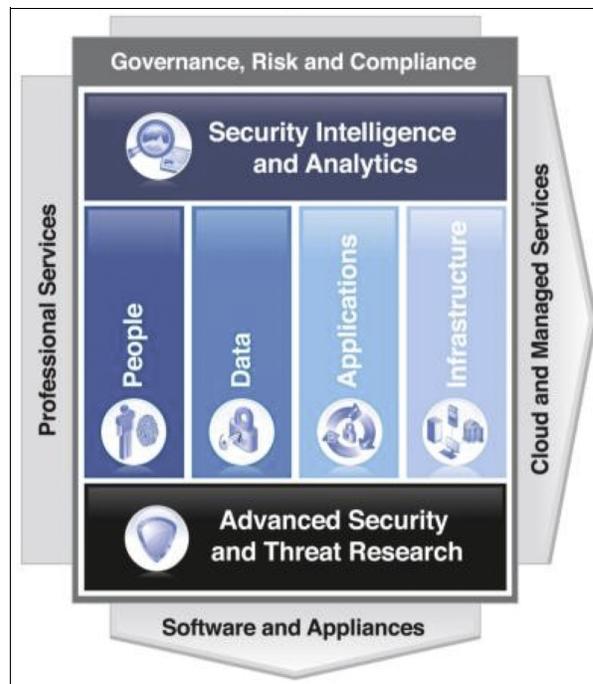


Figure 1-2 The IBM Security Framework

The capabilities that are described by the IBM Security Framework are based on a foundation of the Advanced Security and Threat Research infrastructure. The solutions that are provided within the security domains and additional layers can be delivered through software, hardware (including appliances), and as services, whether managed, professional, or cloud-based.

IBM provides the full breadth and depth of solutions and services that can enable organizations to take this business-driven, secure by design approach to security in alignment with the IBM Security Framework.

1.3.2 IBM Security Blueprint

The IBM Security Framework divides the area of business-oriented IT security into four major security domains and three support layers. The next step is to break down these domains and layers into further detail to work toward a common set of core *security capabilities* that are needed to help an organization securely achieve its business goals. These core security capabilities are called the *IBM Security Blueprint*.

The IBM Security Blueprint uses a product- and solution-neutral approach to categorize and define security capabilities and services that are required to answer the business concerns in the IBM Security Framework.

The IBM Security Blueprint was created after research into many customer-related scenarios that were focused on how to build IT solutions. The intention of the blueprint is to support and help design and deploy security solutions in your organization.

Building a specific solution requires a specific architecture, design, and implementation. The IBM Security Blueprint can help you evaluate these items, but does not replace them. Using the IBM Security Blueprint in this way can provide a solid approach to considering the security capabilities in a particular architecture or solution.

IBM chose to use a high-level service-oriented perspective for the blueprint that is based on the IBM service-oriented architecture (SOA) approach. Services use and refine other services (for example, policy and access control components affect almost every other infrastructure component).

To better position and understand the IBM Security Blueprint, see Figure 1-3 on page 9.

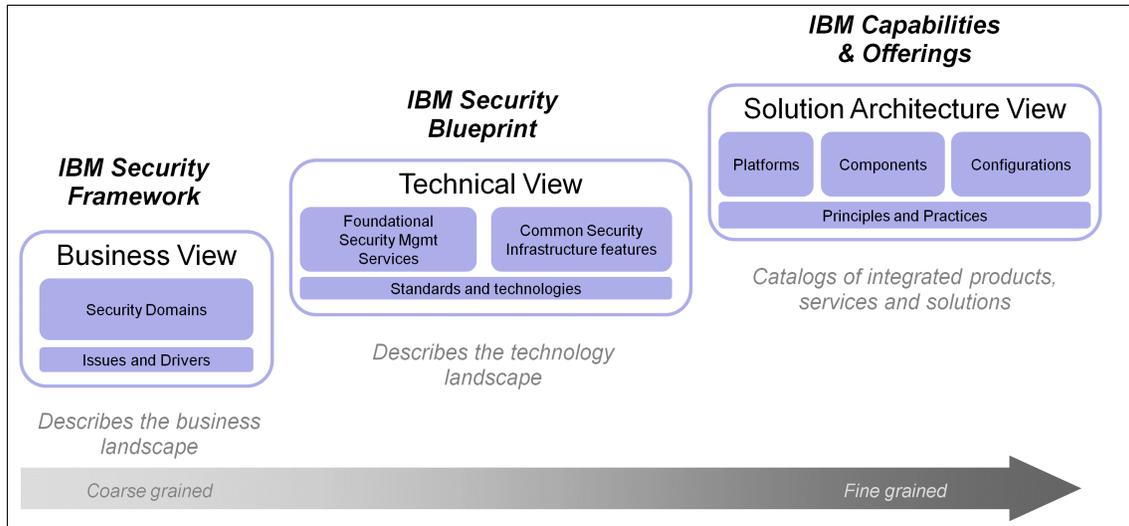


Figure 1-3 IBM Security Blueprint positioning

The left portion of Figure 1-3 represents the IBM Security Framework, which describes and defines the security domains from a business perspective.

The middle portion in Figure 1-3 represents the IBM Security Blueprint, which describes the IT security management and IT security infrastructure capabilities that are needed in an organization. As described earlier, the IBM Security Blueprint describes these capabilities in product and vendor-neutral terms.

The right portion of Figure 1-3 represents the solution architecture views, which describe specific deployment guidance particular to an IT environment and the current maturity of the organization within the respective security domains. Solution architecture views provide details about specific products, solutions, and their interactions.

Figure 1-4¹ shows the complete IBM Security Blueprint, and each layer and component is described in the following sections.

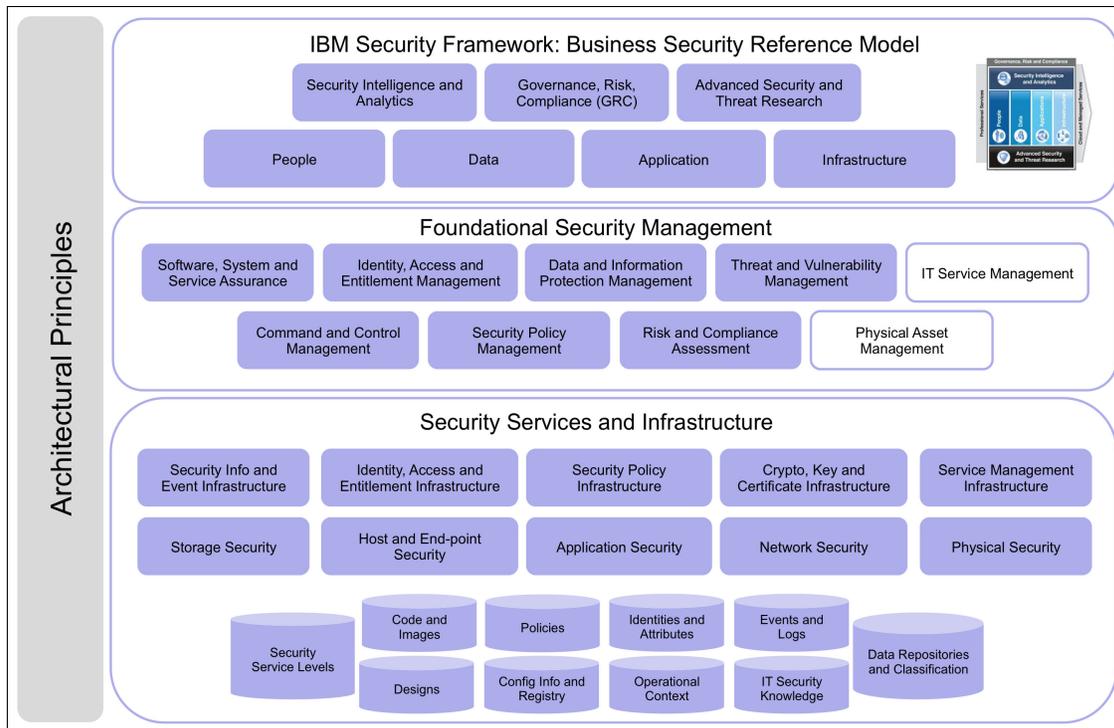


Figure 1-4 The IBM Security Blueprint

Foundational Security Management

The Foundational Security Management layer contains the top-level components that are used to direct and control IT security from a policy-based, risk management perspective. These components are described in more detail in Chapter 2, “The components of the IBM Security Blueprint”, of *Using the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, SG24-8100.

¹ White boxes in Figure 1-4 and other diagrams represent services or components that are not solely security-related, but might be connected with other IT service areas.

1.4 Enterprise key management and the IBM Security Blueprint

This publication focuses on the Cryptography, Key, and Certificate Infrastructure subcomponents, which belong to the Security Services and Infrastructure layers. The Cryptography, Key, and Certificate Infrastructure subcomponents are used by many other IBM Security Blueprint components and capabilities:

- ▶ Risk and Compliance Assessment

This component handles the compliance reporting to the bodies that are relevant to your business. This is a perpetual process for your key management operations.

- ▶ Command and Control Management

Security policies, including those for encryption and key management, are defined as part of the strategic role of Command and Control Management. Further key management has physical security requirements, such as secured rooms with dual access control, which is provided by Command and Control Management.

- ▶ Data and Information Protection Management

In the context of encryption and key management, this component deals with encryption of stored data, in particular tape encryption and encrypting disk drives. Centralized key management helps ensure the availability of keys for this purpose.

- ▶ Software, System, and Service Assurance

Individual keys are provided per application and per usage type by IBM Enterprise Key Management Foundation. To assist efficient development, a series of application programming interfaces (APIs) are available.

- ▶ Identity, Access, and Entitlement Management

For financial institutions implementing this component, it entails PIN-based cards, in particular chip-based cards following the standards from EMVCo.² Other smart cards or RFID-based cards using cryptographic mechanisms fall into this category as well.

² EMV is a global standard for credit and debit payment cards that are based on chip card technology. For more information, go to <http://emvco.com>.

1.5 Conclusion

This chapter introduced the need for encryption as a mechanism to provide information security, how it leads to the need for a centralized key management system, and how centralized key management fits into the IBM Security Blueprint.



Solution architecture

This chapter focuses on the following architectural aspects in regard to the IBM Enterprise Key Management Foundation:

- ▶ Functional overview
- ▶ Logical and physical components
- ▶ Sysplex technology
- ▶ Disaster recovery
- ▶ Roles and responsibilities
- ▶ Migration considerations

2.1 Functional overview

This chapter presents a functional overview of the IBM Enterprise Key Management Foundation.

2.1.1 IBM Enterprise Key Management Foundation highlights

IBM Enterprise Key Management Foundation powered by the Distributed Key Management System (DKMS) provides a centralized management system for cryptographic keys and certificates on any number of servers, including mainframes and distributed servers. The emphasis is on high security, high availability, and recovery of all key material, even in disaster situations.

With IBM Enterprise Key Management Foundation, you can manage DES, TDES, AES, RSA, and ECC keys that are designated for IBM cryptographic hardware complying with the IBM Common Cryptographic Architecture (CCA) and keys for non-IBM cryptographic hardware using plain cryptograms or various key variant-techniques.

Additional resources: You can find more details about the different crypto standards online.

For DES, go to the following website:

<http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard>

For TDES, also called Triple DES, go to the following website:

<http://www.techopedia.com/definition/4144/triple-des>

For AES, go to the following website:

<http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard>

For RSA, go to the following website:

<http://searchsecurity.techtarget.com/definition/RSA>

For ECC, go to the following website:

<http://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography>

Furthermore, IBM Enterprise Key Management Foundation manages certificates and associated keys for both IBM RACF and distributed servers, for example, SSL terminating devices.

An easy to operate and customizable graphical user interface (GUI) makes it possible to support an organization's standards and procedures.

On top of the essential key and certificate management functionality, IBM Enterprise Key Management Foundation offers comprehensive EMV support, remote key loading for ATMs, and PIN printing capabilities.

IBM Enterprise Key Management Foundation is an IBM solution that is developed by the IBM Crypto Competence Center in Copenhagen, Denmark. It is developed and extended in close cooperation with many banks and financial institutions.

2.1.2 Benefits of IBM Enterprise Key Management Foundation

Centralized key management implies that procedures can be simplified and that the key management process is more cost-efficient. IBM Enterprise Key Management Foundation also helps you to comply with requirements from the brands such as Visa and MasterCard, and standards such as PCI-DSS, PCI-HSM, PIN regulations, ANSI 9.24, and ISO 11568.

Furthermore, IBM Enterprise Key Management Foundation delivers a number of benefits that are related to security, productivity, flexibility, and operability.

High level of security

IBM Enterprise Key Management Foundation allows you to manage securely the complete lifecycle of keys using IBM cryptographic hardware for the generation, protection, distribution, and use of these keys within different cryptographic devices and services.

Centralized management for all keys and certificates

The importance of certificates and the keys that are related to them is growing each year, as many applications are using certificates for authentication, single sign-on (SSO), and other purposes.

IBM Enterprise Key Management Foundation offers a centralized solution to manage all keys and certificates for different systems, even if the systems are geographically separated. Key management is performed by using a single set of enforced operational key management procedures and also a single set of recovery and audit procedures.

Keys are managed securely throughout their lifecycle, from generation, to distribution, replacement, and expiry, until they finally are destroyed.

From the central management location, keys are pushed to the servers that host the applications that eventually use the keys. Distribution of keys can be performed within and across different IBM parallel sysplexes independent of the systems having different master keys.

High availability and full recovery of keys and certificates

Keys and certificates under the control of IBM Enterprise Key Management Foundation can be distributed in real time to the systems where their presence is required. These keys are always restorable, even in disaster situations using normal backup/recovery procedures.

Segregation of duties

Daily key management work is performed by security officers that are specially trained to work with keys and who are aware of the rules and procedures that are related to key management. Their main role is to generate and distribute keys to different cryptographic devices, but also to exchange keys with other organizations. The configuration and maintenance of your system, and preparing the key management procedures, is performed by a different set of people in your organization; these people are not supposed to manage the keys.

IBM Enterprise Key Management Foundation supports segregation of duties, and based on your role concept, allows you to create groups and group-of-groups of people with the same responsibilities, where special tasks automatically are enforced for dual access and logon.

Flexibility

IBM Enterprise Key Management Foundation can be customized in a way that matches the system setup, the business application keys, the procedures, and the organization of the client. Different operations can be separated logically and customized within different menus for different groups of users with different access to cryptographic services.

Increased productivity

A large variety of high-level programming interfaces are offered on top of IBM Enterprise Key Management Foundation, which provides you with many already implemented functions and nonstandard cryptographic services. This facility can save an organization a great deal of development costs and can save time for other purposes.

2.1.3 IBM Enterprise Key Management Foundation functions

IBM Enterprise Key Management Foundation makes it possible to generate, store, distribute, exchange, use, and delete DES/TDES, AES, RSA, and ECC keys securely based on PCIe 4765 Cryptographic Coprocessor (IBM 4765).¹ IBM Enterprise Key Management Foundation is a comprehensive key management system with many features that are aimed at making the tasks of key management as easy as possible without compromising security.

Basic functions for key management

Basic functions in IBM Enterprise Key Management Foundation include the following ones:

- ▶ Centralized management of cryptographic keys through their lifecycle, also for different, geographically separated locations.
- ▶ Secure key generation within the physically secured boundary of an IBM 4765 using a true random number generator.
- ▶ Keystore management for writing keys to the Integrated Cryptographic Service Facility (ICSF) and IBM 4765 keystores on servers, which are connected to IBM Enterprise Key Management Foundation.
- ▶ Separation of keys based on their usages and purposes.
- ▶ Backup of all key material into the IBM DB2® Key Repository.
- ▶ Role-based, fine-grained access control system that is based on access rights to functions, keys, customization data, and cryptographic operations.
- ▶ Enforced dual control for specific operations.
- ▶ Crypto hardware support for IBM platforms with cryptographic offerings. Detailed information can be found at this website:
<http://www.ibm.com/security/cryptocards/pciicc/overproduct.shtml>
- ▶ Key management and provisioning for several non-IBM cryptographic platforms and software keystores.
- ▶ Support of every cryptographic entity on the network (terminals, institutions, HSMs, and z/OS systems), with each entity having its own key hierarchy.
- ▶ Usage of key templates allow for the separation of the privileges for defining and creating keys.

¹ The IBM 4765 is a FIPS140-2 level 4 certified tamper detecting and responding cryptographic device that offers the highest security level possible, as a key never appears in the clear outside the secure hardware.

Basic DES/TDES functions

Besides the above listed functions, IBM Enterprise Key Management Foundation DES² functions include:

- ▶ Support for management of several key types for different purposes by using the concept of control vectors
- ▶ Key distribution and exchange in clear parts or encrypted with other DES or RSA keys
- ▶ Organization-specific key mailers for distribution of keys

Basic RSA functions

IBM Enterprise Key Management Foundation provides the following RSA-specific functions:

- ▶ Secure generation within the secure boundary of the IBM 4765 of up to 4096-bit RSA³ keys, with exponent 3, $2^{16}+1$, and random
- ▶ Support for specific RSA key purposes, for example, key management purposes for encrypting symmetric keys, or for signature purposes
- ▶ Public key exchange in the following formats:
 - IBM CCA token
 - PKCS#7
 - PKCS#10
 - X.509 V3 certificates
- ▶ Private key exchange with other IBM Enterprise Key Management Foundation systems or Trusted Key Entry (TKE) Workstations, including import in PKCS#12 format with or without password

Basic AES functions

IBM Enterprise Key Management Foundation provides the following essential functions for AES key management:

- ▶ Storage of AES keys in the EKMF Key Repository, which are protected by RSA public keys or AES keys
- ▶ Distribution of AES keys, which are protected by RSA public keys or AES keys
- ▶ Import and export of AES keys, which are protected by RSA public keys or AES keys
- ▶ Managing of AES keys in keystores, status change, remove, and restore

² With DES, we denote further in this document all DES and TDES-related issues.

³ The IBM cryptographic hardware implements the ANSI 9.31 standard for RSA key generation.

Basic ECC functionality

IBM Enterprise Key Management Foundation provides the following ECC specific functions:

- ▶ Secure generation within the secure boundary of the IBM 4765 with up to 512-bit ECC keys using the Prime curve type
- ▶ Support for specific ECC key purposes, for example, key management, signatures, or both
- ▶ Public key exchange in the following formats:
 - IBM CCA token
 - PKCS#10
 - X.509 V3 certificates

PKI and certificate management

The Public Key Cryptography Standards (PKCS) are specifications that are produced by RSA Laboratories. IBM Enterprise Key Management Foundation lets you participate in the creation of a Public Key Infrastructure (PKI) by supporting the key relevant mechanisms according to the PKCS specifications:

- ▶ Generation of public/private key pairs and creation of PKCS#10 certificate requests for public keys
- ▶ Importing the CA self-signed public key together with the certificate and also with the CA signed certificate in X.509 or PKCS#7 formats or from a PKCS#12 file
- ▶ Monitoring of keys and certificates from an application point of view, giving an overview of the certificate's status and ensuring that certificates are replaced before expiry

The certificate management functions support both RSA and ECC keys.

RACF certificate management

The management of certificates in RACF can be time consuming and inefficient because of the complexity of the **RACDCERT** command that is used for this purpose. IBM Enterprise Key Management Foundation provides better search facilities for searches in the RACF database and direct links between rings and certificates, facilitating faster overview and easing management work. This work includes the following tasks:

- ▶ Key generation and installation of a private key in ICSF
- ▶ Installation of certificates in a RACF database
- ▶ Key ring maintenance, which is associating certificates and key rings, and basic key ring management, such as adding and removing key rings

SSL key management

SSL key management support centralizes and unifies most of the tasks traditionally performed manually for components using SSL. Furthermore, functions are offered that ease administration of certificates for a large population of SSL servers. IBM Enterprise Key Management Foundation supports numerous SSL server implementations by building keystores in any of the PKCS#12, KDB, or Java Key Store formats.

EMV support

EMV is a common specification for integrated chip (IC) based payment cards that are developed by EMVco, an organization that was established by Europay International, MasterCard, and Visa.

IBM Enterprise Key Management Foundation covers the whole spectrum, from Brand Certificate Authorities over EMV card issuers to acquirer functions. It includes the following items:

- ▶ Brand Certificate Authority (CA) function
 - Generation of the CA root key and certificate
 - Receiving certificate request from issuers and signing the issuer public key (Visa, MasterCard, and American Express formats)
- ▶ Issuer certificate handling with regard to the card organizations CAs (Visa, MasterCard, and American Express formats)
- ▶ Card issuing functions, such as signing static data for Static Data Authentication (SDA), derivation of card-specific DES keys, and RSA key generation and certification to support Dynamic Data Authentication (DDA) and Combined DDA/generate application cryptogram (CDA)

x.509 certificate authority

Besides the capability to request certificates for private keys, IBM Enterprise Key Management Foundation also offers the capability to issue X.509 V3 certificates. This particularly is interesting for internal servers or employee certificates, where an externally known, public CA root key is not needed for verification.

The CA function includes the following items:

- ▶ Generation of the CA root key
- ▶ Extraction of the root key in a self-signed certificate
- ▶ Generation of an Internet conforming X.509 V3 certificate from a PKCS#10 request

ATM remote key load

Newer ATMs support an RSA key based key exchange scheme. In such a scheme, the acquirer must have a public key that is certified with the ATM vendor. IBM Enterprise Key Management Foundation supports this process for Cryptera⁴, Diebold, Hyosung⁵, NCR, and Wincor Nixdorf Encrypting PIN Pads (EPPs). Furthermore, IBM Enterprise Key Management Foundation offers APIs that can generate terminal master keys (TMKs) in the formats that are required by the major vendors and supports the various RSA-based key exchange protocols.

PIN print

The PIN print function is based on customized PIN-mailers. IBM Enterprise Key Management Foundation can receive and print PIN codes that are generated, but it can also generate the PIN codes, print them, and return the encrypted PIN codes.

Terminal support

Terminal support provides keys for various crypto-modules in ATMs, POS terminals, and other terminals, for example, NCR ATM, Nixdorf ATM, Diebold ATM, Visa Cash Secure Application Modules, IBM 3624, and IBM 478x ATMs.

MasterCard on-behalf key management

The MasterCard on-behalf feature provides support for the key exchange that is needed if the bank wants to use the MasterCard on-behalf services. Both the two- (DES only) and three-layer (RSA+DES) key exchange hierarchies are supported.

2.2 Logical and physical components

This section describes the logical and physical components of the IBM Enterprise Key Management Foundation system, their purpose, properties, and how they interact.

⁴ Formerly Sagem/BBS Denmark

⁵ Nautilus Hyosung

2.2.1 Component overview

Figure 2-1 shows the components of a basic IBM Enterprise Key Management Foundation (EKMF) system in a typical configuration.

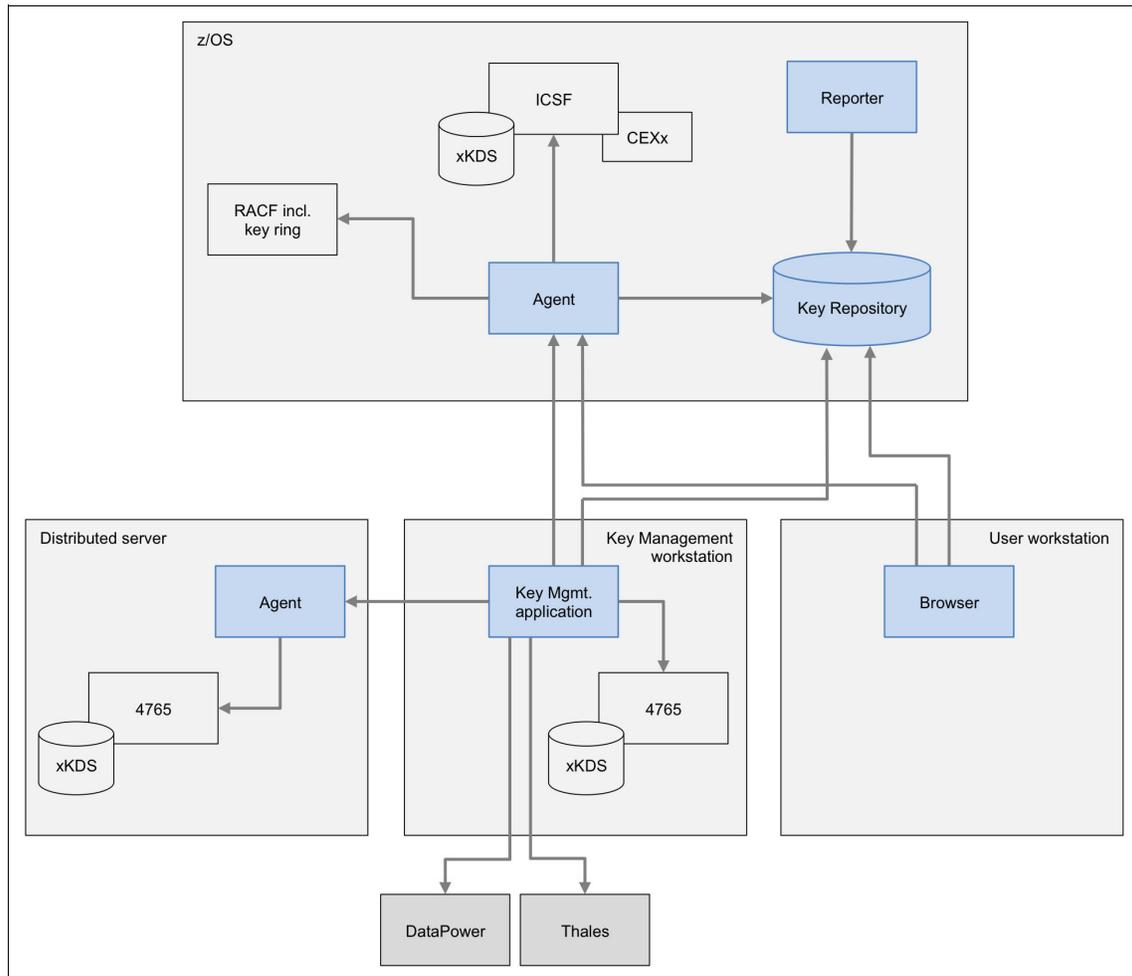


Figure 2-1 Basic IBM Enterprise Key Management Foundation system in a typical configuration

Key Repository

Central to IBM Enterprise Key Management Foundation is the Key Repository. The Key Repository is a DB2 database containing all the keys and certificates that you manage with IBM Enterprise Key Management Foundation. It also contains key templates describing the rules and requirements for the different key types, and it contains policies of a more general nature. Furthermore, the Key Repository contains information that is used by various applications.

Key Management Workstation

The Key Management Workstation contains the Key Management Application, which provides the main interface to key management. If you are a key manager, this is where you log on to manage keys and certificates. If you are an administrator, this is where you log on to configure properties and policies.

The Key Management Workstation is a dedicated workstation using an IBM PCIe 4765 Cryptographic Coprocessor for all sensitive cryptographic operations. It must be placed in a secured environment. You can configure the workstation to require multiple users to log on using personal smart cards.

The workstation can connect through a network to remote Agents to manage the Key Repository and to manage keys in IBM 4765, ICSF, and RACF keystores. The application uses a direct JDBC connection to manage some parts of the Key Repository. The workstation also connects through a network to manage keys in IBM DataPower and Thales devices. All network connections can be protected with encryption.

Browser

The Browser provides an interface that lets you perform tasks that safely can be done without the need for the level of security that is provided by the Key Management Workstation.

The Browser runs on a user workstation. It lets you browse the Key Repository, RACF, IBM 4765, and ICSF keystores, and it lets you issue requests for key management tasks to be carried out by key managers on the Key Management Workstation.

The Browser connects through a network to remote Agents to browse the Key Repository and IBM 4765 and ICSF keystores. The Browser uses a direct JDBC connection for browsing some parts of the Key Repository. All network connections can be protected with encryption.

Reporter

The Reporter is a started task running on z/OS. You can configure it to monitor the Key Repository and generate and distribute reports by email based on custom defined surveillance criteria. For example, you can create reports on expiring keys and certificates.

Agent

The Agents provide access to local IBM 4765, ICSF, and RACF keystores and to the Key Repository. Agents are tasks that can run on IBM System z, IBM System x, IBM System p®, and IBM System i®. The Key Management Workstation and the Browser connect to the Agents to access keystores and the Key Repository.

IBM 4765, ICSF, RACF, DataPower, and Thales

These are the cryptographic systems and devices whose keystores can be managed by using the Key Management Workstation.

The following sections provide more details for some of these components.

2.2.2 Key Repository

IBM Enterprise Key Management Foundation keeps managed keys and certificates together with related information in the Key Repository.

Figure 2-2 on page 25 shows an overview of the Key Repository.

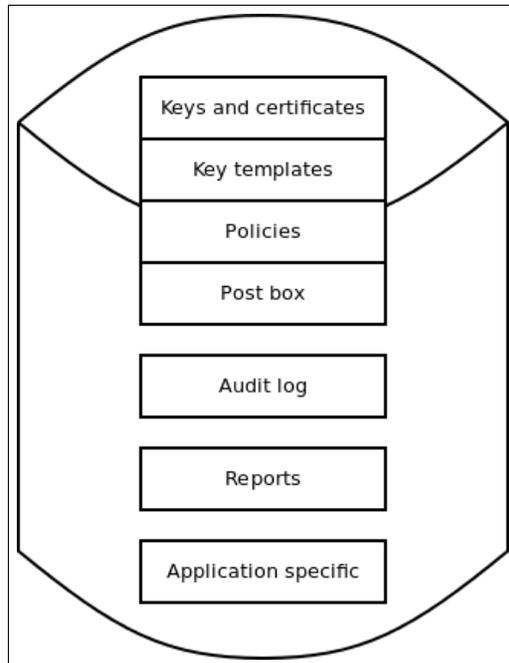


Figure 2-2 Key Repository

Keys and certificates

The repository contains records for all the managed keys and certificates. The information includes the keys and certificates themselves, key check values, relationships between keys and certificates, distribution information, and an event log for each key and certificate.

Key templates

Key templates are, as their name implies, not keys, but templates for keys. To generate or import a key into the Key Repository, you need a key template. Key templates describe the attributes and policies that are related to individual key types, for example, name, algorithm, key size, and distribution information. They are used when generating, importing, exporting, printing, and translating keys, and each template can be used to create many keys.

Policies

You can define system-wide policies that the key management application verifies before key management operations can be carried out. For example, you can define that single-length DES keys are not allowed, which prevents the generation or import of such keys. Policies are defined in an easily comprehensible manner through simple logical expressions.

Post box

The main purpose of the post box is to facilitate workflow-based key management. The post box is basically a list of pending tasks. For example, when you use the browser to request a key renewal, the browser posts the request in the post box. When key managers later log on to the key management workstation, they can see and run the request. It also is used for certain tasks that cannot be carried out as single operation by a single person, such as the entry of a key in multiple key parts.

Audit log

The audit log provides a chronological audit trail for operations that are carried out through the key management application.

Reports

All reports that are generated by the Reporter are stored here. Key Managers can log on to the key management workstation to examine the reports and run appropriate key management actions that are based on report content.

Application-specific information

Some of the optional features and APIs that are available for IBM Enterprise Key Management Foundation, such as the Remote Key Loading, X.509 CA, and EMV features, keep various application-specific information in the Key Repository.

The Key Repository is a DB2 database that can be deployed on IBM System z and IBM System x.

2.2.3 Key Management Workstation

The Key Management Workstation (or EKMF Workstation) hosts the Key Management Application. This is the place where all sensitive key management operations take place. The workstation provides a secure environment with an elaborate access control system, and lets you control who is authorized to perform which of the sensitive key management operations. For enhanced security, *dual control* and *split knowledge* can be enforced.

Common tasks that you can carry out through the Key Management Workstation include the following ones:

- ▶ Key and certificate generation and administration
- ▶ Key and certificate distribution
- ▶ Key and certificate import and export
- ▶ Key and PIN letter printing

The workstation is equipped with specialized hardware and software components to support these tasks. The following sections describe these components:

- ▶ Key Management Application
- ▶ IBM 4765
- ▶ Smart card components
- ▶ Configuration database
- ▶ Printer

Figure 2-3 shows the Key Management Workstation.

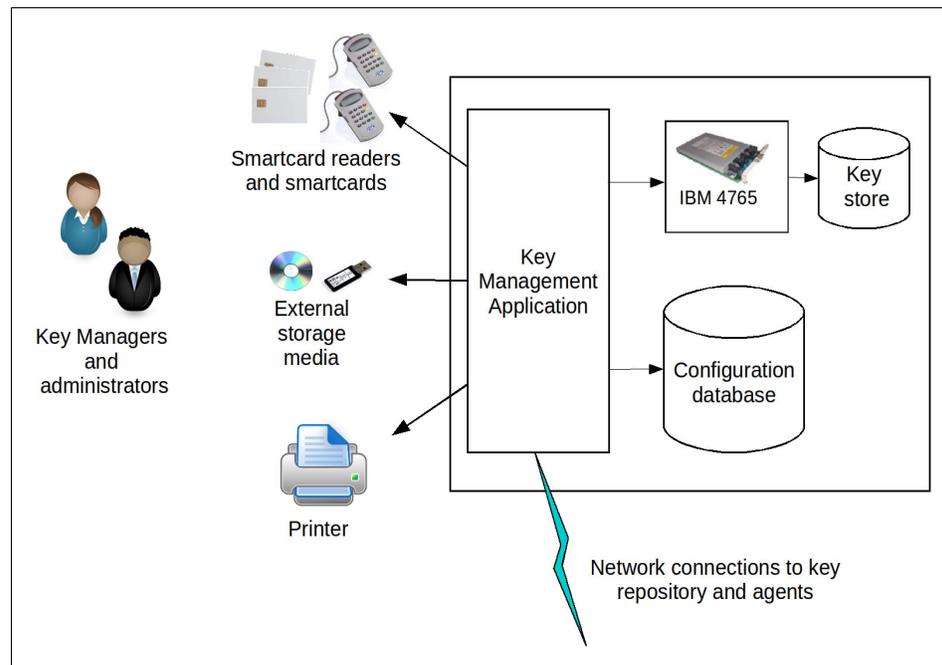


Figure 2-3 Key Management Workstation

Key Management Application

This is a Java based Key Management Application (also known as the DKMS Application) that provides the user interface to all the key management functions.

IBM 4765

This is the tamper-resistant FIPS 140-2 level 4 certified IBM PCIe 4765 Cryptographic Coprocessor (IBM 4765). The Key Management Workstation uses an IBM 4765 to perform all sensitive cryptographic operations, such as key generation, key encryption, signature generation, and signature verification. The IBM 4765 also contains the master keys that protect all keys in the Key Management Workstation key hierarchy. The workstation uses the IBM 4765 hardware enforced access control system to protect keys and cryptographic functions from access by unauthorized people.

Smart card components

The Key Management Workstation uses smart cards for logon, for storage of master key parts, and for the secure entry of key parts. Using smart cards with the Key Management Workstation provides a high level of security because provides the ability to keep key parts and logon keys from ever appearing in clear text.

Smart card support requires the following components:

- ▶ Two smart card readers
- ▶ Smart cards (also called TKE smart cards)
- ▶ IBM 4765
- ▶ Key Management Workstation that is initialized with the smart card option

Smart card support is described 2.5, “Smart card support” on page 36.

Configuration database

The configuration database is an integrity protected database with information that is required for the Key Management Application to operate the following items:

- ▶ Key Management Application access control
Information about the authorizations of users to functions, features, and data of the Key Management Application.
- ▶ Device configuration
Information about the Key Repository, keystores, and cryptographic devices, and how to connect to them through the EKMF Agents.
- ▶ Application data
Information that is related to various EKMF features, for example, the configuration of certificate authorities and the PIN printing application.

- ▶ Miscellaneous data

Other information that is required for the proper and secure functioning of the Key Management Application.

Printer

A printer can be attached to the Key Management Workstation for printing key letters, and (in case you use the PIN printing feature) PIN letters.

Physical protection: It is of paramount importance that you can trust the Key Management Workstation. Therefore, to prevent anyone from tampering with it, you must protect it physically. Place it in a secure room where no person can enter alone.

2.2.4 Browser

The Browser is intended for use in unsecured environments and therefore provides much more restricted access to IBM Enterprise Key Management Foundation than the Key Management Workstation.

The Browser has read access to the Key Repository, RACF, IBM 4765, and ICSF keystores. The Browser does not have any write access, except for the Post Box, where it can write key management requests that you issue from it. The requests do not run until they are approved by a key manager working from the Key Management Workstation.

The Browser provides a graphical user interface to perform the following tasks:

- ▶ Browse keys and key templates in the Key Repository.
- ▶ Browse keys in IBM 4765 and ICSF keystores.
- ▶ Browse certificates and key rings in the RACF database.
- ▶ Browse reports that are generated by the Reporter.
- ▶ Issue requests for key management tasks to be carried out by key managers at the Key Management Workstation. The requests are written to the Post Box in the Key Repository, and can be viewed and processed with the Key Management Application.

2.2.5 Agent

The EKMF Agent provides an interface that the Key Management Workstation and the Browser can use to access resources on the server where the Agent is installed. The Key Management Workstation and the Browser connects to the EKMF through an (optionally) encrypted protocol. The Agent comes in two flavors: one for System z, and one for System x, System p, and System i.

The EKMF Agent for z/OS provides access to ICSF keystores, RACF keystores, and key rings. If the Key Repository is deployed on System z, the Agent provides access to that as well.

The EKMF Agent for Windows, Linux, and IBM AIX® provides access to IBM 4765 keystores. If the Key Repository is deployed on System x, the Agent provides access to that as well.

2.2.6 Reporter

The Reporter provides automatic monitoring of keys and certificates in the Key Repository. It can be configured to generate reports regularly based on searches in the Key Repository looking for specific predefined states in the attributes of keys and certificates or violation of defined policies. An example might be looking for certificates that are about to expire, a task that must be performed at regular intervals. The Reporter replaces manual processes that are performed by key managers and aims to reduce the number of work hours required.

The Reporter is intended to run at regularly scheduled intervals and automatically notify recipients when the monitoring process finds items of interest. Configuration of the reporter determines the scope of keys and certificates to monitor.

The Reporter stores generated reports in the Key Repository. You can query generated reports from the Key Management Workstation. You also can define mailing lists to have the reporter distribute reports automatically by email.

2.3 Sysplex technology

A sysplex (or SYStems comPLEX) consists of 1 - 32 z/OS systems that are integrated into one multisystem environment (somewhat like a cluster in the UNIX world). To be a member of a sysplex, all the participating systems must share a common time source and a common set of data sets (called *Couple Data Sets*). They must also be able to communicate with each other over a set of links called cross-system coupling facility (XCF) signaling paths.

The individual z/OS systems communicate and cooperate through a set of multisystem software and hardware components to process work as a single entity. When individual z/OS systems are integrated into one sysplex, it allows for greater application availability, easier system management, and improved scalability.

2.3.1 System z logical partitioning

An LPAR on System z is the virtualization of the physical hardware through a built-in hypervisor. For the purposes of this book, the LPAR is a distinguished entity sharing resources with other LPARs in an IBM Parallel Sysplex®. Disk-based storage is shared between LPARs running the same operating system. z/OS and its major subsystems, such as DB2, are developed to be most efficient in a Parallel Sysplex. We expand on this concept as we develop the IBM Enterprise Key Management Foundation infrastructure in our design.

As shown in Figure 2-4, resources are shared between multiple z/OS images in a Parallel Sysplex. Keys in the ICSF data sets are accessible from all LPARs, and DB2 data is shared by subsystems running in the different z/OS images. Far more than ICSF and DB2 are shared in this manner, but these are used to illustrate the concept because the IBM Enterprise Key Management Foundation specifically uses these products.

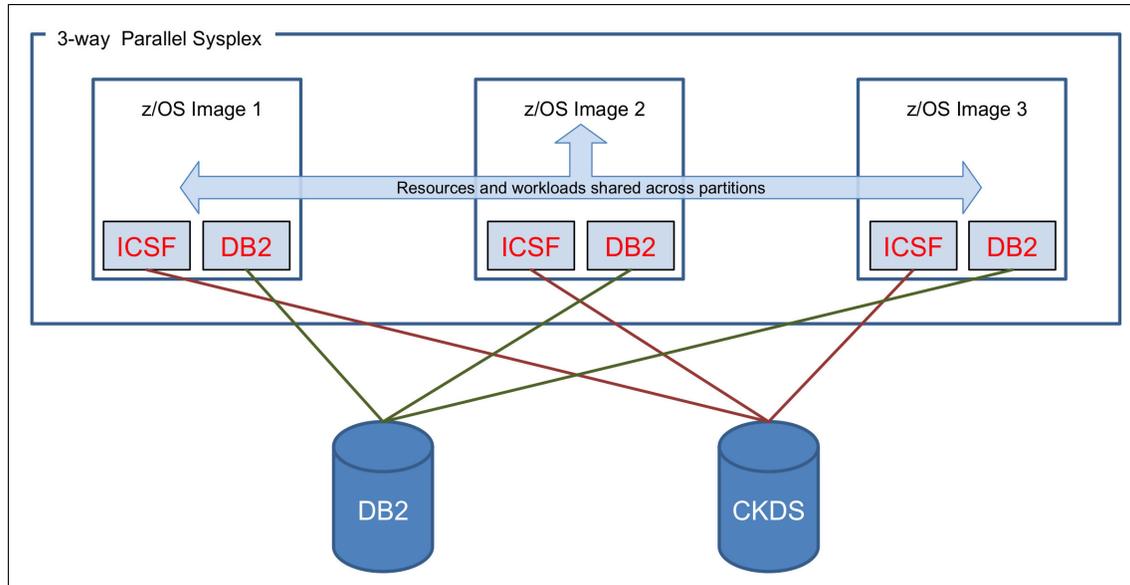


Figure 2-4 Shared resources in a sysplex

2.3.2 Parallel Sysplex usage

z/OS is designed to run in a Parallel Sysplex. Multiple virtual systems running on separate LPARs are interconnected through shared disk storage, and a shared memory device that is called a *Coupling Facility* (CF). A Parallel Sysplex is intended to be a high availability environment with a single system perspective. If any of the constituent systems become unavailable, the workload is distributed to the remaining systems, which allows the installation to apply maintenance to systems in the sysplex without the entire sysplex becoming unavailable.

At our fictional bank, the EKMF Agent for z/OS runs on each z/OS image in the Parallel Sysplex. Each Agent is configured for all required operations using similar configuration files, a shared DB2 database and ICSF data sets, and a common RACF database. This Parallel Sysplex configuration allows for the workstation to connect to any of the z/OS images in the Parallel Sysplex and still maintain a consistent view of the key management environment.

2.3.3 Network architecture

z/OS network capability includes a fully featured Communications Server with integration of SNA and TCP/IP protocols, making the mainframe a large server that can server many worldwide clients simultaneously.

The IBM Enterprise Key Management Foundation uses TCP/IP for communication between the workstation and the Agents.

IBM z/OS Communications Server provides the means to implement a dynamic Virtual IP address (VIPA) as a single network-visible IP address for a set of hosts that belong to the same Parallel Sysplex cluster. Any client that is anywhere in the IP network can see the Parallel Sysplex cluster as one IP address, regardless of the number of hosts that it includes. Using Sysplex Distributor, a workload can be distributed to multiple server instances within the Parallel Sysplex without requiring changes to clients or networking hardware and without delays in connection setup.

In addition, Sysplex Distributor ensures high availability of the IP applications running on the Parallel Sysplex cluster, even if one physical network interface fails or an entire IP stack or z/OS system is lost.

2.4 Disaster recovery

For a key management solution that is built on the IBM Enterprise Key Management Foundation, a disaster recovery (DR) plan is a vital component of securing regular business continuity. Although, in general, a temporary disruption in service is less critical than loss of cryptographic data, the level of importance can vary from one implementation to the next. The ability to recover data in the event of a disaster is an absolute requirement, but the recovery time objective⁶ might be less stringent for some implementations. To effectively meet the various requirements, the IBM Enterprise Key Management Foundation allows for different levels of DR readiness. A lower readiness offers benefits in the form of reduced maintenance but requires additional work for recovery.

This section describes the various components and tools that are available for use in a DR plan.

2.4.1 Key Management Workstation

The Key Management Workstation has some specialized requirements for its DR setup because of the sensitive nature of its operations. For most setups, a specific plan must be formulated.

Required components

The DR plan for the Key Management Workstation should follow the usual preferred practices for DR. In particular, the following items must be available:

- ▶ A remote DR site
- ▶ Duplicate hardware components
- ▶ Backup software and data:
 - The operating system installation media
 - Custom hardware driver media, such as printer drivers
 - The EKMF Installation media
 - An EKMF workstation backup archive file with a checksum file
 - IBM 4765 masterkeys in parts or EKMF workstation recovery keys in parts
 - Any custom documents, settings, or programs that are installed on the Key Management Workstation

For the Key Management Workstation, a DR site must be prepared with the relevant infrastructure in place, including a secure room with the required network, connections, and access protocols for the security officers. Also, the site should be removed physically from the production site.

⁶ The maximum service restore period before the consequences becomes unacceptable.

At the DR site, all the physical components must be available, including the System x server, a crypto adapter, smart card readers, printers, and any other component that is used by the primary Key Management Workstation. The hardware should match closely in make and model with the hardware that is used in production. If the production site uses other items, such as printed procedure documents, these should be available as well.

Backup software components and data must be available at the DR site, including the SLES operating system, EKMF installation media, and custom driver media. Depending on the specific setup, some components already might be installed and configured. Care must be taken to synchronize the software that is available for DR with the software running in the production environment.

A regular backup of the primary Key Management Workstation is used to ensure that changes that are made in the production environment are exported and made available to be applied to the DR workstation. These backup images must be available and regularly updated.

Key Management Workstation backup utility

The backup utility is a small tool that is used to assist in backing up a Key Management Workstation installation and restoring an installation from a backup. The backup feature gathers the data that is required for a backup and stores the result in an archive that can be moved to a safe location. The utility is intended to assist in an otherwise complex backup procedure allowing for faster setup and a simplified process. It is included as a standard component of the IBM Enterprise Key Management Foundation.

The utility supports backing up a workstation running in a stand-alone configuration. In this configuration, the Key Repository is included in the backup.

The archive file that is produced by the backup utility contains settings and configuration data and should be handled as sensitive information. Having access to the archive file corresponds to having access to the Key Management Workstation, but not having login access to the IBM 4765 or Key Management application.

The data that is gathered is the standard configuration data that is required to restore a Key Management Workstation after an unrecoverable failure of one or more components. All the included key material is encrypted.

Here is a list of items that are included in the backup archive:

- ▶ Key Management Application configuration data
When a configuration is found by the backup process, the entire content of the directory is copied to the backup archive, including any custom files that are in this directory and any modified configuration files.
- ▶ IBM 4765 configuration
A backup of the contents of the IBM 4765 is performed. This backup includes the device logon groups and profiles and the keystore files. The masterkeys of the adapter are not included in the backup and must be made available separately. Also, in the case of password-based profiles, the password cannot be backed up, so a new one must be entered during the restore process. Smart card based profiles are not affected by this limitation.
- ▶ Agent configuration
The configuration file from a local Agent in a stand-alone Key Management Workstation is included in the backup file.
- ▶ Local Key Repository
A local Key Repository that is used by a local Agent in a stand-alone Key Management Workstation is included in the backup.

Along with the archive file, the backup process creates a file containing a value that is used to check the integrity of the backup archive upon restore. This integrity file can be stored separately from the backup archive for use in tamper detection of the backups.

For a regular Key Management Workstation setup with the Key Repository on a remote server, the normal day to day work of generating keys does not require frequent backups of the Key Management Workstation because this type of work updates only the Key Repository. New backups are required only when changing settings on or customizing the Key Management Workstation. For a stand-alone Key Management Workstation environment, the backups must run more regularly because the locally hosted Key Repository is continuously updated during the day to day work.

An important tool for setting up the DR site is the EKMF workstation backup utility. This utility can collect and package all of the standard Key Management Workstation configurations, including the roles, profiles, and keystores of the IBM 4765. The master keys are under the protection of the IBM 4765 and cannot be included in the backup, so they must be made available for DR, preferably as parts on smart cards. The backup utility covers only the Key Management Workstation; a remote Key Repository must be handled separately, as described in 2.2.2, “Key Repository” on page 24.

Maintaining DR

Although having all of the required components available at a DR site can be considered a minimal plan with the least amount of maintenance, it is preferable to proceed a little further with the setup to verify the recovery process and the availability of all components.

The process for setting up a DR Key Management Workstation is the same as setting up the production Key Management Workstation, except that no configuration is done. Instead, the configuration data is loaded through the recovery portion of the backup/recovery utility.

Adhering to the restore process can ensure that the DR Key Management Workstation is configured with the same master keys, access controls, and Key Management Application settings as the production Key Management Workstation, essentially establishing a fully functional Key Management Workstation able to take over if the primary Key Management Workstation suffers a breakdown.

To maintain a ready DR Key Management Workstation, changes to the primary Key Management Workstation, such as adding users, changing access controls, customizing basic workstation settings, and adding or changing keys in the workstation keystore, should be applied also to the DR Key Management Workstation. Keeping configuration changes in a backup adds to recovery time, as these changes must be restored at the DR site before it is ready to take over.

More fundamental changes, such as operating system upgrades and upgrades to the IBM 4765 driver and firmware levels, should be replicated on the DR site shortly after upgrading the production server rather than waiting.

2.5 Smart card support

The Key Management Workstation uses smart cards for logon, for storage of master key parts, and for secure entry of key parts. Using smart cards with the Key Management Workstation provides a high level of security because it provides the ability to keep key parts and logon keys from ever appearing in clear text.

Smart card support requires the following items:

- ▶ Smart card readers

The smart card readers provide a keyboard with a direct connection to the inserted smart cards, so that PIN and key part entry can take place without the clear PINs and key parts ever being revealed to the software on the Key Management Workstation.

- ▶ Smart cards

The smart cards are protected with six-digit PINs that must be entered on the smart card reader to access the functionality and data on the smart cards.

- ▶ IBM PCIe 4765 Cryptographic Coprocessor (IBM 4765).
- ▶ Key Management Workstation initialized with the smart card option.

The Key Management Workstation with smart card support can provide the following capabilities.

- ▶ Generate, store, and use an IBM 4765 logon key on smart cards.

To access the key management application and the cryptographic functions in the IBM 4765, users must log on to the IBM 4765. A user can be assigned a personal smart card with a signature key. To log on, the user must present the smart card, enter the corresponding PIN, and then the smart card signs a logon request that is verified by the IBM 4765.

- ▶ Generate and store IBM 4765 master key parts on smart cards.

The IBM 4765 can generate key parts for DES and PKA master keys and store them on smart cards for later loading into the IBM 4765 master key registers. Before exchanging key parts, the IBM 4765 and the smart card establish a secure session to ensure that the key parts always are encrypted when outside the IBM 4765 and the smart card. An IBM 4765 and a smart card can exchange key parts only if they are enrolled in the same smart card zone (for more information, see 2.5.1, “Zone concepts” on page 38).

- ▶ Intermediary storage of key parts during the secure entry of exchange keys.

Entry of clear keys requires special security measures. You do not want clear keys and key parts to appear in the Key Management Workstation memory where they are vulnerable to disclosure and unintended storage on disk in a swap file. Therefore, the workstation keyboard is not a good device for clear key entry. Instead, the workstation uses a smart card with specialized software to encrypt key parts that are entered on the smart card reader keyboard and sends them through a secure session to the IBM 4765 for further processing.

The smart card support on the Key Management Workstation works in a functionally similar way as that of the IBM Trusted Key Entry (TKE) workstation feature.

2.5.1 Zone concepts

Smart card support is designed around the concept of a *zone*, which ensures the secure transfer of master key parts and logon keys. Here are the members of a zone:

- ▶ CA smart card
- ▶ IBM 4765
- ▶ TKE smart cards

TKE smart cards (the naming for these cards is inherited from the IBM Trusted Key Entry (TKE) workstation feature) are general-purpose smart cards that can store key parts or a logon key.

A member of a zone is referred to as an *entity*. Entities must be in the same zone before they can exchange key information. A zone is created through a CA smart card. TKE smart cards and IBM PCIe 4765 Cryptographic Coprocessors are enrolled to the zone.

The zone is checked when exchanging key parts between two TKE smart cards or between a TKE smart card and an IBM 4765. The zone is also checked when a CA smart card is used to unblock a TKE smart card (for example, because of too many incorrect PIN attempts), but not when using the IBM 4765 logon key that is stored on a TKE smart card.

So, the logon key of a TKE smart card that is created in one zone can be used to log on to the IBM 4765 in another zone, but the key parts on the TKE smart card cannot be exchanged in this zone (because the TKE smart card is enrolled in a different zone).

It might be preferable to implement multiple zones, especially if you use multiple Key Management Workstations. In fact, it is a preferred practice to create separate zones for testing and production systems. These separate zones can prevent keys from becoming intermixed. Entities can be a member of only one zone at a time.

Figure 2-5 on page 39 shows a deployment example with two smart card zones.

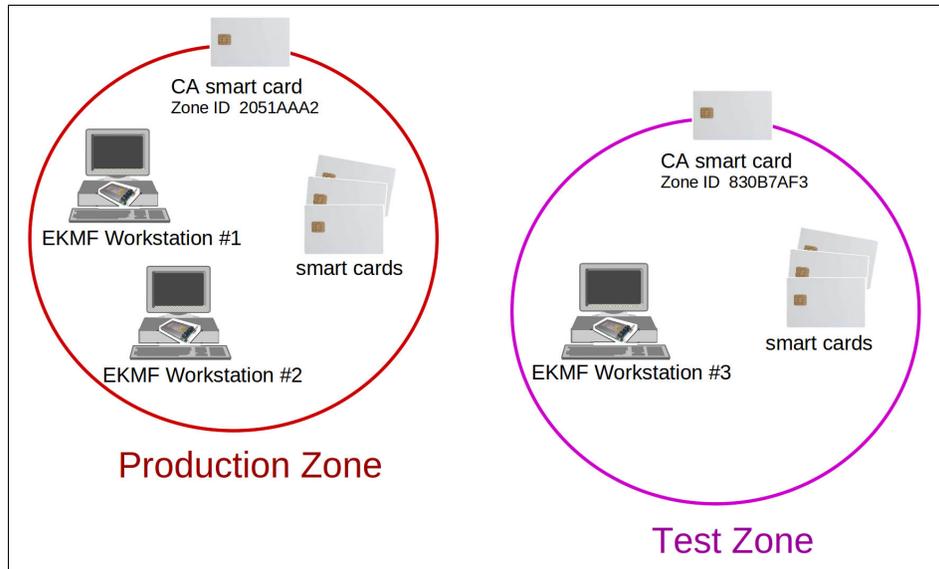


Figure 2-5 Smart card zones

The CA smart card within zone ID 2051AAA2 certifies the smart cards and the IBM PCIe 4765 Cryptographic Coprocessors in Key Management Workstations #1 and #2 in the production zone. Therefore, the smart cards in the production zone can store master key parts and participate in secure key entry with the IBM PCIe 4765 Cryptographic Coprocessors in workstations #1 and #2, and key parts and logon keys can be copied between smart cards in the production zone.

Likewise, the CA smart card with zone ID 830B7AF3 certifies the smart cards and the IBM 4765 in Key Management Workstation #3 in the test zone. Therefore, the smart cards in the test zone can store master key parts and participate in secure key entry with the IBM 4765 in Workstation #3, and key parts and logon keys can be copied between smart cards in the test zone.

Thus, a smart card zone can be regarded as a small PKI with the entities shown in Figure 2-6.

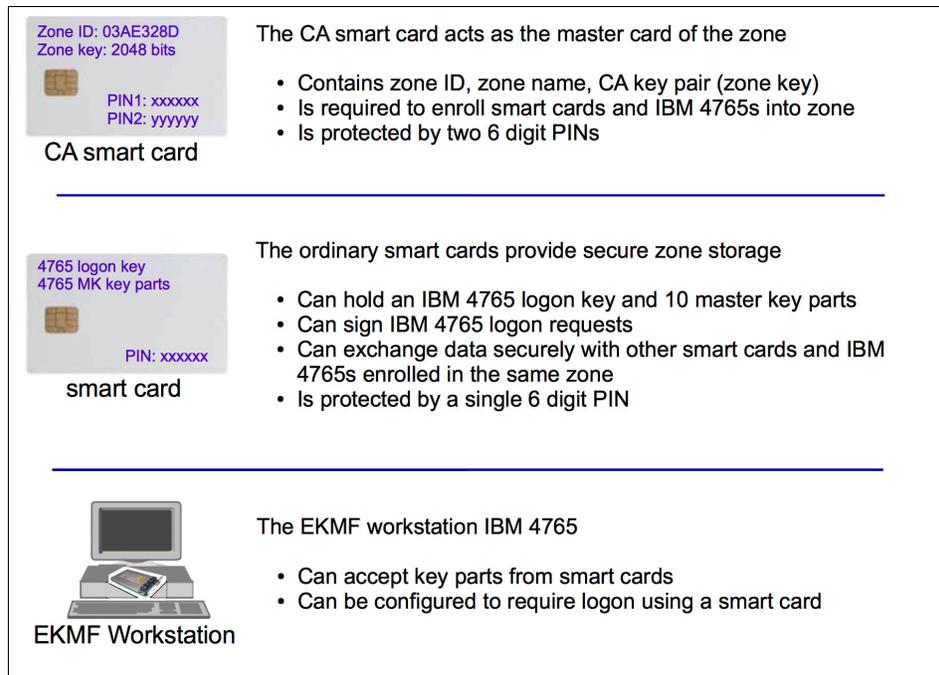


Figure 2-6 Smart card zone entities

2.5.2 CA smart card

When a CA smart card is created, the system generates an eight-digit zone number, which is a *zone ID*. The zone ID has similar properties to the zone description. The main difference is that the zone ID is created by the system. It is derived from the system clock of the workstation that created the CA smart card.

The CA smart card is protected by two six-digit PINs. To ensure dual control, the two PINs should be controlled by different people. Both PINs must be entered for all functions requiring a CA smart card. If either of the PINs of a CA smart card is entered incorrectly five times, the CA smart card is blocked permanently.

As the CA smart card cannot be unblocked, you cannot enroll IBM 4765 cards into the zone, and you cannot unblock any blocked TKE smart cards containing key parts or a logon key.

Have backups of the CA smart card and PINs available. CA backup smart cards with associated PINs are necessary if the original CA smart card is misplaced, destroyed, or blocked.

2.5.3 Enrolling an entity

To enroll an entity into a zone, you need the CA smart card for the zone. With the CA smart card, you can enroll the following entities:

- ▶ IBM PCIe 4765 Cryptographic Coprocessors
- ▶ TKE smart cards

During enrollment, the entity receives and stores the root certificate of the CA smart card. The root certificate is then used to verify other entities that are enrolled in the same zone. Additionally, the CA (smart card) issues a certificate for the entity, enabling the entity to perform the following tasks:

- ▶ Prove to other entities that it is enrolled into the zone.
- ▶ Allow a session key to be encrypted by the public key that is included in the entity certificate to exchange key parts.

The certificate that is issued to the IBM 4765 by the CA smart card is destroyed if you initialize the coprocessor.

2.5.4 TKE smart cards

TKE smart cards can hold the following items:

- ▶ IBM 4765 DES/PKA master key parts (up to 10 parts).
- ▶ Application/exchange key parts (temporarily during secure key entry).
- ▶ One IBM 4765 logon key.

After the TKE smart card is initialized, enrolled in a zone, and personalized, it can store and exchange master key parts (up to 10 key parts), and store one IBM 4765 logon key for logging on to a Key Management Workstation.

During the personalization of a TKE smart card, a six-digit PIN and an optional 20-character card description can be entered. The description can be changed if the TKE smart card is personalized again. The description can be used to distinguish between TKE smart cards.

If the PIN of a TKE smart card is entered incorrectly three times, the TKE smart card is blocked. It is possible to unblock a TKE smart card through the Smart Card Utility Program (SCUP) and a CA smart card of the same zone. The unblocking process resets the PIN failure counter on the TKE smart card. It does *not* reset or change the PIN value.

Separate key parts: The custody of key parts is often split into two or more groups of people, but the logon keys are often recognized as a personal item. Separate key parts and logon key on different TKE smart cards.

Before a TKE smart card can be used to log on to a Key Management Workstation, an IBM 4765 logon key must be generated on the TKE smart card and an administrator must create an IBM 4765 user profile for the user. The primary security feature for a TKE smart card that contains a logon key is the combined possession of the smart card and knowledge of the PIN.

For a TKE smart card that is used to hold key parts, the zone environment is the primary security feature (not the combined possession and knowledge of the PIN). Even if attackers gain access to several TKE smart cards containing all key parts for a certain key, and they manage to get access to the PINs of those smart cards, they cannot access any of the key parts. The TKE smart card exports only its key parts to other entities in the same zone, and the key parts always are encrypted during such transfers.

2.5.5 Reuse of TKE smart cards between EKMF and TKE workstations

As it is often the same person managing both the EKMF Workstation and the TKE workstation, EKMF logon smart cards can be reused on the TKE workstation (and vice versa).

In addition, the logon smart card also can hold an *authority key* (used with the TKE workstation only) resulting in a *one person with one smart card* approach, which can reduce considerably the required number of smart cards in an organization.

Smart cards that are associated with part number 45D3398 or 74Y0551 are supported on both the EKMF Workstation V8.4 and the TKE workstation V7.3.

The zone key length is also compatible with both the EKMF Workstation V8.4 and the TKE workstation V7.3 (either 1024-bit or 2048-bit), but zones always should be separate to avoid using a common master key on separate systems.

There are some TKE smart card compatibility issues between the EKMF Workstation V8.4 and the TKE workstation V7.3 that must be addressed:

- ▶ The applet version that is created by the TKE workstation is 0.8, and the one that is created by the EKMF Workstation is 0.6. This situation can be problematic when you initialize and enroll TKE smart cards on the EKMF Workstation to be used with the TKE workstation.
- ▶ Also, TKE smart cards that are based on applet version 0.6 can contain only up to 10 key parts, where applet version 0.8 implements support for up to 50 key parts.

An IBM 4765 logon key that is created on TKE workstation V7.3 will be 2048 bits, but a logon key that is created on the EKMF Workstation V8.4 is only 1048 bits. However, you can log on with a 2048-bit logon key on the EKMF Workstation V8.4, so if 2048-bit logon keys are important, then generate them on the TKE workstation V7.3.

2.6 Roles and responsibilities

The skills and knowledge that are required to set up, manage, and operate the EKMF Workstation covers many areas:

- ▶ The physical security of the workstation is concerned with a secure room with controlled access, surveillance issues for the workstation, and so on.
- ▶ The installation and future maintenance of the workstation components requires further skills and knowledge:
 - Administration of the SLES operating system, the built-in firewalls, and the SLES active services
 - Administration of the IBM PCIe 4765 Cryptographic Coprocessor
 - Administration of the key management application
- ▶ The backup and recovery of the workstation requires preparation and planning from the people administering the workstation components.
- ▶ The management of smart cards requires some operational procedures and skills for the creation, handling, and maintaining an inventory of smart cards and their contents.
- ▶ The setup and management of the access control systems for the workstation requires administrative skills for SLES and deep knowledge about the access control systems of the IBM 4765 and the Key Management Application.
- ▶ The management of cryptographic keys requires a structured approach and an exact definition of the key types and attributes.

The requirements for these highly diverse skills and knowledge areas should not come as a surprise for an organization that is used to manage cryptographic keys and systems. But, despite the fact that various cryptographic systems have much in common, there are often differences between operating systems, Hardware Security Modules (HSMs), and key management applications, so there usually is a debate about how to reorganize and adapt to existing policies.

It is preferable that a single authority is held accountable for the secure setup and overall operation of the Key Management Workstation.

It is also preferable that an *Administrators* unit within the IT Security organization is responsible for performing all the tasks that are needed to provide and maintain a secure Key Management Workstation. Common obstacles, such as the lack of knowledge or time, should not be accepted as an excuse to push responsibility to others outside the distinct unit. The responsible persons within the Administrators unit should either acquire the required skills, or consult subject matter experts (SMEs) from other parts of the organization.

Consider the common misconception for a Key Management Workstation that a general server maintenance group must be responsible for the workstation and the SLES operating system; this is an unnecessary risk. The Key Management Workstation should be seen as an isolated appliance more than a normally connected workstation or server.

The Key Management Workstation maintains a closed firewall with no open external services. The consequence is that most available patches are not applicable for the Key Management Workstation, and for this reason, IBM recommends not applying general patches to the Key Management Workstation, but only security-relevant patches, as notified by IBM. The Administrators unit must cover “all of the above” in an attempt to provide and maintain a secure Key Management Workstation for others to operate.

To better understand the different roles and responsibilities, look at the following details:

- ▶ Basic concepts
- ▶ Access control systems
- ▶ Role concept

2.6.1 Basic concepts

The most critical part when working with the IBM Enterprise Key Management Foundation is related to how you set up your access control system and how you manage your keys.

You can manage many keys on the Key Management Workstation:

- ▶ The IBM 4765 master keys
- ▶ Some system keys for Key Management Application, such as the keys that are used to secure the network communication or to provide a secure way for integrity checks
- ▶ The application keys, which outnumber all other keys on the workstation

Some of these keys are generated on the Key Management Workstation and some of the keys must be exchanged with a communication partner, either in clear key parts or encrypted under a transport key, which were exchanged previously in clear key parts with the communication partner.

Under these circumstances, there must be a clear separation between the people controlling the key manager processes (often called *key managers*), the people managing the encrypted or clear key material (also called *key custodians*, *key operators*, or *security operators*), and the people who administer the Key Management Workstation (the *Administrators*).

It is a preferred practice to group users that have the same responsibilities so that they can easily be assigned with or restricted from further authorized tasks. It is even more important to establish a separation of duties because you want to ensure that a single individual is not enough to weaken the security of the Key Management Workstation.

Separation of duties

Separation of duties, also known as segregation of duties, is the concept of having more than one person that is required to complete a task.⁷

The primary objective of separation of duties is to prevent any conflict of interest, fraud, and errors.

The second objective of the separation of duties is the detection of control failures that include security breaches, information theft, and circumvention of security controls, policies, and procedures.

⁷ “Separation of duty, as a security principle, has as its primary objective the prevention of fraud and errors. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users. This principle is demonstrated in the traditional example of separation of duty found in the requirement of two signatures on a check.” Quoted from “Separation of duties for access control enforcement in workflow environments” by R.A. Botha, et al, in IBM Systems Journal, Volume 40, Issue 3.

Principally, separation of duties covers both of the following items:

- ▶ Sequential separation, such as the separated custody of clear key parts or the splitting of the passphrase for an administrator account into two halves.
- ▶ Individual separation, such as the ability to log on as a group or to authorize key management functions to a group of users.

The need for sequential separation is perhaps the most easily recognized requirement, but individual separation is as important. The integrity of the Key Management Workstation can be trusted only if it is correctly configured and based on authorized software. To protect the integrity of the Key Management Workstation, it is therefore preferred practice to require participation of at least two people when operating the workstation.

Create a role concept for the tasks to be done to identify the people, and groups of people, that are needed to run the tasks, and also to define their further responsibilities. First, look at access control.

2.6.2 Access control systems

Basically, there are four access control systems in play:

- ▶ Physical access control for the secure room, where the Key Management Workstation is.
- ▶ SLES access control system.
- ▶ IBM PCIe 4765 Cryptographic Coprocessor access control system (CCA access control).
- ▶ Key management application access control system.

We do not go into details about the physical access control, but the preferred practice is a secure room with strong physical protection (even RF shielding). Access must be logged (even monitored or supervised), the participation of at least two authorized people is required to enter the room, and the same two people are required to leave the room.

The other three access control systems are described in the following sections.

Access control system: SLES

The Linux access control system is the first logical line of defense. To a certain extent, it can prevent unauthorized access to the Key Management Workstation, which is preferable, as you want to protect the integrity of the workstation.

Although it is possible to use the Key Management Workstation with the Linux root user account, it is preferable that you create Linux user accounts with more access limitations.

When installing the Key Management Workstation applications, the access rights, called *Access Control Lists* (ACLs) in Linux, that are necessary to use the applications and the various tools and utility programs are granted to the Linux standard group *users*. Because you install the Key Management Workstation applications using the root user, the ownership of these applications is set to the root user and the users user group.

Linux user accounts created later, which also belong to the users group, automatically inherit these ACLs and must be able to start these programs. However, most programs need further authorizations to enable the function.

Depending on your organization's policies, you can use generic Linux user accounts or individual/personal Linux user accounts. Although a generic Linux user account is the least complicated to manage, for an auditor or administrator, it is important to track the user logins on the workstation. The preferred way to do this task is by setting up individual/personal accounts and use them to log on to Linux on the Key Management Workstation.

Access control system: IBM 4765

IBM Enterprise Key Management Foundation components rely on the capability to use the embedded IBM PCIe 4765 Cryptographic Coprocessor. The IBM 4765 implements an access control mechanism that uses the roles and profiles concepts (see Figure 2-7). The roles and profiles are defined using the CCA Node Management (CNM) utility according to your security policy. The CNM utility is included in the installation package of the IBM 4765 and therefore part of IBM Common Cryptographic Architecture (CCA) concept. Access to administrative functions within CNM requires proper user authentication and corresponding privileges to access the administrative functions.

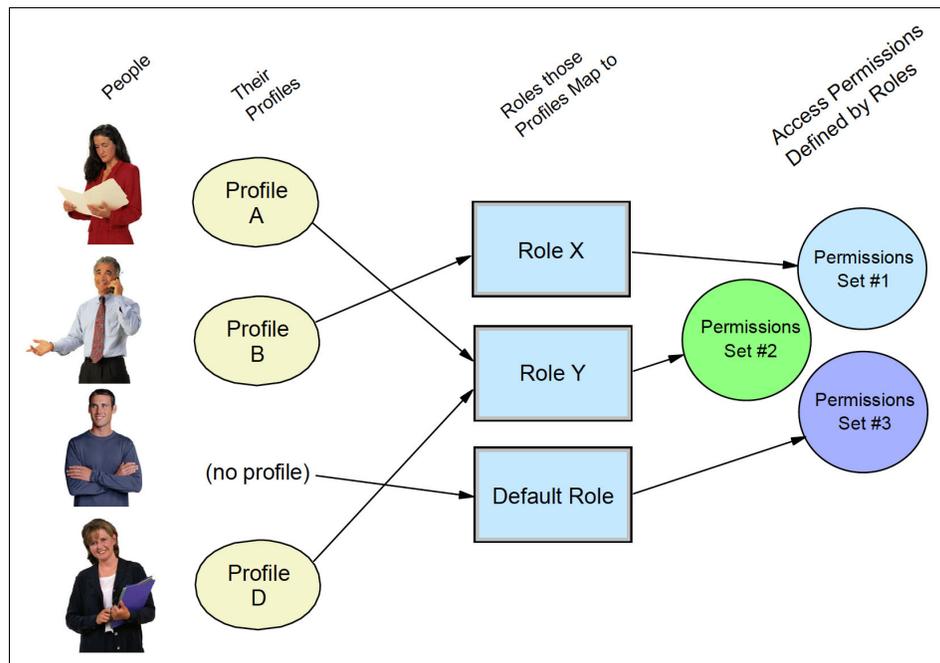


Figure 2-7 Role-based access control

The following subsections provide an overview of the access control concepts of the IBM 4765.

Roles

A role is identified through its Role ID (up to eight alphanumeric characters), and defines a class of users who can run a set of IBM 4765 operations. When creating or changing a role, an administrator defines the IBM 4765 commands and operations (the permission set) that are authorized for users who are mapped to this role. The permission set is a list of active Access Control Points (ACPs) that permit IBM 4765 commands and operations.

A special ACP: There are a few Access Control Points (ACPs) that are designed to restrict IBM 4765 commands and operations when the ACP is set active. One example of such an ACP is the *Disallow Weak Transport Key* command (offset X'0328'), with the ability to *prevent* wrapping a strong key under the protection (encryption) of a weaker key.

The IBM 4765 provides a DEFAULT role. Use of the DEFAULT role does not require users to authenticate themselves through a logon to the IBM 4765. Any user can use the services that are permitted by the DEFAULT role, if they choose not to log on and authenticate themselves to the IBM 4765. Therefore, it is essential that the DEFAULT role is limited to a restricted number of non-critical operations.

Profiles

Profiles are identified by a Profile ID of up to eight alphanumeric characters.

Profile ID: It is recommended that Profile IDs are restricted to alphanumeric *capital* characters, as lowercase characters are not supported by the access control system of the DKMS application.

A profile, in the simplest form, defines a specific user and is mapped to only one role, although as many profiles as needed can be created and mapped to the same role.

After a logon to the IBM 4765 with a specific profile, the user can perform only the actions that are authorized by the role that is mapped to the profile.

Depending on the policies and the requirements you might have in your organization, you can choose between passphrase profiles, smart card profiles, or a mix of both. For added security, it is preferable that you choose smart card profiles whenever possible (an initial but temporary passphrase profile always is required).

Smart card support: The IBM 4765 always should be installed with support for smart cards even if they are not required at the present time. Adding the support to an existing installation later requires updating the IBM 4765 support program installation and reinstalling the DKMS application.

Group profiles

A group profile has almost the same structure as a user profile, but the authentication mechanism for a group profile relies on the authentication mechanisms of m out of n configured user profiles. So, a group profile is not mapped only to a role, but it also maps to some user profiles of the same type (either passphrase or smart card profiles).

The scope of the group profile depends on the number of members (m) that is required to authenticate the group. If you require only one member to authenticate, then it is a convenient way to track individual user profiles with similar access patterns. If you require two or more members to authenticate, you have a group logon where m (out of n) configured user profiles must be successfully authenticated to authenticate the group and therefore enforcing the presence of multiple individuals.

Figure 2-8 shows an example of a group profile SO1 containing n members (user profiles), SO11 to SO1 n , where only one member out of n is required to log on.

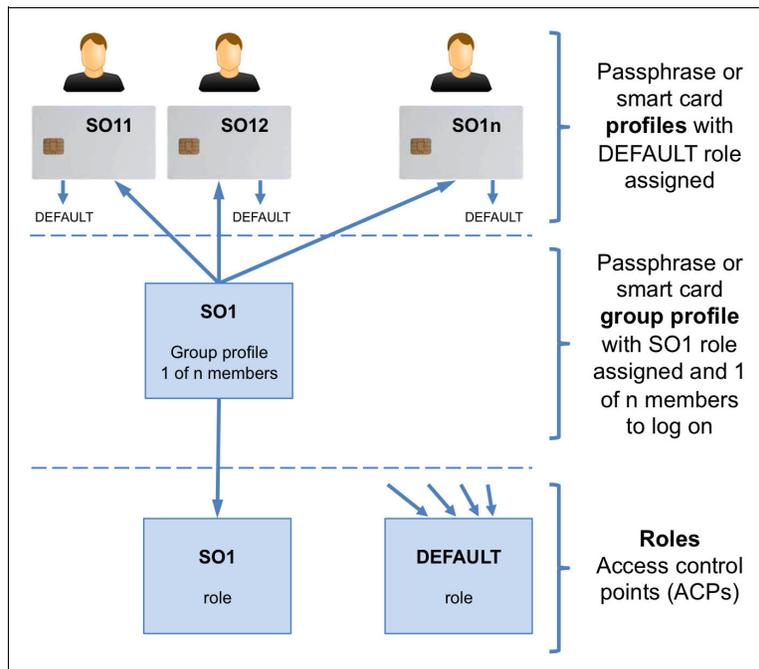


Figure 2-8 Group profile concept

The individual SO11 to SO1 n user profiles are all mapped to the DEFAULT role, which does not allow any key management or administrative functionality, so it is pointless to attempt a logon directly under any of these user profiles. But the SO1 group profile is mapped to the SO1 role, which enables some cryptographic functionality, so an attempt to log on to the SO1 group profile can be carried out by one of the individual SO11 to SO1 n user profiles.

This single logon group profile concept can be used when security officers must manage specific key parts of a key, where the organization or national policies mandate a single logon requirement.

Note: Using the single logon group profile concept that is illustrated in this example, which does not enforce dual logon, might threaten the integrity of the EKMF workstation.

Group-of-groups profiles

A *group-of-groups profile* has almost the same structure as a group profile, but the authentication mechanism for a group-of-groups profile relies on the authentication mechanisms of w out of v configured group profiles. A group-of-groups profile is mapped to a role, and is also mapped to some group profiles of the same type (either passphrase or smart card groups).

The support for group of groups is useful if you want to authorize m out of n users from w out of v groups of users.

Figure 2-9 shows an example of a group-of-groups profile SO containing two members (group profiles) SO1 and SO2, where both members are required to log on.

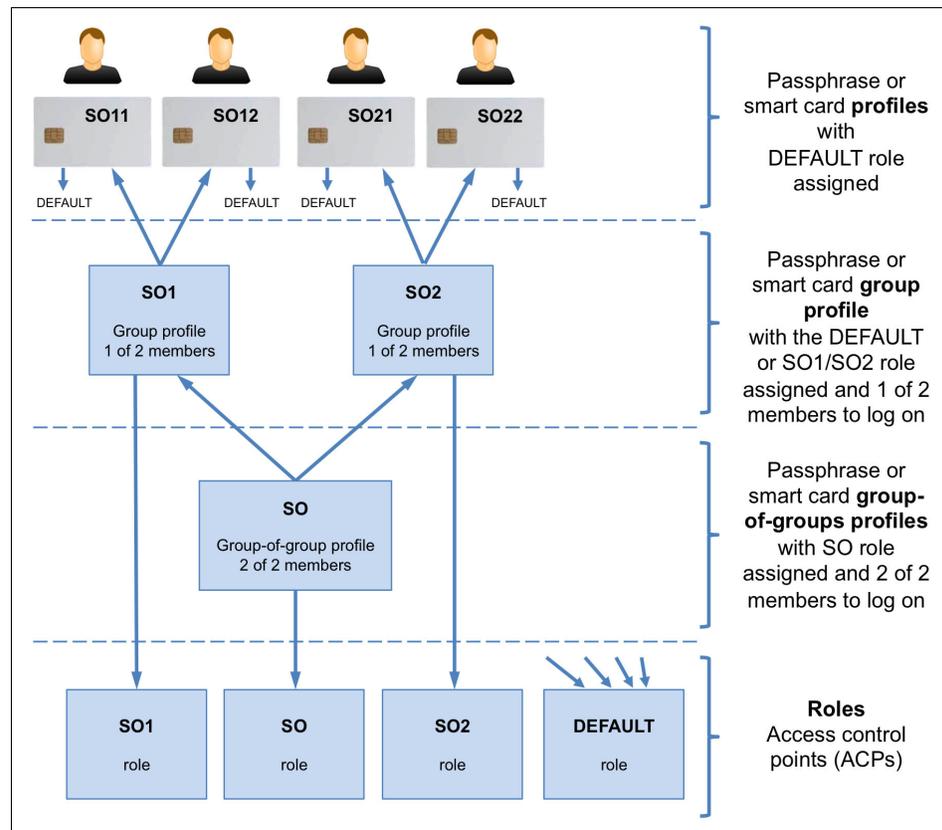


Figure 2-9 Group-of-groups profile concept 1

The SO group-of-groups profile is mapped to the SO role, which enables some cryptographic functionality, which is suitable for a dual logon session. To authenticate the SO group-of-groups profile, one member (user profile) from each group profile must be authenticated successfully.

The two members (SO1 and SO2) are constructed similarly to the previous group profile example shown in Figure 2-8 on page 50 (except that the SO2 group profile is mapped to a corresponding SO2 role).

In this example, a single individual also chooses to log on through one of the group profiles, as each group profile maps to distinct roles that enable cryptographic functionality.

Just an example: This example illustrates an exception to the general preference to enforce dual logon always.

Figure 2-10 shows another example of a group-of-groups profile SO containing two members (group profiles) SO1 and SO2, where both members are required to log on.

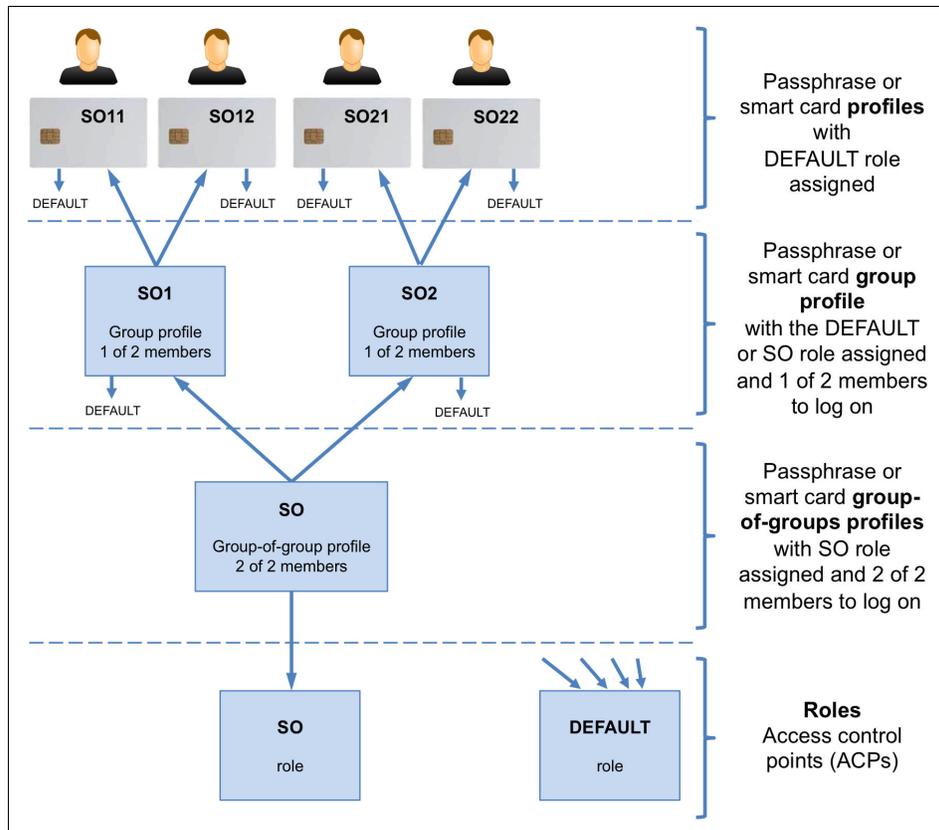


Figure 2-10 Group-of-groups profile concept 2

The two members (SO1 and SO2) are here mapped to the DEFAULT role so that a single individual cannot log on and do anything meaningful.

This example illustrates the general preference to enforce dual logon always (individual separation) to protect the integrity of the Key Management Workstation.

Access control system: DKMS application

The access control system of the DKMS key management application is similar to the role-based access control system of the IBM 4765.

The primary difference between the two access control systems is that DKMS relies on the IBM 4765 access control system to authenticate individual users or a group of users.

Another (minor) difference is with the naming of the access control entities. The IBM 4765 entities consist of (user) *profiles*, which map to *roles*, which enable a *set of ACPs*. The corresponding DKMS access control entities are called (DKMS) *users*, which map to *access groups*, which enable a set of *DKMS application resources*.

The logon to the DKMS application is a logon to a specific profile, group profile, or group-of-groups profile that is defined in the IBM 4765. The profile ID must also be defined as a DKMS user (name) in the DKMS access control system.

Watch out: A profile ID cannot be recognized in the DKMS access control system if it contains lowercase characters.

Also, the profile must map to a role where the active set of ACPs includes the *DKMS Logon* command (offset X'8001'). Otherwise, the profile ID is not visible in the DKMS logon dialog.

Profiles of the IBM 4765 can be mapped to one or more access groups within the DKMS application access control system, and each access group can be configured to authorize a unique predefined set of DKMS application resources, in the form of menu entries, key templates, and programs and functions.

Mapping your profiles: Profiles that are not mapped to any access group typically are visible in the DKMS logon dialog, but a logon attempt using these profiles fails because no DKMS resources are authorized.

Also, profiles that are mapped to more than one access group enable the combined set of resources that are associated with the linked access groups. The general recommendation for any access control system is to keep it simple, so you should try to avoid multiple access groups for a profile.

A DKMS access group is identified by its access group ID of one alphanumeric capital character (0 - 9 and A - Z). The access group ID "1" automatically is created and is reserved for administrative purposes. Profiles that are mapped to access group ID "1" generally enjoy a powerful set of administrative privileges.

In addition, there is the following small set of DKMS *super user* privileges that cannot be achieved alone through an access group membership:

- ▶ Recalculation of integrity information⁸ (MAC) on the following items:
 - DKMS application configuration data and tables
 - UKDS6 and RSA Key Repository dB tables
- ▶ Delegation of access to Key Definitions and Key Templates
- ▶ Creation of a Device Group under Device Configuration Management

To become a DKMS super user, you must log on through a profile named ADMIN% (the profile name must be ADMIN, or start with the word ADMIN), and this profile must be associated to DKMS access group “1”.

Associating the ADMIN profile for DKMS: The DKMS super user mode is required initially and occasionally later on. It is therefore important to have at least one ADMIN% profile (group profile or group-of-groups profile) that is associated with the DKMS access group “1”.

To run a specific task in the DKMS Key Management application, you must have the necessary access rights in both environments:

- ▶ The DKMS Key Management application
 - The access group that the logon profile is assigned to must have the corresponding authorizations to the specific DKMS Key Management application resources.
- ▶ The IBM 4765
 - The role that is assigned to the logon profile (or group or group-of-groups profile) must have the corresponding authorizations to the specific ACPs in the IBM 4765.

Additional DKMS access groups should be created and named depending on the roles and responsibility of the individuals belonging to that specific group.

2.6.3 Role concept

The fine granularity of the IBM 4765 access control system allows you maximum flexibility in planning the roles, the users, and group of users working with the cryptographic function.

⁸ There also is integrity information in the Key Template table and in the general UKDS7 Key Repository tables, but here the recalculation can be authorized to an access group.

Depending on the internal and external policies your organization is following, you might want to create the maximum separation of duties that is possible and also restrict the access control system of the IBM 4765 to the minimum of authorizations that are necessary. Considering that for each individual with a specific role you also might want to provide a second individual as backup in case of emergency, you need a high number of people for running all tasks.

If you are not bound to restrictive regulations or you are short on personnel, you must plan a simplified access control setup, which often ends in melding some roles together or using the same people for different roles. The next sections show some sample role concepts that can be created within the IBM 4765.

In the following sections, we restrict the sample role concepts to only two groups or two members of a group. If you are planning to have, for example, three parts of a key, you must enhance the role concept to have three groups of people, each handling one key part.

Sample role concept 1

Figure 2-11 on page 57 presents a detailed role concept that requires many people, but reflects also the maximum separation of duties and splitting knowledge that are possible.

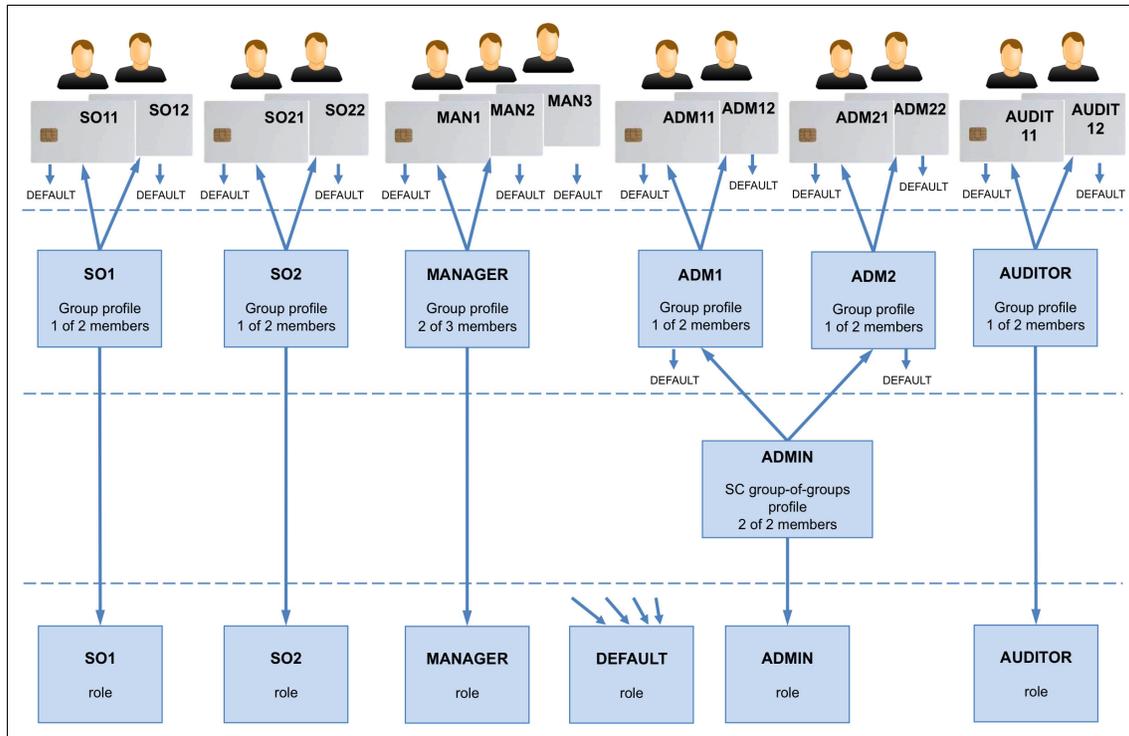


Figure 2-11 Sample role concept 1

The role concept is based on the following roles:

ADMIN

Administers the IBM 4765 and the EKMF application, which manages the access control issues, is the card holder for the CA cards, but has no access to key material. As this is a powerful role, running any administrative task requires at least two people to be present. Therefore, log on as a group to the system, where the first individual logs on using the group ADM1, and the second individual logs on using the group ADM2.

SO1

The first security operator group, and therefore group of key custodians of the first key part. A logon using this role requires only one key custodian to log on, either SO11 or SO12.

SO2

The second security operator group, and therefore group of key custodians of the second key part. A logon using this role requires only one key custodian to log on, either SO21 or SO22.

| | |
|----------------|--|
| MANAGER | Plans the key management-related processes, defines the key types and attributes, manages keys and their lifecycles, and exchanges encrypted keys with communication partners, but has no access to key parts. This role requires two people to log on. |
| AUDITOR | Audits the system, views the logs and audit trails, and verifies that the organizational procedures and processes are followed. This is an optional role, and often the auditor role in an organization is taken by a different department that has no knowledge about the IBM Enterprise Key Management Foundation, and therefore attends the key management operations only as an observer and controller of executed tasks and processes. |
| DEFAULT | Limited to the minimal functionality, such as reading and setting the internal clock of the IBM 4765. |

The following people are involved in this role concept:

| | |
|-----------------------------|--|
| SO11 and SO12 | Members of the group SO1, who can manage only the first key part. They can, therefore, perform a single logon and run the cryptographic functions that are provided by the SO1 role. |
| SO21 and SO22 | Members of the group SO2, who can manage only the second key part. They can, therefore, perform a single logon and run the cryptographic functions that are provided by the SO2 role. |
| MAN1, MAN2, and MAN3 | Members of the group MANAGER, who can run all cryptographic functions that are enabled by the MANAGER role. As these people are also allowed to create and edit key templates, which define the attributes of a key, you must consider that this is a task that should be performed in the presence of two people. Therefore, a group logon of two managers is necessary. Add the ADMIN group-of-groups. |
| ADM1 and ADM2 | Group profiles that contain further single profiles as members. The DEFAULT role is assigned to these group profiles, enforcing that none of the members can perform any administrative activity by means of a single logon. |

ADM11 and ADM12

Members of the group ADM1. This group profile has the DEFAULT role mapped. Therefore, its members are not enabled for administrative tasks when performing a single logon. As the group ADM1 requires one member to log on and the ADMIN group-of-group requires two members to log on, one of the ADM11 or ADM12 and also one of the ADM21 or ADM22 members must perform a group logon to run any administrative functions that are enabled by the ADMIN role that is mapped to the group-of-group profile ADMIN.

ADM21, ADM22

Members of the ADM2 group that is similarly constructed to the group ADM1. Add AUDITOR persons to the list.

Here are the advantages of this role concept continuity:

- ▶ Maximum separation of duties, also enforced at the people level.
- ▶ Maximum splitting knowledge when handling clear key parts.
- ▶ Compliance is fulfilled.

The disadvantage of this role concept is that many people are needed to implement this concept. Even if you do not consider the AUDITOR as a necessary role and individuals, you need 11 people.

Sample role concept 2

In Figure 2-12, we have left out the auditors, the auditor group profile, and auditor role. Auditors are often external people or from a different department of the company who typically do not have access to the EKMF workstation. They ask the managers or the administrator to provide specific audit records.

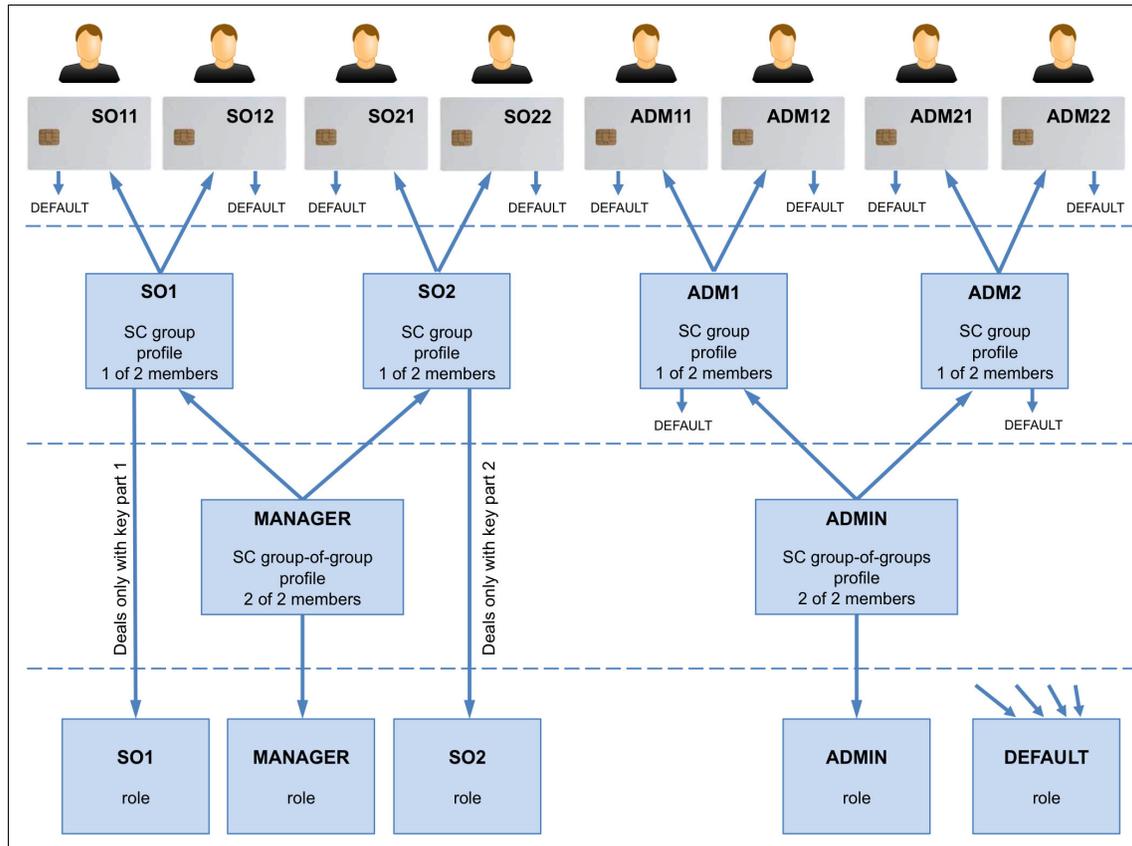


Figure 2-12 Sample role concept 2

The other role names and responsibilities are the same as shown in “Sample role concept 1” on page 56.

The scope of this role concept is to reduce the number of people that is needed to fulfill the key management and administrative functions at the Key Management Workstation to eight people.

The security officers SO11 / SO12 and SO21 / SO22 log on to the IBM 4765 and also to the EKMF application through the group profile SO1 and SO2 to manage key parts, but additionally they can handle encrypted keys too if they are running a group logon through the MANAGER group-of-groups profile.

Reducing the required number of personnel: You can further reduce the number of people to four if you assign each pair of smart cards (SO11 - ADM11, SO12 - ADM12, SO21 - ADM21, and SO22 - ADM22) to individual persons. These persons now have several roles, but from a compliance point of view, they still *act* according to the separation of duties concept with different roles.

Here are the advantages of this role concept:

- ▶ Maximum separation of duties, but here the Security Officers are also in charge of the MANAGER role.
- ▶ Maximum splitting knowledge when handling clear key parts.
- ▶ Compliance is fulfilled.
- ▶ Number of people that is needed is reduced to eight or even four.

The disadvantage of this role concept is that separation of duties is performed at the role level, so the same people act with different roles in the system.

Sample role concept 3

The roles that are shown in Figure 2-13 are again the same as shown in “Sample role concept 1” on page 56. However, this is another attempt to reduce the number of people that is required to operate the same set of roles.

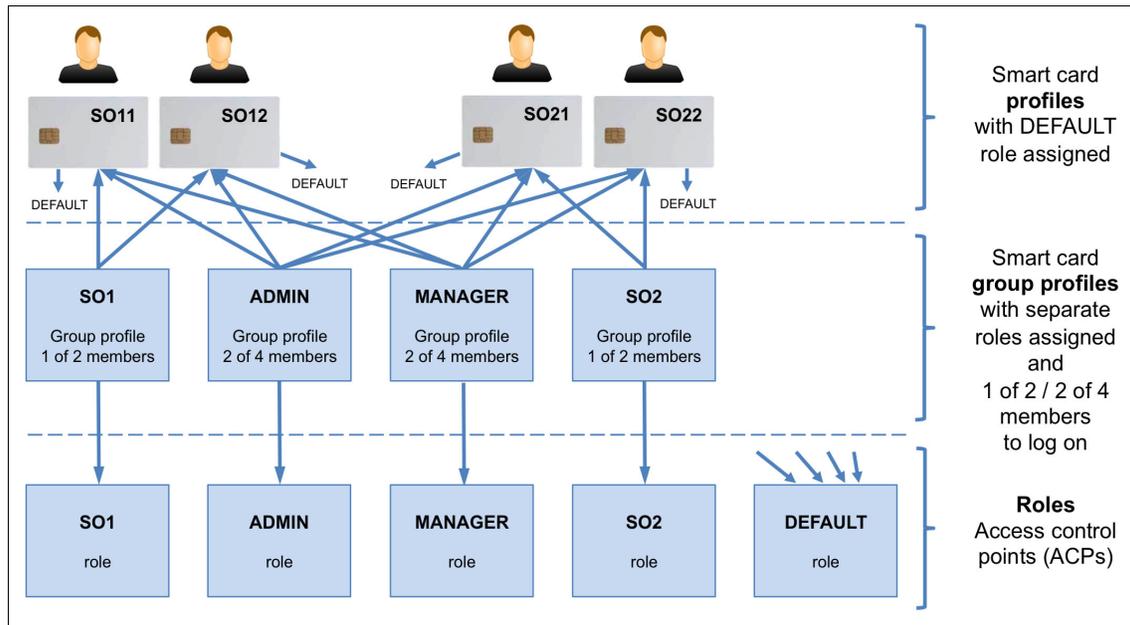


Figure 2-13 Sample role concept 3

The SO1 role and the SO2 role are the same as with the previous sample role concepts.

If you look at the sample from an organizational point of view, you are down to two groups of people (Security Officers), each consisting of two people. To operate the SO1 role, the SO11 user or SO12 user must be present, and to operate the SO2 role, the SO21 user or the SO22 user must be present.

But the two groups combined form a logical third group that can be used to operate the ADMIN role and the MANAGER role. This is a more flexible approach because *any* two out of four users can log on to the ADMIN role and to the MANAGER role because of the fact that these are just group profiles, and not group-of-groups profiles.

As the absence of two out of two persons is more likely to occur than the absence of three out of four persons, you are better off with respect to the MANAGER and ADMIN tasks. That is fortunate because you typically will spend most of your time on key management tasks through the MANAGER role.

Here are the advantages of this role concept:

- ▶ Maximum separation of duties, but here the Security Officers are also in charge of the MANAGER and the ADMIN roles.
- ▶ Maximum splitting knowledge when handling clear key parts.
- ▶ Compliance is fulfilled.
- ▶ Number of people that is needed is reduced to four.

The disadvantage of this role concept is that the separation of duties is performed at the role level, so that the same people act with different roles in the system.

2.7 Migration considerations

A common concern when introducing IBM Enterprise Key Management Foundation in to an organization that already uses a cryptographic function is the steps of adapting or migrating the existing cryptographic infrastructure in a way suitable for the IBM Enterprise Key Management Foundation.

Therefore, within the phase of planning the implementation of the IBM Enterprise Key Management Foundation, an inventory of the existing cryptographic infrastructure, key landscape, and cryptographic applications running within the infrastructure should be completed and correlated with the business and functional objectives of the organization.

Nomenclature: In the following sections, we denote with the word *keys* all symmetric and asymmetric keys that are known at this moment, and both our own and foreign certificates that are related to the keys of an organization.

Here is a list of the aspects that you should consider when planning for IBM Enterprise Key Management Foundation:

- ▶ Physical cryptographic infrastructure
 - Are you using mainframes? Are you also using ICSF and Crypto Express coprocessors on the mainframe?
 - Are you using IBM distributed systems?
 - Which systems?
 - Are you also using cryptography on the distributed systems?
 - Are you using other vendor systems? Which systems?
 - Are you using cryptography products from other vendors? Which ones?

- ▶ Systems that are planned to be provisioned with keys by IBM Enterprise Key Management Foundation
 - Which IBM systems should be provisioned with keys by IBM Enterprise Key Management Foundation?
 - Which platforms and cryptographic devices from other vendors should and can be provisioned with keys by IBM Enterprise Key Management Foundation?
- ▶ Keys and certificates that are planned to be managed by IBM Enterprise Key Management Foundation
 - Are you already using keys on the mainframe?
 - Do the keys exist on the mainframe? Do they need to be imported into and managed by IBM Enterprise Key Management Foundation?
 - In which keystores are the keys?
 - Are there any keys on the distributed IBM systems and should these keys be imported into and managed by IBM Enterprise Key Management Foundation?
 - Are there any keys in the cryptographic devices and platforms that are provided by other vendors? Should these keys be imported into and managed by IBM Enterprise Key Management Foundation?

After comparing the list of the systems to be provisioned to the list of the supported systems, and comparing the list of key types and certificates that are managed without IBM Enterprise Key Management Foundation to the key types and certificates that are managed by IBM Enterprise Key Management Foundation, you find yourself in one of the following situations, or a combination of them. You must consider these situations when introducing IBM Enterprise Key Management Foundation in to your organization.

- ▶ The existing key types are supported by IBM Enterprise Key Management Foundation and the keys are in a format that allows IBM Enterprise Key Management Foundation to import or enter them.
 In this case, you can import or enter the keys into IBM Enterprise Key Management Foundation, and unless you have further situations to handle from this list, no special migration action is necessary.
- ▶ The existing key types are supported by IBM Enterprise Key Management Foundation and the keys are in a format that does not allow IBM Enterprise Key Management Foundation to import them.
 IBM can provide you with the necessary support to migrate these keys from the existing systems and format into a IBM Enterprise Key Management Foundation known and importable format.

- ▶ Additional systems must be provisioned with keys that are supported by IBM Enterprise Key Management Foundation.

One of the greatest features of IBM Enterprise Key Management Foundation is that, besides using it as a backup Key Repository, you can use IBM Enterprise Key Management Foundation also to push or provide these keys online into several keystores on other IBM or non-IBM systems. The migration of keys from these systems into the backup Key Repository of IBM Enterprise Key Management Foundation might be a challenge and require some migration activities.

- ▶ The existing key types or the systems to be provisioned with keys are not supported by IBM Enterprise Key Management Foundation.

IBM Enterprise Key Management Foundation constantly is being developed and improved to cover various keys and systems that are used by organizations. Nevertheless, it is possible that some of the key types or systems you are using are not supported by IBM Enterprise Key Management Foundation. Custom coding for the IBM 4765 and the system is one way to accommodate such key types and systems. IBM will engage with you or with the vendors of the non-supported systems to investigate and elaborate on possible solutions.

2.8 Conclusion

This chapter described the functions that IBM Enterprise Key Management Foundation can provide. This chapter introduced the components that make up the system, and how these components are deployed in an enterprise environment with Parallel Sysplexes. Operational aspects, such as recovery, and role and responsibilities, are other important aspects that were covered. Finally, this chapter described how a migration to IBM Enterprise Key Management Foundation can be achieved.



Deployment, administration, and maintenance

This chapter highlights some of the important aspects that are involved when designing a solution based on the IBM Enterprise Key Management Foundation. These aspects include determining different deployment options, and important administration and maintenance tasks for the initial deployment and the ongoing runtime environment.

This chapter includes the following sections:

- ▶ Understanding deployment options
- ▶ Maintenance of the installation
- ▶ Administering users
- ▶ Providing applicable logging
- ▶ Tracing for troubleshooting
- ▶ Ensuring consistent backup and restore procedures

3.1 Understanding deployment options

IBM Enterprise Key Management Foundation can be deployed in different ways to suit installation-specific requirements, and fit into and support an existing cryptographic infrastructure. This section takes a closer look at some deployment options.

3.1.1 Configurations

A minimal IBM Enterprise Key Management Foundation configuration consists of a Key Management Workstation and a Key Repository. In a stand-alone configuration, the Key Repository is deployed on the Key Management Workstation itself, and in an online configuration, the Key Repository usually is deployed on a separate server, either IBM System z or IBM System x. This section describes some common configurations, starting with the most simple one. For an explanation of the individual components that are used in the configurations, see 2.2, “Logical and physical components” on page 21.

Stand-alone configuration

In a stand-alone configuration, shown in Figure 3-1, the EKMF Key Repository is deployed on the EKMF Workstation itself.

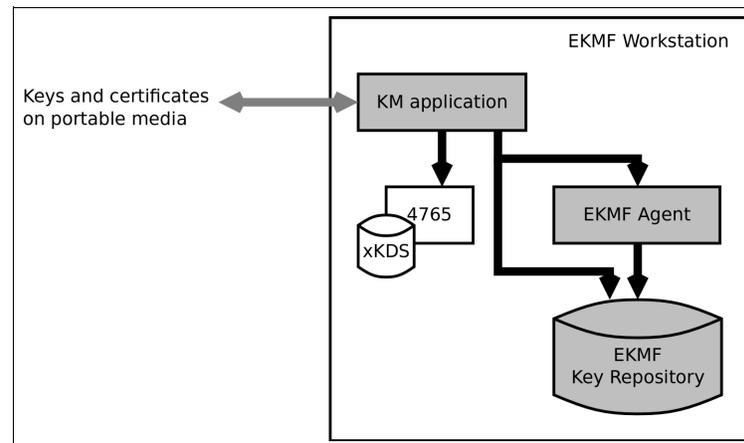


Figure 3-1 Isolated stand-alone configuration

The Key Management Application uses a JDBC database connection and a connection to a local EKMF Agent to access and maintain the EKMF Key Repository. In this configuration, the EKMF Workstation has no network connection, and you must migrate keys and certificates manually on portable media to distributed servers and external parties. This configuration is suitable for certificate authorities, where the protection of private signing keys is of the utmost importance.

Figure 3-2 shows a network-connected, stand-alone configuration that is used for PIN printing.

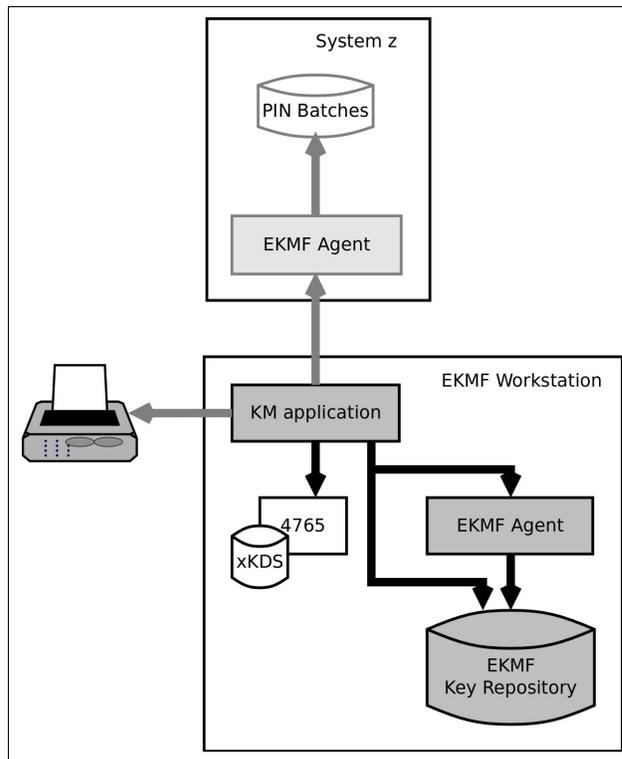


Figure 3-2 Stand-alone configuration for PIN printing

In this configuration, the EKMF Workstation is network-connected to a System z server to retrieve batches of encrypted PINs to be printed on a locally attached printer that is dedicated to printing PIN letters. The keys that are required to decrypt the PINs were exchanged previously through portable media.

Here are the properties of the stand-alone configuration:

- ▶ The Key Repository is on the Key Management Workstation and has the same level of protection as the workstation itself, for example, the protection that is provided by a secure room with controlled access.
- ▶ The Key Management Workstation can be disconnected from the network, thus minimizing the threat of unauthorized access.
- ▶ This configuration is simple to maintain and requires a minimum of hardware.

Online configuration with the Key Repository on System z

In this configuration, the Key Repository is deployed on an System z server. This configuration requires network connectivity. The EKMF Workstation uses a JDBC database connection and a connection to an EKMF Agent on the server to access and maintain the EKMF Key Repository.

Many IBM Enterprise Key Management Foundation APIs and applications for System z require access to the Key Repository. When the Key Repository is deployed on a System z, you can use those APIs and applications.

Figure 3-3 on page 71 shows an online configuration with an EKMF Workstation, distributed servers, APIs, and applications.

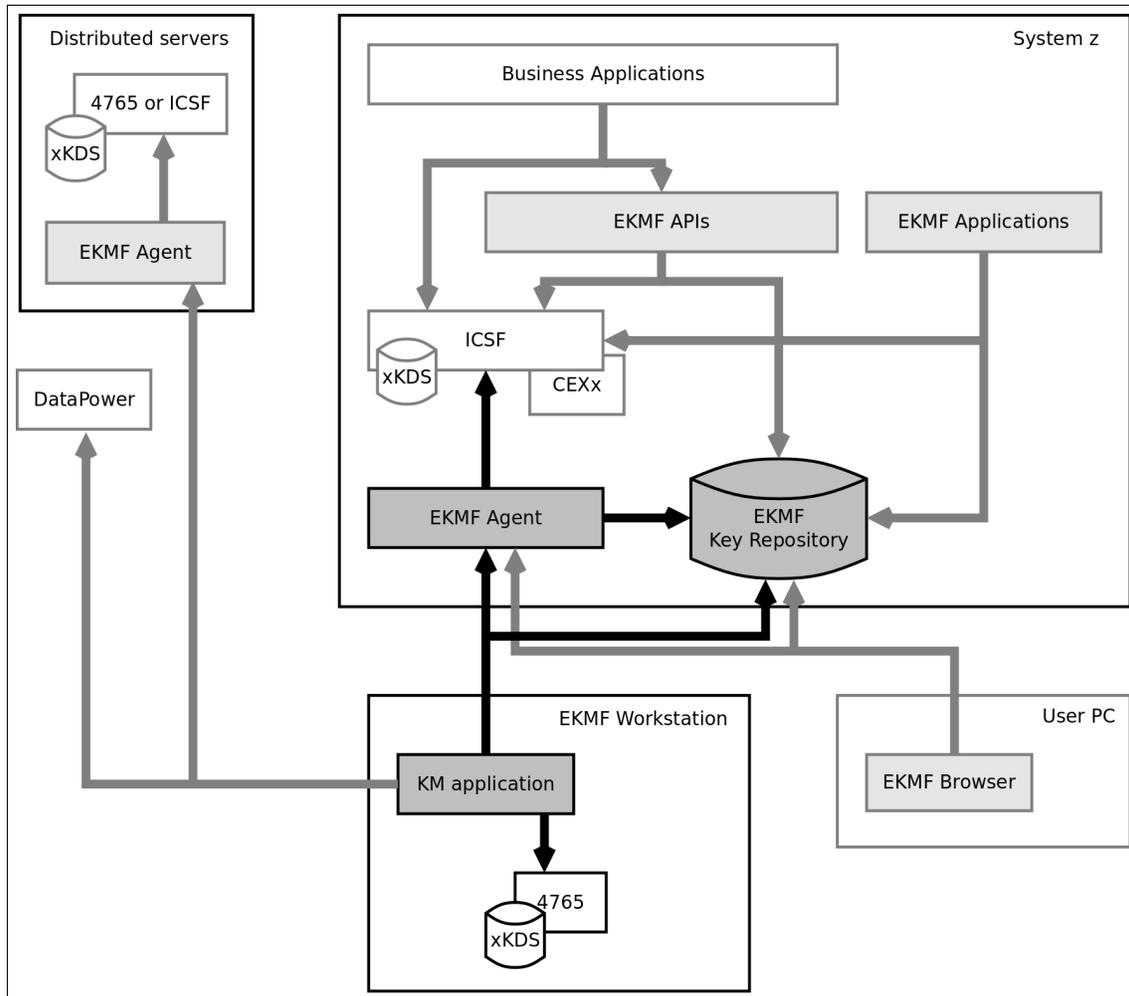


Figure 3-3 Online configuration with Key Repository on System z

Here are the properties of this configuration:

- ▶ The Key Repository is available for use by IBM Enterprise Key Management Foundation APIs and applications on the server hosting the Key Repository. The Key Repository can be shared with other systems through JDBC database connections or sysplex sharing.
- ▶ Multiple Key Management Workstations can access the Key Repository.
- ▶ Supports the Browser and the Reporter.

- ▶ Prerequisite for using some IBM Enterprise Key Management Foundation APIs and applications, including the following items:
 - Reporter
 - Remote key loading API
 - PKCS#1 and PKCS#7 APIs
 - EMV card-issuing APIs
 - eTicket API (some functions)
 - General-purpose API
 - RSA key pool
- ▶ The Key Repository can be backed up as part of the normal DB2 backup procedures.

This configuration (or a similar configuration) is commonly used by financial institutions to deal with transaction processing, terminal handling, and related applications.

Online configuration with Key Repository on System x

In this configuration, the Key Repository is deployed on an System x server. The configuration is similar to the configuration in “Online configuration with the Key Repository on System z” on page 70, where the Key Repository is on a System z. You can use this configuration if you do not need the IBM Enterprise Key Management Foundation APIs and applications that are available on System z.

Figure 3-4 shows such an online configuration.

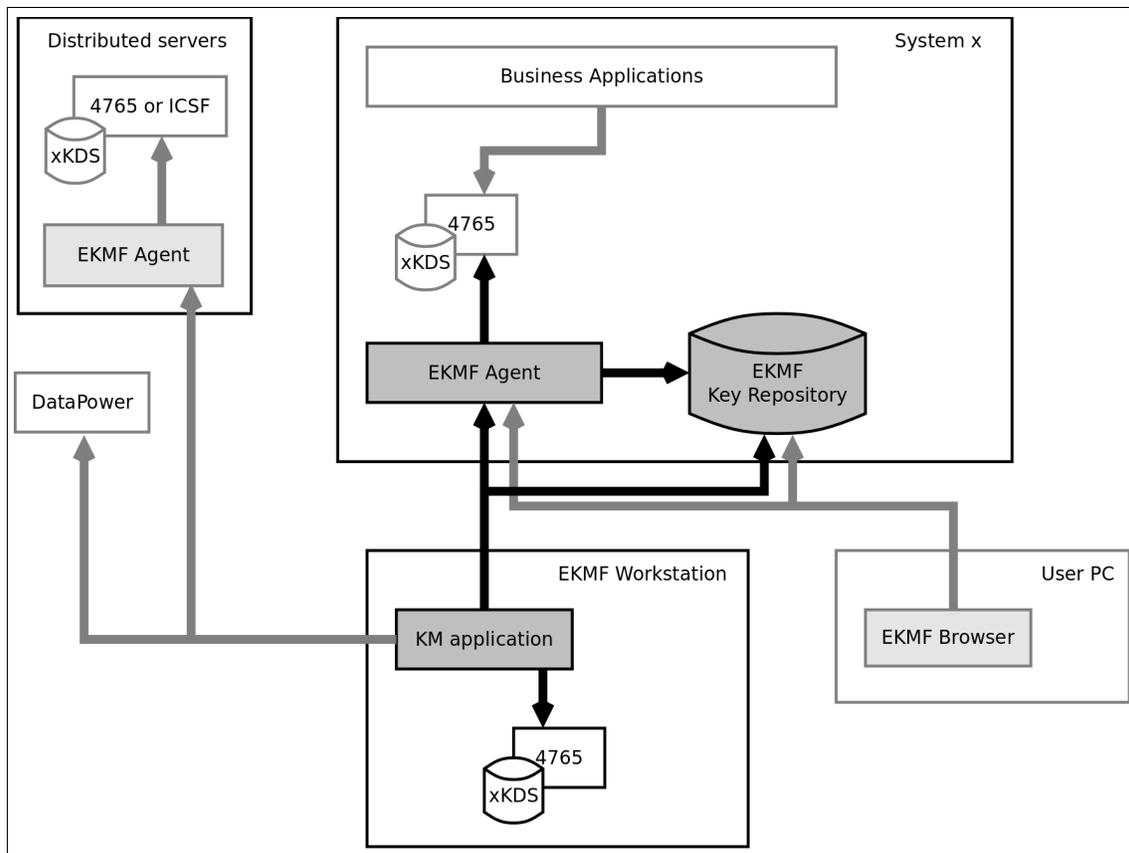


Figure 3-4 Online configuration with the Key Repository on a System x

Here are the properties of this configuration:

- ▶ Multiple Key Management Workstations can access the Key Repository.
- ▶ It supports the Browser.
- ▶ The Key Repository can be backed up as part of the normal DB2 backup procedures.

3.1.2 Environments

Most if not all IT installations have at least two separate environments: one for production and one for development and test. Some have additional environments for system test and quality assurance.

Separation of environments

Separation of environments is needed for several reasons. One reason is the ability to develop and test applications without affecting production systems. Another reason, which usually is as important, is the ability to separate test and production data, as production data is often critical and confidential and should not be exposed to developers and testers.

The need for encryption arises from a need to protect especially sensitive production data and systems. When encryption is employed, controlling access to cryptographic keys is essential, and cryptographic keys that are used for production should never be available on test systems and vice versa.

So, you should extend the separation of environments to include the entire key management system. You should at least separate the production environment from the other environments, meaning that you need at least two Key Management Workstations and two Key Repositories, one for the production environment and one for the other environments. Whether you need further separation depends on the circumstances of your installation.

Key Management Workstations that are used for the production environment should always be placed in a secured room. Key Management Workstations that are used for test and development do not need to be placed in a secured room, making access to those systems easier and more convenient. This separation of environments is illustrated in Figure 3-5 on page 75.

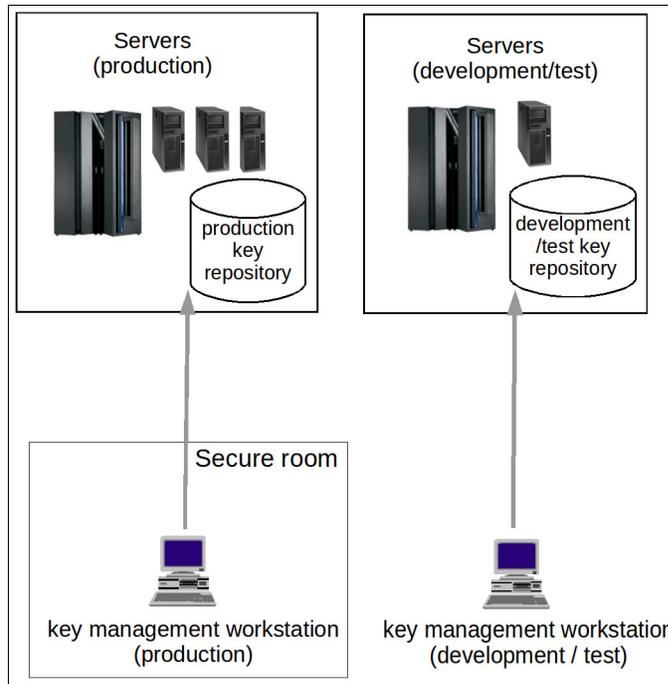


Figure 3-5 Separation of production and development environments

Using the same workstation for multiple environments

IBM Enterprise Key Management Foundation supports sharing of the Key Management Workstation, Key Repository, and cryptographic hardware in several environments. This is useful if you want to have several environments, such as test, development, and quality assurance, but have no need for watertight separation of keys and data.

Separate DKMS application environments on the same workstation can be individually configured. By applying the appropriate naming conventions that ensure keys for different environments have different names, it is possible to let environments share the Key Repository and cryptographic hardware.

3.1.3 Online Key Repository access

To access the IBM Enterprise Key Management Foundation Key Repository, you need an Agent and a JDBC database connection, regardless of the location of the Key Repository, whether it is installed locally on the Key Management Workstation or on a remote System z or System x server.

Figure 3-6 shows how the Key Repository is accessed when installed on either an System x or an System z server. The Key Management Workstation connects to the server through TCP/IP over a network connection. The connection through the Agent is used for reading and updating the Key Repository, and the JDBC database connection is used to read and search certain parts of the Key Repository.

If the server is equipped with optional cryptographic hardware (IBM PCIe 4765 Cryptographic Coprocessor on System x, ICSF on System z), as shown in t Figure 3-6, then you can enable hardware-based link encryption of the connection between the Key Management Workstation and the Agent.

The JDBC database connection can be protected through software-based SSL/TLS encryption.

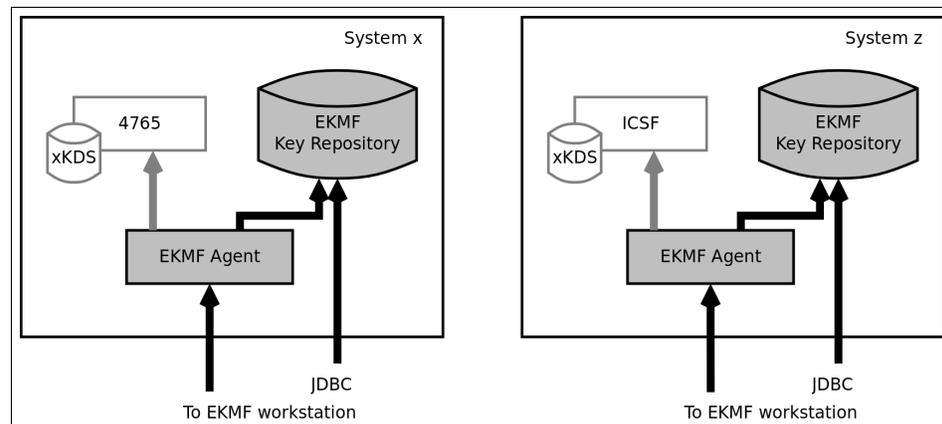


Figure 3-6 Online Key Repository access

3.1.4 Online keystore access

The Key Management Workstation can manage various keystores on remote servers through Internet Protocol network connections. IBM 4765 keystores, ICSF keystores, and RACF keystores all require an Agent to be installed on the remote servers to be managed. The Key Management Workstation connects to the Agent to manage the corresponding keystore. DataPower keystores can be managed directly without the need for an Agent. The Key Management Workstation connects directly to the DataPower device to manage its keystore.

Figure 3-7 on page 77 shows how the keystores are accessed.

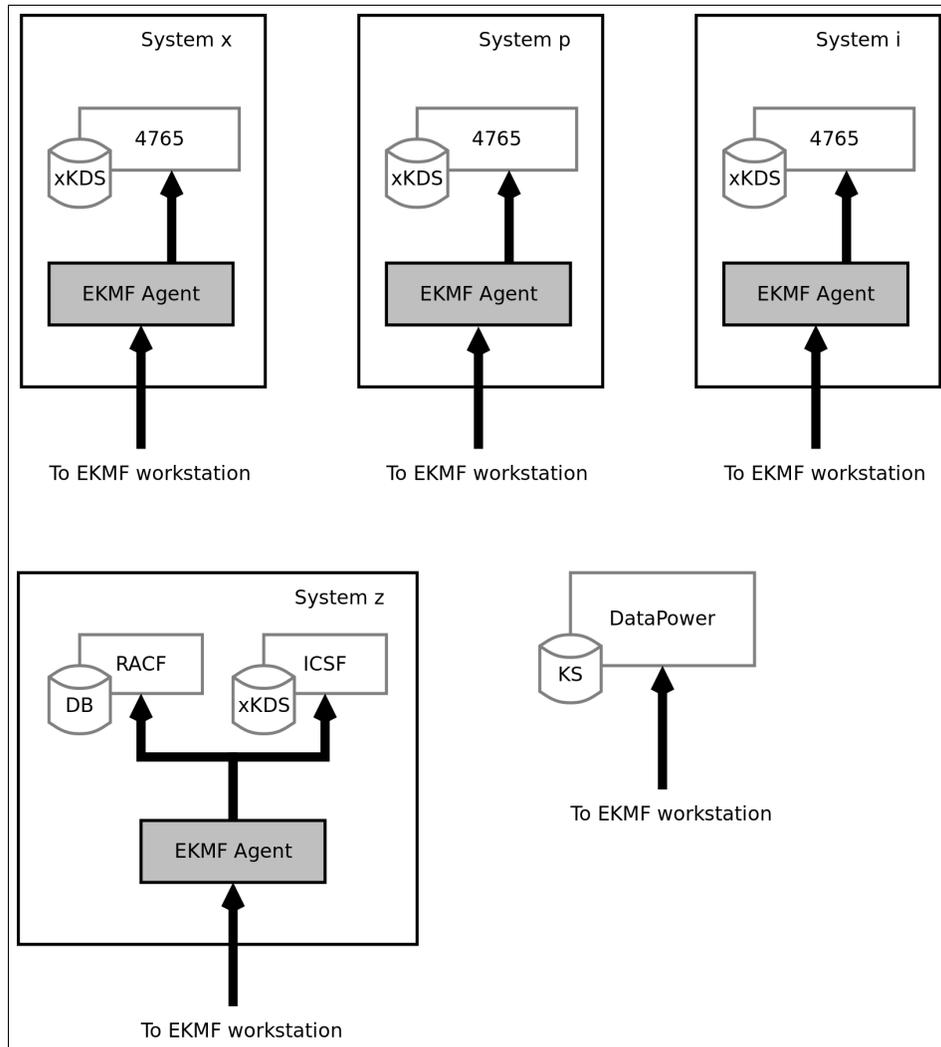


Figure 3-7 Online keystore access

From a key management perspective, the support for different platforms provides flexibility in the choice of platform where you run cryptographic applications. All major IBM hardware platforms (System z, System x, IBM System p, and IBM System i) are supported.

For details about the actual operating systems, see the IBM Enterprise Key Management Foundation documentation.

3.1.5 Designing the security organization

As described in 2.6, “Roles and responsibilities” on page 43, there are several options to reflect the security organization in the application’s and IBM 4765 access control systems.

It is important to consider how the organization wants to staff the key management activities. It should start with the employees that are available. Consider who can take on *administrative responsibilities*, who can perform as the *key manager*, and if you must have separate *key custodians* that are involved only when key parts are handled.

Minimum profiles: There must always be an ADMIN profile for the key management application to work. There should be at least one other profile for daily key management work, so that it can be demonstrated during audits that the ADMIN profile is occasionally used only for administrative purposes.

Using the role concept that is shown in “Sample role concept 3” on page 62, only four people are required. Often, two people are actively managing this setup, each having a designated backup person who is rarely involved. However, you must ensure that everyone that is involved is trained for their role.

3.2 Maintenance of the installation

Maintaining an IBM Enterprise Key Management Foundation solution involves applying updates and fixes from IBM when a change process requires the software to be updated. The software levels and patches that are applied to the workstation operating system, crypto driver, and crypto firmware are dictated by the level of the IBM Enterprise Key Management Foundation installation.

The maintenance that is covered in this section concerns keeping the IBM Enterprise Key Management Foundation running consistently and should not be confused with the key management work for which the installation is used.

3.2.1 Maintenance of the workstation

The Key Management Workstation generally is updated when a new level of IBM Enterprise Key Management Foundation is required for fixes or for new features. The Key Management Workstation functions as an appliance with no, or only tightly controlled, outside access. Therefore, it is not preferable to include the workstation in a regular server maintenance process, as this often involves unacceptable remote access.

Most fix releases, and many feature releases, require that only the IBM Enterprise Key Management Foundation installation itself is updated. The installer that is included in the installation media installs a new version separately from the existing version, allowing for quick rollback should any issues arise. The old version can then be removed later.

Some feature releases use new or updated features of the crypto hardware, necessitating that these features be updated as well. In total, a feature release might require updating all of the following workstation components:

- ▶ Operating system
- ▶ Crypto adapter driver
- ▶ Crypto adapter firmware
- ▶ IBM Enterprise Key Management Foundation installation

Additionally, there might be the need to update the Agent and Key Repository as well when they are outside the Key Management Workstation. Updating the remote components is mentioned in 3.2.2, “Maintenance of Agents and data tables” on page 80. In stand-alone configurations, where the Agent and Key Repository is on the Key Management Workstation, the IBM Enterprise Key Management Foundation installer performs the update of these components as well.

Updates to the operating system of a Key Management Workstation are most often done by applying a new service pack. The most common driver for this update is updated versions of the crypto adapter, where a new version might require updated packages to build its kernel driver. When such an update is required, it is tested by IBM and any custom steps in the process are described in the installation guide. The update process requires that you acquire the updated installation media from the operating system vendor and use them to apply the service pack updates to the Key Management Workstation. When required, this task must be done before updating the crypto adapter.

At times, the crypto adapter driver also must be updated to support new features that are available in the firmware. The driver is updated through the crypto software support installation program that is available on the IBM Enterprise Key Management Foundation installation media or as a downloadable component. A correctly applied update does not impact the configuration of users or masterkeys in the crypto adapter.

Likewise, the firmware of the crypto adapter sometimes is updated with features and fixes. The tools for maintaining the firmware are delivered with the crypto software support installation.

Updates for either the operating system, crypto adapter driver, or firmware are described in the IBM Enterprise Key Management Foundation release notes and installation guide.

To update or not to update: Even if a feature release contains an update to crypto adapter components, performing this update might not be required immediately. In particular, if the update relates only to new features, the update of the crypto adapter may be postponed while these features are not in use.

Always check the release notes and installation guide for details.

Except for the occasional operating system updates, the IBM Enterprise Key Management Foundation installation media generally contains the required software and documentation to perform a Key Management Workstation update.

3.2.2 Maintenance of Agents and data tables

New features, enhancements, or fixes might require updates to Agents and the Key Repository DB2 tables. These updates generally are kept to a minimum, in particular for System z installations, to not generate more change processes than are required. In general, the Agents must be updated only when some new features require the new functions that are available in the Agent. Likewise, the Key Repository tables change rarely and usually only when new features require it. In practice, the Key Management Workstation might go through several updates over a period without the Agents or the Key Repository requiring any changes.

For Agents and Key Repositories running on System x or System p, the updates are delivered as a new Agent code archive, which replaces the existing Agent, and as database command files, which update the Key Repository tables. Unlike the Key Management Workstation, these servers are not always dedicated key management servers, but may serve other functions as well, so the Agent is considered a normal running service and maintained as such. The updates to the Agent also include installation instructions.

On System z, the Agent package is delivered as binary TSO xmit-format files that are ready to transfer into data sets through the TSO receive command. The package contains load modules, DB2 DBRM members, definitions for the DB2 database, and sample JCL for setup and customization. The processes to install and run System z packages are tightly controlled by the existing processes for the System z installation.

3.3 Administering users

User administration for the IBM Enterprise Key Management Foundation is required when there are changes in the security organization over time.

New users can be added as explained in 2.6.2, “Access control systems” on page 46. This process includes creating user accounts for SLES, assigning a profile in the IBM 4765, and associating the new user with one or more IBM 4765 groups. When adding new users, you must verify whether they were an EKMF user before. In this case, you must ensure that the user is not granted access to critical resources, as described in the following list.

When an employee leaves the job in the security organization, you must ensure that all access rights are removed or revoked. This task includes the following actions:

- ▶ Revoke access to the Key Management Workstation facilities.
- ▶ Revoke TSO access for accessing the Agents.
- ▶ Remove a user ID on the Key Management Workstation operating system.
- ▶ Change a profile in the IBM 4765 in the Key Management Workstation.
- ▶ Revoke access to key material, such as key parts in safes.
- ▶ Remove a personal user in the key management application (if the setup is not based on groups only).
- ▶ Revoke a user certificate for Browser access.

Keep critical information: To ensure that split knowledge is in place at all times, it must be registered which users have access to which key parts. If an employee leaves the security organization, the information must be kept, to ensure that the employee is not getting access to the other half of a key in case he returns to the security organization.

3.4 Providing applicable logging

The purpose of audit logging in the IBM Enterprise Key Management Foundation is to prove what actions were performed, but also to prove what actions were not performed. To achieve this goal, the logging mechanism requires special permissions to add entries into the audit log, and it must not be possible to insert fake entries in the log.

IBM Enterprise Key Management Foundation can write audit log entries in to the SMF logs on z/OS, and in to the DB2 database on the mainframe that is used by the Key Management application itself to store further configuration information. For each DB2 log entry, a MAC can be calculated through a key that is available only on the Key Management Workstation. The fact that SMF is used makes it difficult to remove log entries, and because each log entry is MACed, it is hard to insert fake entries.

In configurations where System z is not used, the Agent can write log entries to a text file. Access to this file should be protected by the operating system.

The PROG0360 Audit control program within the Key Management Workstation application allows you to search log entries within the DB2 database and change the log configuration.

This audit log contains information about all events that potentially changed the system. Each log record is limited in size to fit the SMF records. This can make it difficult sometimes to find specific log entries, and understand their content.

Therefore, another event log exists within the Key Management Workstation, which does not use SMF. This log contains a detailed history of all the actions run with the keys, and for each log record a MAC is calculated. This log is available through the PROG0323 Symmetric Key Management program within the Key Management Application.

For daily operations on the Key Management Workstation, the login PROG0323 Symmetric Key Management can help you answer questions such as: Why is key X inactive? Who entered key Y into the system?

The audit log stored in DB2 and accessible through the PROG0360 Audit control program typically is used by an auditor to answer questions such as: Who logged on to the system in the last year? Has the configuration been changed from outside the Key Management Workstation?

The SMF log can be additionally investigated, if fraud is suspected. The log contains entries from other applications running on System z as well, so you might need to filter the log before relevant information can be found.

3.5 Tracing for troubleshooting

If a problem occurs on the Key Management Workstation, the application shows a dialog box that contains a short description of the problem. If you cannot solve the problem based on this information, you might have to contact IBM. Usually, such problems occur only during the installation of IBM Enterprise Key Management Foundation, or when the configuration is changed. It is preferable to first apply the changes to a test system similar to the production system, so that problems can be solved before the production system is affected.

To solve the problem fast, it is important to describe the problem in detail. Screen captures of the entire problem scenario can be helpful. If a software bug is suspected, you must produce trace data from the affected applications so that the developers can map the problem to the source code. The trace files might be of too little or no use at all for the standard user, but helpful for IBM.

The trace files contain information about the state of the application and information about the data passing through the application. Sensitive data such as passwords and clear key values are not included in the trace.

Usually, trace files from the Key Management Workstation are sufficient to find a problem, but in some cases a trace from the Agent or other components also are required. Table 3-1 provides an overview of the different trace sources and when to use them.

Table 3-1 Traces from IBM Enterprise Key Management Foundation

| Application | When to use trace | Comment |
|----------------------------|--|--|
| Key Management Workstation | When problems occur within the Key Management Workstation. | The Key Management Workstation consists of a number of subcomponents. Traces from all components are placed in the trace directory for the active environment: <code>/var/opt/dkms/<environment>/trace</code> To start a new trace, clear the content of this directory, start IBM Enterprise Key Management Foundation again and select the Enable Trace check box in the splash window after the application starts. Send the entire content of the directory to IBM, as part of the problem description. |

| Application | When to use trace | Comment |
|------------------------------|---|---|
| Agent | When problems occur within the Key Management Workstation, where KMG or 69 is returned as the System ID. | The System z Agent writes to SYSOUT when trace is enabled. To enable trace, use the &TRACE-DATAIN, &TRACE-DATAOUT, and &DEBUG parameters in the option data set. Set each parameter to YES. The Agent for other platforms writes the debug information to standard out. To enable trace, use the debugLevel parameter in the cserver.properties file. Set it to 2 to enable full trace. |
| Crypto Node Management (CNM) | Problems in CNM usually are caused by a crypto card that is not set up correctly. The return codes usually are enough to pin point the problem. The trace can give a more detailed view, if the return codes are not enough to solve the problem. | CNM writes trace information to standard out. To enable the trace, run the following command: /opt/IBM/4765/cnm/csulcnm /D |
| csulclu | If it is not possible to obtain the status of the crypto card from the application. | The application always writes two files: The log file that is specified by the user, and one ending with .mr1. |

3.5.1 Other tools for troubleshooting

The Key Management Workstation relies on the IBM 4765 for all crypto operations. Many common problems can be traced back to a wrong setup of the IBM 4765. A utility program that can export the current setup of the IBM 4765 to a text file is installed during the IBM 4765 installation if the smart card option is selected for installation.

To use the utility, run a command similar to the following one:

```
/opt/IBM/4765/cnm/cca_hsm_util.e -F 4765config.txt
```

The configuration is then written to the 4765config.txt file.

The Key Management Workstation communicates with Agents and the DB2 database (for a remote installation with DB2 on the mainframe) over the network. If there are problems that are related to network connections, some standard tools that are provided by the Linux based operating system that is installed on the workstation can be useful, as shown in Table 3-2.

Table 3-2 Network troubleshooting

| Application | When to use | Example |
|-------------------|--|---|
| traceroute | If a connection cannot be established, traceroute can display how far the communication gets. This can be useful if communication is blocked by a firewall. | traceroute db2host.company.com |
| nslookup | If names are used rather than IP addresses to identify servers in the Key Management Workstation, you can check whether the name is resolved correctly in DNS. | nslookup mvsf.prv.dk.ibm.com |
| telnet | To tell if an Agent runs on a specific port. | telnet db2host.company.com 50001 <ul style="list-style-type: none"> ▶ If nothing is running on port 50001, telnet returns connection refused. ▶ If a firewall is blocking the connection, it times out. ▶ If an Agent is running, the connection is accepted, and closed when Enter is pressed. |
| ping | General test of the network on the Key Management Workstation. Ping a host that is known to be reachable to verify that the network interface is functional. | ping db2host.company.com |

3.6 Ensuring consistent backup and restore procedures

The IBM Enterprise Key Management Foundation Backup and Restore utility, which is described in 2.4.1, “Key Management Workstation” on page 33, is an additional application that can be used on the Key Management Workstation.

The utility must be installed separately from the IBM Enterprise Key Management Foundation installation media. Both the backup and restore functions are available in the same installation package.

The utility is provided in the form of a command-line program with two executable files: one for backup and one for restore.

A feature of the backup portion is that it can run a full backup without user input, which allows for the backup to be set up as an automated task, for example, to run every night. Alternatively, the backup can be run manually as a final step in a change process, thus storing any changes that are made to the workstation configuration. When running automatically, the return code of the backup process can be checked to see whether any error occurred during the backup.

Although the backup process runs without interaction, the restore utility allows you to choose between restoring everything in the archive, or only a specific item. For example, if the crypto adapter is replaced in a workstation, you can opt to restore only the adapter contents, or if multiple IBM Enterprise Key Management Foundation configuration instances are present in the archive, you can choose to restore only one of them.

The backup process produces three files: the archive file, a log file, and a checksum file. The archive file with a `.tar.gz` designation contains the actual backup data, the log file contains the session log, and the checksum file contains an SHA256 checksum of the backup file. During the restore process, the checksum of the backup is compared to the checksum value of the file to detect any modifications or errors in the backup archive. The checksum file is not required to perform a restore, but the utility shows a warning if the checksum file is missing or if the checksum values do not match. Keeping a separate secure copy of this checksum file allows it to be used as a tamper detection mechanism to protect against modifications of the backup archive. Should the backup archive be modified or corrupted, the change is detected when the archive is brought together with the hash for a restore operation.

The backup media must be moved to safe storage. Because the IBM Enterprise Key Management Foundation does not allow incoming connections, the options are to either push the backup over SSH or, if the backup is done manually, move it through USB media.

If there are problems with the backup or restore process, the utility shows error information in window and adds additional information to the log file. Through configuration settings, the amount of logging can be increased if required.

3.7 Conclusion

Planning the deployment and maintenance of a IBM Enterprise Key Management Foundation installation is an essential step to setting up a solid foundation on which to add all the features and solutions that are used to fulfill the business requirements. Regular maintenance, such as applying updates, provisioning new users, and preparing for disaster recovery, helps ensure that the installation does not degrade or increase risk, but continues to provide a full value and compliance.

This chapter described the various options for deployment and the tools and processes of maintaining a secure and efficient installation.



Part 2

Use case scenario

This part describes a use case scenario for a typical financial company. The company issues payment cards and operates a network of ATMs and payment terminals and has a demand for functioning key management processes.

This part includes a description of the bank's requirements and the need to support the following five areas:

- ▶ Establishing the basic key management infrastructure
- ▶ ATM key management
- ▶ EMV card issuance
- ▶ Workflow and reports
- ▶ Certificate management

This book covers the first of these five requirements.



Overview of scenario, requirements, and approach

This chapter introduces a typical business scenario of a fictional financial company, referred to as *The Fictional Bank (FB)* or *the bank*. It shows how the bank can use the IBM Enterprise Key Management Foundation to create a powerful and flexible key management system that allows the centralized creation and management of all encryption keys.

This chapter includes the following sections:

- ▶ Company overview
- ▶ Business requirements
- ▶ Functional requirements
- ▶ Architectural decisions
- ▶ Solution overview

4.1 Company overview

The Fictional Bank is a large retail bank serving ten million customers with Internet banking, debit, and credit cards. The bank has an established network of branch offices and runs 2000 ATMs across the country. The bank also operates a network of Point-of-Sales (POS) terminals.

4.1.1 Current IT infrastructure

The IT infrastructure of The Fictional Bank is based on a mixture of IBM zEnterprise Systems running z/OS and Intel based servers for the web front end.

For issuing cards, the bank is running an issuing system in one z/OS LPAR. In this LPAR, card requests are received and all preparation for card data takes place here. For the cryptographic operations, The Fictional Bank is using Integrated Cryptographic Services Facility (ICSF) and Crypto Express.¹ The outcome of the process is a file with card personalization data that eventually is sent to the card bureau that produces the cards and loads personalization data onto them. Currently, The Fictional Bank is in the process of introducing chips on their debit and credit cards, the so-called Europay MasterCard, Visa (*EMV*) cards.

The card holders that have these new EMV cards use them at terminals, either in ATMs or at POS terminals. These transactions flow into the bank's authorization system running in two z/OS LPARs that are set up as a Parallel Sysplex. In the authorization system, the bank is using ICSF and Crypto Express for the cryptographic operations. To manage the Crypto Express, the bank uses Trusted Key Entry (TKE). For more information about TKE, see *System z Crypto and TKE Update*, SG24-7848.

The issuing and authorization system is set up with different master keys in the Crypto Express cards, which is a preferred practice that helps separate the issuing of cards from authorizing transactions. Given the master key separation between the issuing and the authorization systems, the two systems have separate cryptographic key data sets (CKDS) for storing the symmetrical keys and also separate public key data sets (PKDS) for storing the asymmetrical keys. As the authorization system is running in sysplex mode in two LPARs, the CKDS and PKDS keystores for this system are shared between the two LPARs.

Internet banking is provisioned through many Intel based servers, which access data and services through middleware on the two z/OS LPARs, which also host the authorization system.

¹ The term *Crypto Express* is used to denote Crypto Express3 or Crypto Express4S features.

The bank is connected to the Visa and MasterCard networks and can route transactions that are generated at its terminals by foreign bank cards to these network carriers (transactions that are *at us* but not *on us*), and receive transactions that are generated at foreign bank terminals by own cards (transactions that are *on us* but not *at us*) from the network carriers.

4.1.2 Key management issues in the current infrastructure

The Fictional Bank has a number of challenges in the existing key management setup. Most of these challenges are related to manual processes and compliance issues.

ATM network

For the ATM network, the bank traditionally managed the terminal master keys (TMK) by having them on paper in two components. The components originally were generated through ICSF panels and by creating random data. Then, the keys were written down. Recently, this process was improved by using a dedicated IBM 4765 in a designated server.

The keys are loaded in to the terminals by sending out two people, each holding a key part to be entered. This manual process is time consuming. The bank also realized that the personnel frequently do not focus on the proper handling of the key material and on continuously documenting their activities. As such, the bank is facing compliance issues during audits.

Issuing and authorization system and payments network

Considering that the card data is prepared in the issuing system, and later the transactions are processed in the authorization system, the two systems must share a number of keys. For example, a key that is used on the issuing system to generate a card verification value (CVV) requires an instance of the same key on the authorization system to verify the CVV. With ICSF and Crypto Express, these two keys have the same value but different attributes for security reasons.

The bank creates the keys at the issuer system through the Key Generation Utility Program (KGUP) and a number of home-made utilities. To exchange keys with the authorization system, the bank enciphers the keys with a previously exchanged Key-Encrypting Key (KEK). The keys are then moved in files in a manual process, which is not only time-consuming but also error prone. Furthermore, the process does not have built-in audit logging mechanisms, and the bank does not have a central overview of all keys, their lifecycle, and status.

The connection to the payment network requires that keys are exchanged, such that PIN-based transactions can be routed to or from the network for verification. Initially, the bank exchanged a zone master key (ZMK) in components with each network. At the bank, the ZMK was loaded into the authorization system through Trusted Key Entry. Using the ZMK, session keys are generated by the application and exchanged in the network daily.

Internet banking

The Internet banking servers use cryptography primarily for the generation and handling of private keys and certificates for SSL/TLS. The bank acquires all certificates for both Internet facing servers and internal servers from an external certificate authority (CA).

4.2 Business requirements

This section considers the encryption and key management-related requirements of The Fictional Bank. The bank has two primary business requirements:

- ▶ Compliance
- ▶ Cost-effective key management operations

4.2.1 Compliance

The Fictional Bank must adhere to PCI-PIN and PCI-DSS requirements. The bank suffered consequences from many findings in recent audits and the bank's management now has a strong focus on compliance.

4.2.2 Cost-effective key management operations

The bank's management identify two areas that they consider ineffective from a cost perspective:

- ▶ The synchronization of keys between the issuing and authorization system is time-consuming. Errors in this process occurred and led to downtimes in the authorization system.
- ▶ The time that is spent in audits is high because of a lack of an overview of the existing keys and processes, and because of insufficient and dispersed audit logs.

4.3 Functional requirements

The bank has broken their business requirements into a set of functional requirements, which can be summarized in these three bullets:

- ▶ Centralized operations, which consider organizational aspects
- ▶ Basic key management requirements, which focus on security requirements and basic key distribution
- ▶ Extended key management requirements, which consider new business areas, such as EMV and additional key management functionality

Let us examine each of these requirements.

4.3.1 Centralized operations

The bank wants to centralize key management operations because the current decentralized process is causing compliance issues and is error prone.

Here are the bank's requirements:

- ▶ Key management operations must be placed at one entity in the organization.
- ▶ All keys must be managed from that entity.
- ▶ A common set of procedures must be in place.
- ▶ Audit logging of all critical key management operations must take place in a central log.
- ▶ The system that is brought in place must be extendable, such that enterprise key management can be achieved.

4.3.2 Basic key management requirements

The bank went through an analysis and they identified many requirements for the central key management application. The initial usage of the system is for payment cards.

- ▶ The key management application must use a FIPS140-2 level 3 or FIPS140-2 level 4 certified HSM.
- ▶ The key management application must be configurable to require two-factor authentication.
- ▶ The key management application must be configurable to require two users for certain critical functions.

- ▶ The key management must enforce split knowledge for clear key part (key component) entry.
- ▶ It must be possible to print clear key parts (key components) on key mailers through an attached printer.
- ▶ The key management application must provide separation between keys for the following aspects:
 - Key separation for different purposes, such as key-encrypting keys, MAC keys, data encryption keys, and PIN encryption keys.
 - Key separation for different entities or third parties, for example, through separate key hierarchies.
- ▶ All static keys (keys with a long life-time) must be generated centrally.
- ▶ It must be possible to push the static keys over the network to the issuing and authorization systems.

4.3.3 Extended key management requirements

Beyond the basic requirements, the bank requires that the usage of the system be extended to cover all their key management needs, including the following items:

- ▶ ATM key management

The Fictional Bank wants to replace the manual key distribution of ATM terminal master keys (TMK) with a remote key loading scheme, which is described in ANSI X9.24 part II.
- ▶ EMV card issuing

The bank is planning to start issuing EMV cards in the near future. To do so, the bank must manage issuer master keys and certificates, generate RSA private key pairs for every EMV card, and issue card-specific certificates.
- ▶ Workflows and reporting

The new key management processes should be workflow-based, reducing the time that is spent for planning and running the operations and also provide additional monitoring and reporting facilities for the keys and certificates that are close to expiration.

- ▶ Certificate management

Managing certificates is a difficult task for the bank. The usage of certificates, typically for SSL/TLS connections, has grown over the years across the application areas. The bank wants a central overview of all certificates and the capability to issue certificates for internal use. Additionally, for Internet-facing applications, the bank must be able to acquire and manage the necessary certificates.

4.4 Architectural decisions

The bank evaluated IBM Enterprise Key Management Foundation and found that it appropriately addresses the identified functional and non-functional requirements. After careful analysis, the bank made and documented a number of architectural decisions that are based on the requirements:

- ▶ Decision: IBM Enterprise Key Management Foundation will be used as the strategic key management system.

The IBM Enterprise Key Management Foundation Key Management Workstation is where all key management operations, including key generation and distribution, are carried out. The exception is the loading of ICSF master keys, which is performed through the TKE workstation. The TKE workstation is placed in a secure room, and the production Key Management Workstation can be placed next to it.

The Key Management Workstation uses an IBM 4765 FIPS 140-2 level 4 certified HSM for all relevant cryptographic operations.

The Key Management Workstation uses a complex access control system that is provided by the IBM 4765, based on hardware enforced two-factor authentication and granular access control rights to cryptographic functions. The IBM 4765 allows you to permit or restrict access to certain critical operations and to enforce split knowledge during entry and print of clear key parts to specific users, user groups, and combined with the Key Management Application, even user group-of-groups.

The Key Management Workstation supports printing clear key parts on a locally attached printer.

The system is based on the IBM Common Cryptographic Architecture (CCA), which allows separation of keys for different purposes. It is also possible to build key hierarchies to separate keys for different entities and third parties.

IBM Enterprise Key Management Foundation is scalable, allowing for the management many keys and keystores on many servers, and thus can meet the increasing demands as enterprise key management is being rolled out within the organization. Also, optional features supporting remote key loading of ATM terminal master keys, EMV card issuing, workflow, and general certificate management are available.

- ▶ Decision: IBM Enterprise Key Management Foundation will be implemented in online configuration with the key repository in the z/OS LPAR hosting the issuing system.

The online configuration allows pushing keys over the network to the issuing and authorization systems. The decision to place the key repository on z/OS is based on current and anticipated requirements:

- The key repository can be covered by the DB2 backup procedures in place.
- It is a prerequisite for many IBM Enterprise Key Management Foundation features, including the Reporter, the EMV card issuing API, and remote key loading API, which are features that the bank plans to use in the future because these features are available on z/OS only.

Because the issuing system will use the EMV card issuing API, the key repository is placed in the LPAR hosting the issuing system. DB2 sharing will be used to share the key repository with the LPARs hosting the authorization system, should the need arise.

- ▶ Decision: There will be separate ICSF keystores for the issuing and authorization systems.

The issuing system must be isolated cryptographically from the authorization system. Therefore, the bank has two ICSF CKDS keystores, one for the issuing system and one for the authorization system, which are protected by different ICSF master keys. Sysplex sharing is used to share the CKDS between the two LPARs hosting the authorization system. This setup ensures that updates made to the CKDS in one LPAR are picked up by ICSF in the other LPAR. The bank plans to continue using this configuration.

- ▶ Decision: The audit log will be written in both the key repository and SMF.

The Key Management Workstation will be configured to generate an audit trail and write it to the audit log in the key repository and the SMF log. The SMF log is used for other systems and is considered difficult to tamper with.

- ▶ Decision: Three Agents will be deployed on z/OS, one in each LPAR.

One Agent (a crypto and DB2 Agent) is deployed in the LPAR hosting the issuing system. The Key Management Workstation connects to this Agent to manage the ICSF CKDS that is used by the issuing system to manage the key repository, and to write the audit log in SMF.

One Agent (a crypto Agent) is deployed in each of the LPARs hosting the authorization system. The Key Management Workstation connects to one of these two Agents to manage the ICSF CKDS used by the authorization system. The sysplex distributor routes the connection from the workstation to one of the available Agents, such that if one LPAR is not up, the connection is routed to the Agent in the other LPAR. Sysplex sharing ensures that all CKDS updates made from one LPAR are seen by ICSF in both LPARs.

4.5 Solution overview

This section aims to present a detailed overview of the solution that is selected for implementation at The Fictional Bank. The section covers the prerequisites that are required to prepare the solution and the plan for the implementation tasks, which are split up in to individual phases.

4.5.1 The design for the IT infrastructure and processes

The Fictional Bank in cooperation with IBM settled on an infrastructure solution design to meet the current key management objectives and to set the foundation for adding additional application support to the solution.

Prerequisites

The Fictional Bank has in its infrastructure many components into which the IBM Enterprise Key Management Foundation can be integrated. To prepare for the implementation phases, many items were identified that must be completed before the implementation can begin. The preparation covers acquiring the relevant components, preparing and scheduling change requests to the mainframe and the network, and scheduling education for the security officers.

Purchasing the required hardware and software components

The Fictional Bank must purchase the hardware and software that is required to run three Key Management Workstations: one for production, one for disaster recovery, and one for combined test and quality assurance. To cover these needs, Fictional Bank places an order for the following items:

- ▶ Three System x servers with IBM 4765 crypto cards for use as Key Management Workstations.
- ▶ Six smart card readers, two for each workstation. As Fictional Bank already has a Trusted Key Entry workstation, they have the required smart cards.
- ▶ Three SLES 11 32-bit licenses for the workstations.

- ▶ Three IBM DB2 Connect™ licenses to enable the key repository connections.
- ▶ An IBM Enterprise Key Management Foundation license covering all installations.

After coordinating the purchases with IBM, the orders are made in time for the components to be delivered before implementation is scheduled to begin.

Planning change requests

The solution requires that new software is installed in the mainframe environment at Fictional Bank, that new tables are created for the key repository, and that the internal network configuration is updated.

Changes to production systems at Fictional Bank are tightly controlled by internal change request processes and can take place only during planned maintenance windows. To perform the setup of the IBM Enterprise Key Management Foundation mainframe components with due diligence, the bank plans to have everything ready and in place for a suitable maintenance window.

Scheduling education

The admins / superusers need introductory education to train them in the usage of the IBM Enterprise Key Management Foundation to take full advantage of the solution. For this purpose, an instructor-led workshop is planned that will provide classes and hands-on training for the security officers.

The admins / superusers use the taught skills and work experience training to develop processes and procedures that will function alongside the existing workflow at Fictional Bank.

Deployment architecture

As mentioned, Fictional Bank has a production environment, a disaster recovery environment, and a combined test and quality assurance environment. This section focuses on the production environment, although the test and quality assurance environment is where the deployment will start.

Current production environment

Figure 4-1 on page 101 shows Fictional Bank's current production environment from a cryptographic application perspective.

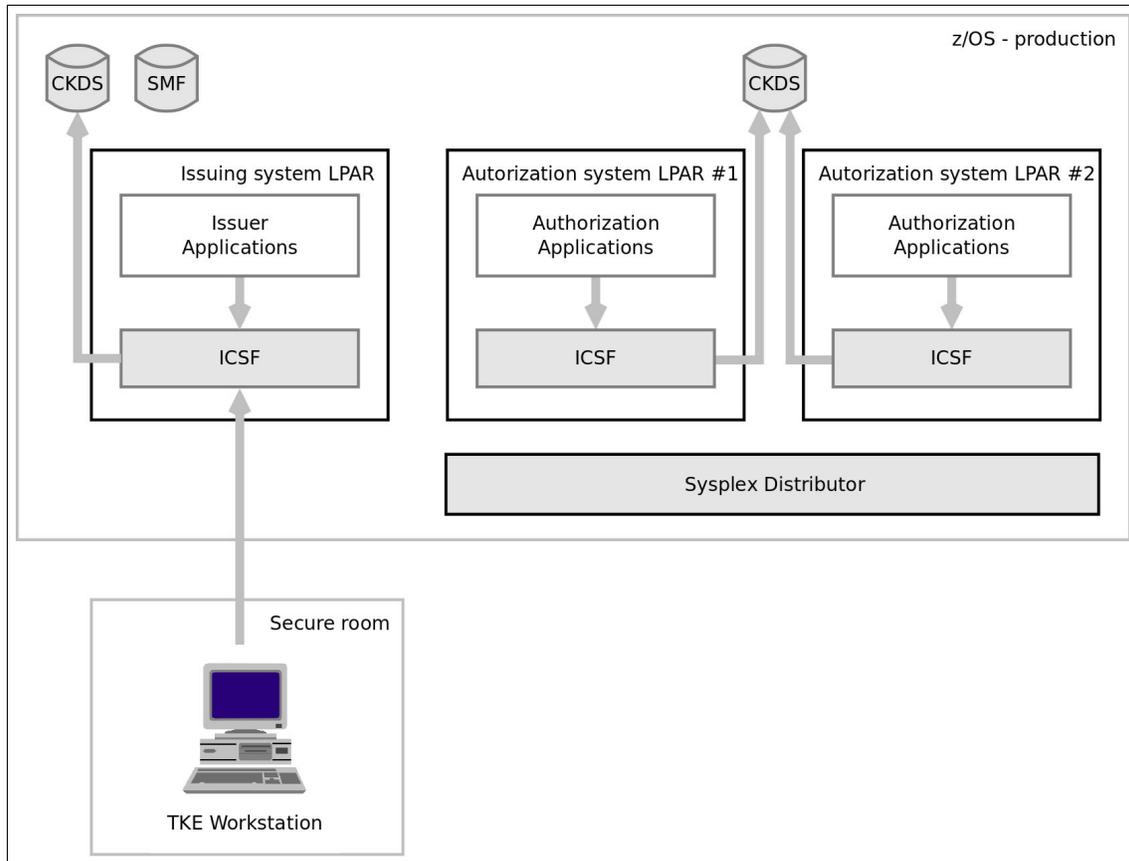


Figure 4-1 Current environment - production

The issuing system has a dedicated z/OS LPAR hosting the issuing applications. The issuing applications use ICSF for cryptographic services, and cryptographic keys are kept in a CKDS that is dedicated to the issuing system.

The authorization system has two z/OS LPARs hosting the authorization applications and uses Sysplex Distributor for load balancing between the two LPARs. The authorization applications also use ICSF for cryptographic services, and cryptographic keys are kept in a CKDS that is shared by the two LPARs through sysplex sharing.

The Trusted Key Entry workstation, placed in the secure room, connects to ICSF in the LPAR hosting the issuing system. Because all LPARs are running in the same mainframe, and thus share cryptographic hardware, the workstation can manage ICSF master keys for all LPARs using that connection.

Planned production environment

Figure 4-2 shows the planned deployment of IBM Enterprise Key Management Foundation in the production system.

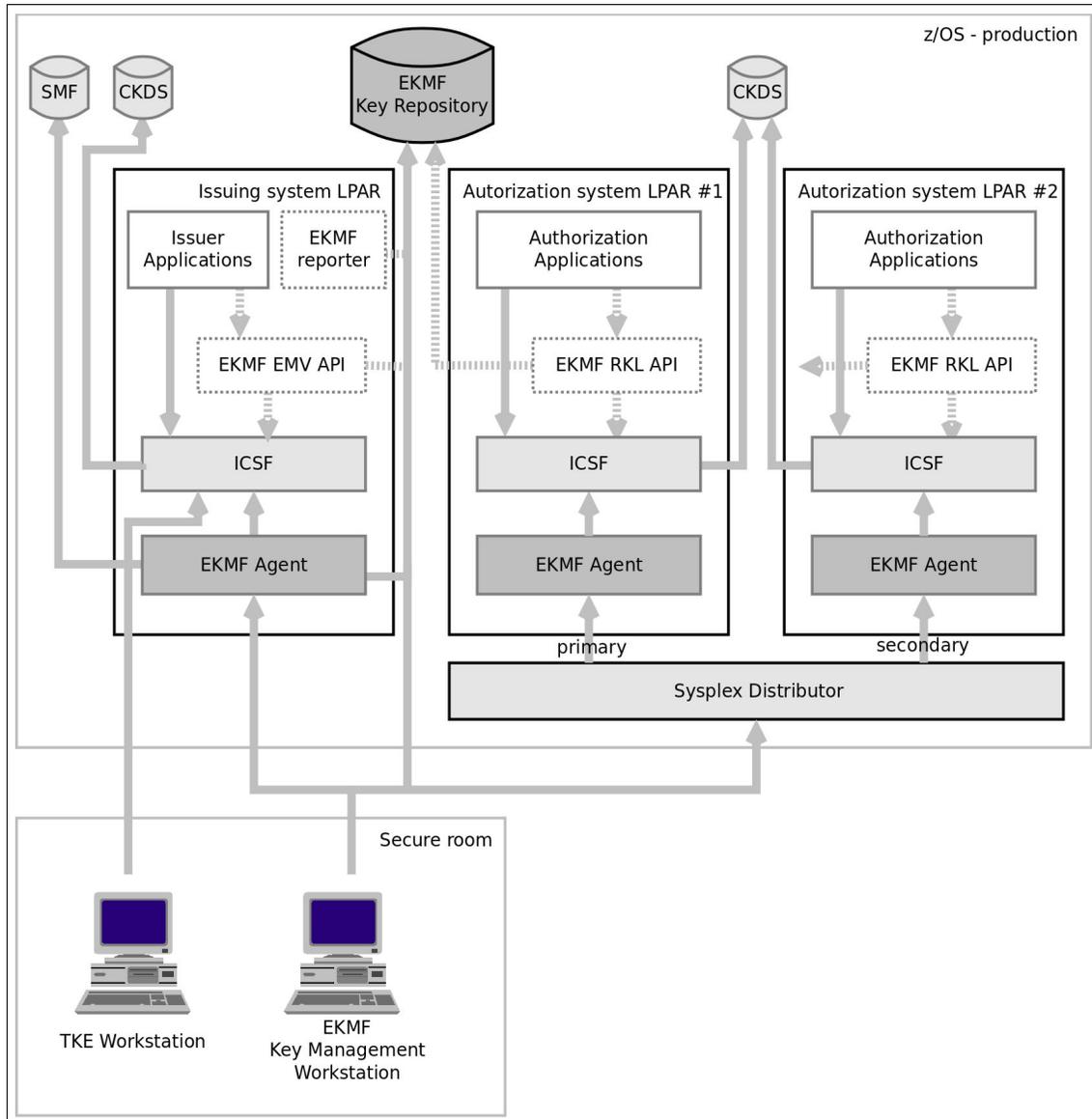


Figure 4-2 Planned deployment - production

The deployment uses a phased approach. In phase one, IBM Enterprise Key Management Foundation is introduced for key management purposes, and later phases introduce the Reporter and EMV and Remote Key Loading support.

Deployment phase one

In phase one, IBM Enterprise Key Management Foundation is introduced for key management purposes. For the production environment, this implies the following things:

- ▶ The key repository DB2 database is created.
- ▶ An Agent is installed in the issuing system LPAR and configured for managing the issuer system CKDS, the key repository, and for writing audit records to the SMF log.
- ▶ An Agent is installed in each of the authorization system LPARs, with both Agents configured for managing the authorization system CKDS. The Agent in authorization system LPAR#1 is the primary Agent for the authorization system, meaning it is the one that is used to manage the CKDS whenever LPAR#1 is active. The Agent in LPAR#2 is used as a backup. Sysplex Distributor is configured to route key management requests to the proper Agent depending on the availability of the two LPARs.
- ▶ The Key Management Workstation is installed in the secure room. It is configured to connect to the Agent in the issuing system LPAR for managing the issuing system CKDS, the key repository, and writing audit records to the SMF log, and to connect to one of the Agents (through Sysplex Distributor) in the authorization system LPARs for managing the authorization system CKDS.

Subsequent deployment phases

According to Fictional Bank's plans, later phases introduce the following items:

- ▶ ATM key management: Remote key loading for ATM terminals is handled by applications in the authorization system through the IBM Enterprise Key Management Foundation remote key loading API. The API accesses the key repository through DB2 sysplex sharing.
- ▶ EMV card issuing: Issuing of EMV cards is handled by applications in the issuing system through the IBM Enterprise Key Management Foundation EMV card-issuing API. Private keys for the EMV cards are generated by using the IBM Enterprise Key Management Foundation RSA Key Generation z/OS feature.
- ▶ Workflows and reporting: Workflows can be implemented without the need for additional components. The Reporter is deployed in the issuing system LPAR.
- ▶ Certificate management: Certificate management can be implemented without the need for additional components.

Deployment across environments

The test system, which is where deployment starts, is configured like the production system, except that it has two LPARs only (as there is no need for duplication of the authorization system), and the Key Management Workstation is placed in a normal office.

The disaster recovery system, which is situated in a different geographic location, is a mirror of the production system, and is configured the same as the production system.

Roles and responsibilities

Fictional Bank has eight trusted employees performing key management duties. However, even though the bank has confidence in their employees, industry preferred practices and compliance regulations still require that separation-of-duties are employed. The bank decides, based on recommendations from IBM, on a hierarchy of split responsibilities that enforces dual control on sensitive operations.

Figure 4-3 shows an overview of the chosen hierarchy.

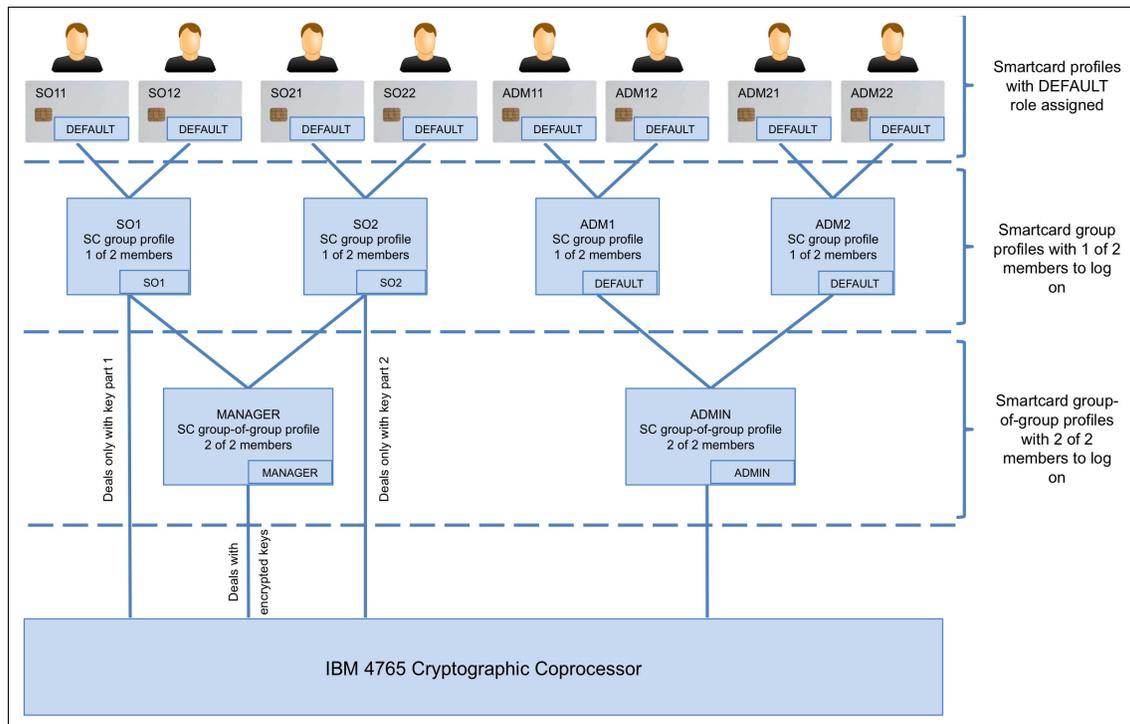


Figure 4-3 Role and profile hierarchy at Fictional Bank

The eight security officers each use the personal smart card that they possess for use with the Trusted Key Entry workstation. Using this smart card, each person is assigned a profile on the Key Management Workstation.

The setup is built on applying the concept of group-of-groups, which is described in 2.6, “Roles and responsibilities” on page 43. In this setup, the privileged role of *Manager* or *ADMIN* can be accessed only when two entities are logged on. The only entities that can log on to the privileged roles are also groups themselves, but these groups require only one entity to log on. Finally, at the top of the hierarchy, there are the individual security officers who log on using their personal smart cards, which are then assigned the rights of the group they are logging in to. When one person from each group is authenticated, the IBM 4765 accepts a logon to the privileged role.

Two users are included in each of the SO1, SO2, ADM1, and ADM2 groups to ensure that any process can be carried out even in the absence of a security officer.

The following roles are defined at Fictional Bank:

| | |
|--------------------|--|
| Default | The Default role is assigned to the individual personal smart card profiles and to the groups of ADM1 and ADM2. This role is restricted from performing any sensitive operations on the IBM 4765, which ensures that no single individual can log on alone and perform unsupervised key management. |
| SO1 and SO2 | Exceptions to the requirement of dual logon are the SO1 and SO2 roles. These roles have permissions that allow them to log on and enter key parts in clear. Because entering a clear key part must be done in only secret by one person, these roles are enabled for only a single user. Splitting the clear entry between two roles ensures that no single person can enter all the key parts of a single key. |
| MANAGER | The MANAGER group is the group that is used to perform most of the day-to-day work of creating and maintaining keys and key templates. This group of people is responsible for handling requests for new keys, renewing expiring keys, and otherwise maintaining the key repository. Additionally, this group supports the development of business applications by designing and creating key templates to enable provisioning any keys that are required in a secure and compliant setup. |

ADMIN

The ADMIN group is responsible for maintaining the key management application configuration settings, adding and maintaining devices in the device configuration and administering users and their access profiles. Also, this group has privileges for maintaining the IBM 4765 adapter. Members of this group do not work with the Key Management Workstation daily, as this sort of configuration is required only intermittently.

The setup by Fictional Bank is designed to fulfill both internal and regulatory requirements for separation of duties while still maintaining availability.

4.5.2 The plan for implementation phases

Fictional Bank decided to implement IBM Enterprise Key Management Foundation in several phases. This covers only phase 1, where the infrastructure is established and the first application is rolled out.

Phase 1: Laying the foundation and the first application

Initially, the infrastructure is established with the capability to generate and distribute keys securely to the issuing and authorization platforms. The bank uses the infrastructure to manage the personal identification number (PIN) and card verification value (CVV) keys that are needed for a few ranges of payment cards.

Phase 2: Remote Key Loading for ATMs

The second phase has the objective to streamline the ATM master key distribution and use asymmetric encryption techniques to load the keys into the ATMs through Remote Key Loading (RKL). This objective requires that all ATMs have PIN pads that support the remote loading of keys, and that the ATM software is updated to handle the RKL protocol. The bank therefore starts planning the ATM RKL project.

Phase 3: Creating and distributing EMV issuer keys

The mandate to implement EMV standards is coming next for Fictional Bank. The bank is planning the EMV rollout. For EMV, the bank must manage issuer keys and certificates, and additional types of symmetric keys for application cryptograms and scripting. The management of EMV keys can be based on the same infrastructure as in phase 1.

Phase 4: Enabling workflows and reporting

To improve the overview of keys and their status, the bank wants to use the IBM Enterprise Key Management Foundation Browser, which enables key managers to access the key repository and verify key status and presence in keystores, all from their own desktop computer. Furthermore, they can create key management requests from the Browser and later run them at the workstation. The bank expects that this function results in fewer visits to the secure room, where they need two people to be present at a time, and thus save time for key management.

Phase 5: Certificate management

The last phase focuses on more efficient certificate management, with a focus on timely renewal and cost savings for internal usage of certificates. The bank uses IBM Enterprise Key Management Foundation to perform regular scans of the network and servers to detect certificates and report on upcoming expirations.

4.6 Conclusion

This chapter introduced our case study, Fictional Bank, which is experiencing numerous problems with key management and compliance. This chapter described their requirements, the solution overview, and the five phases the bank plans for their roll-out of IBM Enterprise Key Management Foundation.



Key management infrastructure setup and deployment

This chapter describes the design, planning options, and implementation details for the IBM Enterprise Key Management Foundation at Fictional Bank.

This chapter introduces the design parameters that are needed for implementation on the following platforms:

- ▶ Key Management Workstation running SUSE Linux
- ▶ z/OS Agent in a sysplex

This chapter describes a specific implementation in a typical IBM Enterprise Key Management Foundation environment.

This chapter includes the following sections:

- ▶ Planning for deployment
- ▶ Implementation
- ▶ Managing keys
- ▶ Link encryption configuration
- ▶ Application keys
- ▶ Key lifecycle management
- ▶ Conclusion

5.1 Planning for deployment

This section describes the design options that are available for a mainframe running z/OS, and a Key Management Workstation running on SUSE Linux.

Fictional Bank is running multiple z/OS images. Each z/OS image is running the IBM Enterprise Key Management Foundation Agent. However, each Agent is configured to perform only those tasks that are required on the specific z/OS image. As described in “z/OS Agent” on page 115, there are several functions the Agent performs.

Fictional Bank has Crypto Agents running on all z/OS images, and the Database Agent running on one z/OS image. The Crypto Agents add keys to the ICSF data sets. The database Agent connects to DB2 to manage the key repository.

5.1.1 System z

The Key Management Workstation communicates with *IBM Cryptographic Services Facility* (ICSF) and the IBM relational database product, *DB2 for z/OS* (DB2) by using an Agent. The IBM Enterprise Key Management Foundation Agent is a started task on *logical partitions* (LPARs) running z/OS.

Parallel Sysplex usage

Multiple IBM Enterprise Key Management Foundation z/OS Agents that are established on separate z/OS images use this high availability environment for key management. The IBM Enterprise Key Management Foundation establishes a session with the Agent running on a preferred z/OS image. This session persists even after the initial conversation with the Key Management Workstation ends, which provides a faster start to the next conversation. Other z/OS images run the Agent as a hot standby. Using Sysplex distributor, a standby Agent may become active if the preferred Agent becomes unavailable.

Hardware security modules on System z

The hardware cryptographic services that are provided in System z are intended to cover the full range of cryptographic operations that are needed for e-business, e-commerce, and financial institution applications, both from the functional and performance standpoints.

The Crypto Express comes as a Peripheral Component Interconnect Express (PCIe) pluggable priced feature that provides high-performance cryptographic functions and a secure cryptographic environment.

Important: Although the CryptoExpress feature is optional from a System z standpoint, only a few cryptographic functions are available without it. The IBM Enterprise Key Management Foundation requires the functionality of the CryptoExpress feature configured as a CCA coprocessor.

The CryptoExpress coprocessor supports symmetric encryption services, random number generation, various types of data hashing, and large number modular math functions for RSA, ECC Prime Curve, and other public-key cryptographic algorithms.

Each Crypto Express3 Feature contains two cryptographic engines that can be independently configured as a CCA coprocessor (CEX3C) or as an accelerator (CEX3A).

Each Crypto Express4S feature contains a single cryptographic engine that can be configured as a CCA coprocessor (CEX4C), as a PKCS#11 coprocessor (CEX4P), or as an accelerator (CEX4A).

When configured as a CCA coprocessor (CEX3C or CEX4C), the full range of cryptographic functions is available to programs invoking the engine, using secure keys whenever they are required. This is the default mode of operation. When configured as an accelerator (CEX3A or CEX4A), the engine provides a limited support that resolves into performing only the RSA operations that are used during the SSL handshake (these are symmetric key wrapping and unwrapping and digital signature verification). However, these operations can then be performed at a higher speed. Secure keys are not supported in accelerator mode.

As shown in Figure 5-1, each co-processor is logically partitioned into 16 separate *domains*.

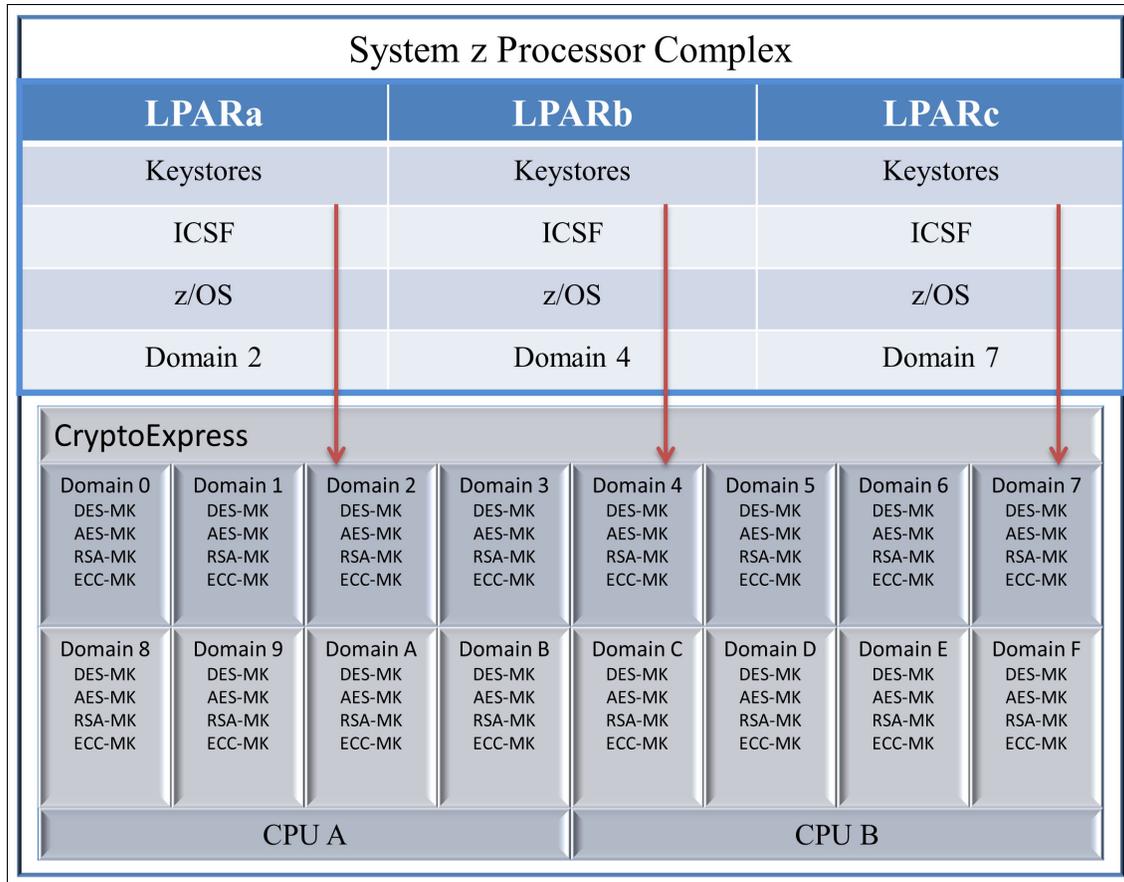


Figure 5-1 Cryptographic domains that are assigned to LPARs

Each domain contains its own set of master keys and operational registers. Cryptographic domains are not shared between LPARs. They operate independently of one another.

Domain numbers are assigned to LPARs through the Hardware Management Console (HMC). An LPAR can have up to 16 domain numbers assigned (this is known as the usage domain index). However, at any given time, the LPAR can use only one of these domains, namely the one that is specified in the LPARs ICSF options data set. When an LPAR is assigned more than one coprocessor, the domain index and the domain number that is specified in the ICSF options data set applies to all those coprocessors.

The IBM Enterprise Key Management Foundation requires the Crypto Express Feature be configured as a coprocessor. Each Crypto Express3 feature contains two PCIe crypto engines. As stated, each engine can be configured as a cryptographic coprocessor or accelerator. During the feature installation, both PCIe adapters are configured by default as coprocessors. Fictional Bank decides to implement the Crypto Express3 feature as two coprocessors.

Figure 5-2 shows Fictional Bank running a System z that is configured in to three z/OS LPARs.

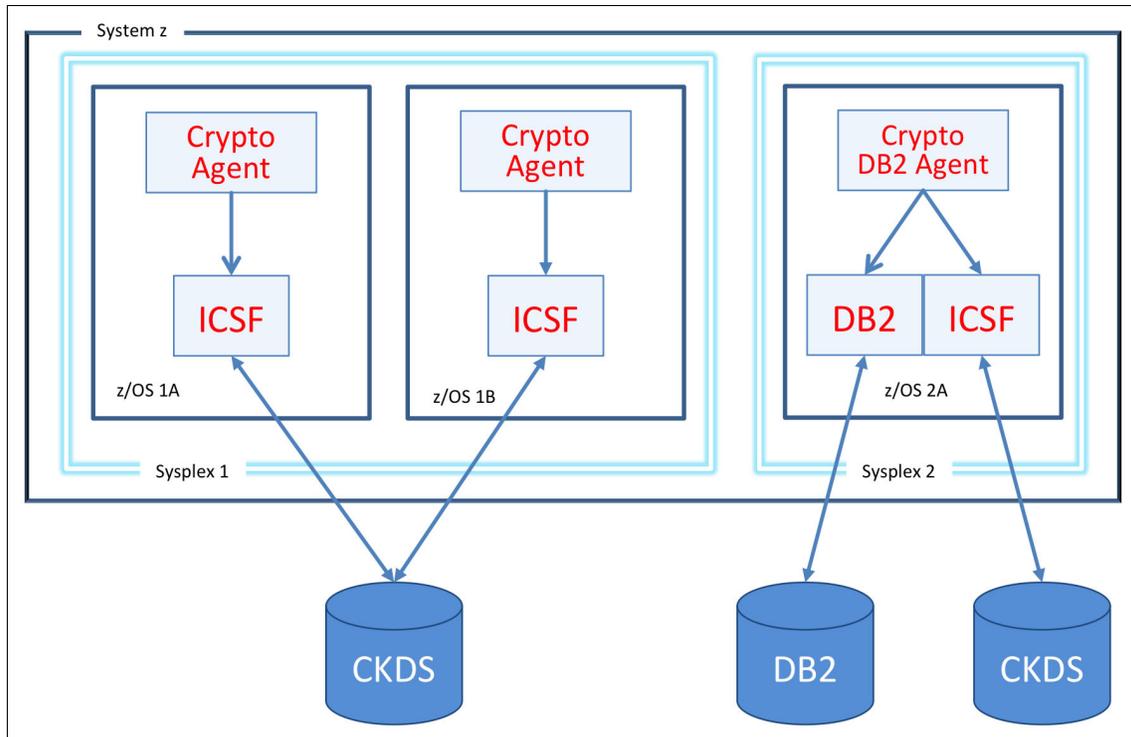


Figure 5-2 System z LPARs on separate CECs

Two LPARs are defined as key acquiring systems, and one LPAR is defined for the issuing system. The acquiring LPARs are a separate sysplex from the issuing LPAR. The acquiring LPARs are logically separated from the issuing LPAR through the usage of different cryptographic master keys. The usage of different master keys requires that the keystores that are accessed by ICSF be separate data sets. Therefore, the cryptographic hardware domains in LPARs z/OS1A and z/OS1B must have the same master keys installed to share the cryptographic key data set (CKDS), while System z/OS2A has different master keys in its domain and therefore a different CKDS. For a description of master keys, see “Hardware security modules on System z” on page 110.

LPARs z/OS1A and z/OS1B are defined to the Key Management Workstation as using keys to verify values such as PIN numbers. LPAR z/OS2A is used for generating these values. Keys that are loaded in to each of the keystores are created by the Key Management Workstation so that they may be used only for the purpose that is indicated by the target system definition.

Security services on z/OS

Requests for security services on z/OS are passed through the System Authorization Facility (SAF). This facility is the interface between system services and the External Security Manager (ESM) that is installed on the system. SAF routes requests for authentication, resource accesses checking, and other security-related processes to the ESM through control points.

There are several security managers available for z/OS. Here are the most popular ones:

- ▶ Secure Server for z/OS Resource Access Control Facility from IBM (RACF)
- ▶ eTrust CA-ACF2 Security for z/OS from Computer Associates (ACF2)
- ▶ eTrust CA-Top Secret Security for z/OS from Computer Associates (TSS)

Installations provide access control rules through the ESM. Each ESM manages these rules differently. Examples within this text use the perspective of IBM RACF to illustrate the concepts.

SAF provides a centralized access control infrastructure that routes application requests to the installed ESM. The applications are independent of the selection of the security manager by the installation. Whether the ESM is RACF, ACF2, or TSS, the application makes the same access control request to SAF. Return and reason codes from the SAF call are the same regardless of the ESM that is installed.

The IBM Enterprise Key Management Foundation Agent requires access to resources in ICSF and DB2, z/OS, the z/OS Communications Server, and RACF. Rules must be established in the security manager to provide the necessary access controls for each resource the Agent is to acquire.

The IBM Enterprise Key Management Foundation requires new profiles in the following classes to support key management operations:

| | |
|-----------------|--|
| FACILITY | General resources that are related to Agent activity |
| CSFSERV | Specific services that are provided by ICSF |
| CSFKEYS | Keys that are stored in ICSF |
| APPL | Connection by the workstation to the Agent |
| DATASET | Data sets that are protected by name |

DB2 security must be established for activities that are related to the Agent. Depending on the local installation, this security may be provided through DB2 grants or security manager profiles.

Some programs that are authorized when started through the TSO/E service facility require inclusion in the AUTHSF list in SYS1.PARMLIB(IKJTS0xx).

Users logging in to the Key Management Workstation must have a z/OS user ID allowing the establishment of a session with the z/OS Agent. The Agent checks rules to ensure that the user has the authority to establish the session.

Audit data collection (SMF logging) can be controlled through a security rule. This allows workstation operations without SMF logging. Access to this rule must be permitted to both the Agent user ID and the user ID at the Key Management Workstation.

Link encryption for the session between the Key Management Workstation and the Agent may be toggled through a security rule. It is advised that this encryption be inactive only during Key Management Workstation and Agent installation and configuration.

ICSF NO-CV key installation is used if keys with the NO-CV option are installed in the ICSF CKDS through IBM Enterprise Key Management Foundation.

The IBM Enterprise Key Management Foundation Agent must have access permission write SMF records.

z/OS Agent

The IBM Enterprise Key Management Foundation z/OS Agent performs several functions for differing request types. The requests might be for the following items:

| | |
|----------------------------|---|
| Cryptography | Accesses ICSF functions. Online management of the keystore, including restore, remove, and backup is supported. |
| Database management | Accesses the IBM Enterprise Key Management Foundation key repository in a DB2 database. |

SMF logging

Requests logging through the z/OS System Monitoring Facility (SMF).

RACF access

Accesses RACF key rings for certificates and key ring information.

Activation of each functional subarea is controlled through configuration options.

The IBM Enterprise Key Management Foundation provides functions for the Key Management Workstation to access keystore files that are controlled by ICSF, and to maintain additional information that is stored in DB2 tables concerning the usage of the encryption keys. All functions are normally activated by using one Agent, and there can be many Agents with cryptography functions that are enabled running in separate LPARs to support the distribution of keys to ICSF.

Figure 5-3 shows multiple z/OS images being managed by a single Key Management Workstation.

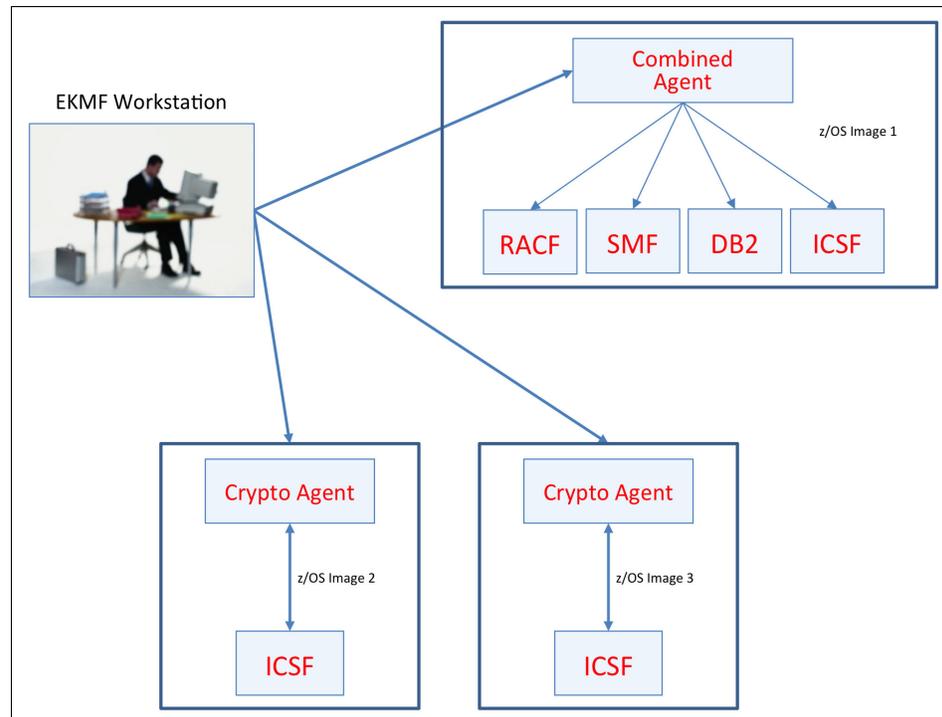


Figure 5-3 IBM EKMF Agents on separate z/OS images

z/OS Image 1 has the z/OS Agent configured for multiple Agent functions. It is used to add keys to the ICSF keystore on that image, store key material and related data in DB2, request SMF recording, and manage RACF key rings.

z/OS Images 2 and 3 are configured only to push keys to ICSF keystores.

ICSF

The IBM Cryptographic Services Facility (ICSF) provides access to the CEX through a set of callable services. The callable services that are available to your applications depend on the configuration of your server and the cryptographic features that are installed. ICSF routes requests with clear RSA keys to accelerators when available.

For more information about the services that are provided by configuration, see *z/OS Cryptographic Services ICSF Overview*, SA22-7519-16.

The ICSF installation options data set describes the cryptographic key data sets, whether ICSF should run in Special Secure Mode (SSM), and what domain should be used. The data set usually is a member of SYS1.PARMLIB. It becomes active when you start ICSF.

DB2

The keys that are generated or imported on the Key Management Workstation are stored encrypted in DB2 on the application server that has the IBM Enterprise Key Management Foundation DB2 Agent configured. To accomplish this task, the Key Management Workstation sends messages to the Agent. The IBM Enterprise Key Management Foundation supports only one DB2 Agent. If you need the information in the DB2 tables on other servers, you must replicate copy or in another way provide the information about the other server.

Keys that are stored in the DB2 key repository may be pushed to keystores when implementing subsequent crypto Agents running on new servers. If a new z/OS image is defined, populating the keystores in ICSF and RACF is done outside of the Key Management Workstation through normal system replication processes, and the Key Management Workstation adds the new z/OS image to its device configuration.

Sysplex distributor HotStandby

The z/OS Communications Server provides sysplex distributor support for a hot-standby server by using a distribution method called *HotStandby*. You configure a preferred server and one or more hot-standby servers. The preferred server that has an active listener receives all new incoming connection requests, and the hot-standby servers act as backup servers if the designated preferred server becomes unavailable. You can rank the hot-standby servers to control which hot-standby server becomes the active server. You can also control whether the sysplex distributor automatically switches back to using the preferred server if it again becomes available, and whether the distributor automatically switches servers if the active target is not healthy.

5.1.2 Key Management Workstation

The Key Management Workstation must be built with specific hardware and software components to fulfill its role as a trusted and secure key management endpoint. In Fictional Bank, the two Key Management Workstations for production are built and configured identically. The test Key Management Workstation is built and configured identically to the production Key Management Workstations, except that it uses different 4765 master keys, a separate key hierarchy, and connects to the test environment instead of the production environment.

The fully configured Key Management Workstation is an IBM System x server with an IBM PCIe 4765 Cryptographic Coprocessor, a network connection, two Omnikey 3812 smart card readers, and a USB attached printer. The server is running the Key Management application on top of the SUSE Linux Enterprise Server operating system

Figure 5-4 on page 119 shows the main Key Management Workstation hardware and software components.

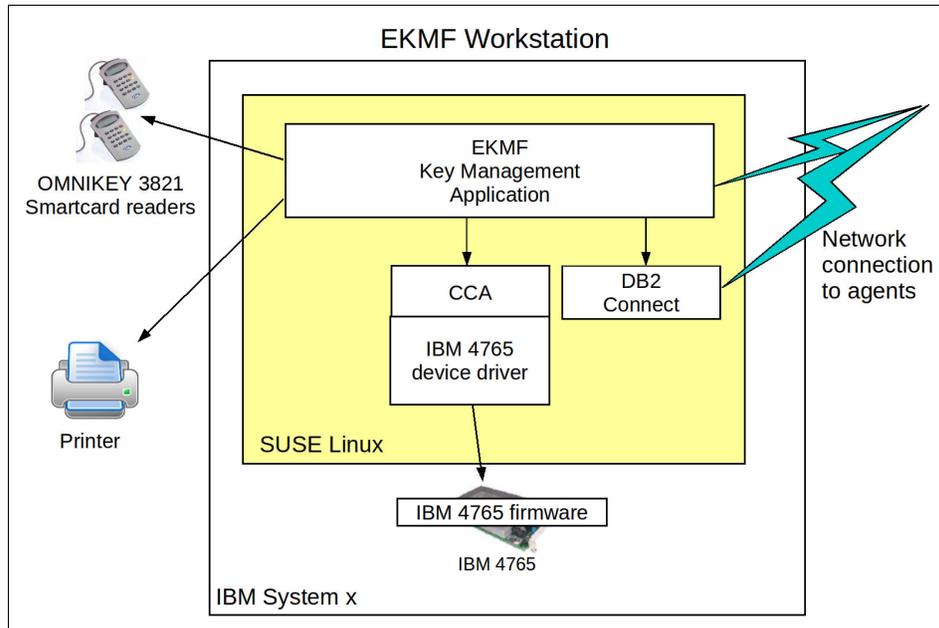


Figure 5-4 Key Management Workstation

Key Management Workstation hardware

Before obtaining the Key Management Workstation hardware, consult with IBM to get information about the available and supported Key Management Workstation hardware configurations. One reason for this task is that only a few server models are tested and approved for use with the IBM PCIe 4765 Cryptographic Coprocessor (IBM 4765).

In the case of Fictional Bank, the Key Management Workstation is based on an IBM x3400 M3 System x server.

IBM PCIe 4765 Cryptographic Coprocessor

The Key Management Workstation needs an IBM PCIe 4765 Cryptographic Coprocessor to authenticate users and run key management-related cryptographic operations securely. The coprocessor must be installed in a PCIe slot in the workstation.

The IBM 4765 must be loaded with specialized firmware. This firmware is specific to, and delivered together with, the Key Management Workstation application software.

The IBM 4765 also must be configured with user profiles, access rights, and cryptographic master keys. Smart cards are used to authenticate users to the IBM 4765 and to keep key parts for the master keys.

OMNIKEY 3821 smart card readers

In Fictional Bank, the Key Management Workstation users authenticate through personal smart cards. Smart cards also are used to store IBM 4765 master key parts. To accomplish this task, the Key Management Workstation must be configured with two USB-attached OMNIKEY 3821 Cardman smart card readers. For most operations, including logon and master key management, only one reader is needed. Two readers are needed in some situations, for example, when initializing smart cards and when copying data from one smart card to another.

Printer

Fictional Bank keeps recovery keys in clear parts on paper and also anticipates the need to exchange keys in clear parts on paper with external parties. The bank therefore needs a USB printer that is attached to the Key Management Workstation.

Network connection

The Key Management Workstation must have wired access to Fictional Bank's internal network to connect to the IBM Enterprise Key Management Foundation Agents on z/OS. This must be considered when establishing secure rooms for the production Key Management Workstations.

SUSE Linux Enterprise Server operating system

The Key Management Workstation uses the SUSE Linux Enterprise Server operating system, which must be obtained and installed. For information about the specific version that is needed, consult the *IBM DKMS Key Management Workstation Installation Guide for SLES 11*, DKMS-4050. In the case of Fictional Bank, the version is 32-bit SUSE Linux Enterprise Server Version 11 Service Pack 2.

IBM DB2 Connect

The Key Management Workstation uses DB2 Connect to access the EKMF key repository on z/OS. The DB2 Connect license must be obtained from IBM and installed. For information about the specific version of the license file that is needed, see the *IBM DKMS Key Management Workstation Installation Guide for SLES 11*, DKMS-4050.

IBM 4765 CCA software

The IBM 4765 needs device drivers, CCA API software, and CCA utilities to be installed on the Key Management Workstation. The required installer is delivered together with the Key Management Workstation application software.

Key Management application

Finally, the DKMS application must be installed and configured. The installation process is described in the *IBM DKMS Key Management Workstation Installation Guide for SLES 11*, DKMS-4050.

After installation, the key management application must be configured with the definition of users, access rights, database connections, Agent connections, and key templates. Also, system keys must be generated.

5.1.3 Keys to be managed

A *key hierarchy* exists where a cryptographic key is encrypted (protected) through another key, which again might be protected by yet another key, and so on. Only the key at the top of the hierarchy exists unprotected by another key and must therefore be protected by hardware. This results in a hierarchy with any number of keys that all are as protected as the master key in the IBM 4765 hardware security module.

The IBM Enterprise Key Management Foundation allows for creation and management of key hierarchies to provide separation of keys. Hierarchies always are tailored to the specific business of an installation, but the default template key hierarchies are designed based on the following criteria:

- ▶ The backup of one key (the *Top key* or the *Disaster recovery key*) in addition to the backup of the key repository is sufficient to recover all keys (if necessary).
- ▶ It must be possible to create individual exchange keys for each group of systems to which the keys can be distributed. These keys usually are referred to as Zone Master Keys (ZMKs).
- ▶ Application keys must not be immediately available for use on the Key Management Workstation. The key management system may provide the keys, but not use them.

Figure 5-5 shows a standard key hierarchy for symmetric keys.

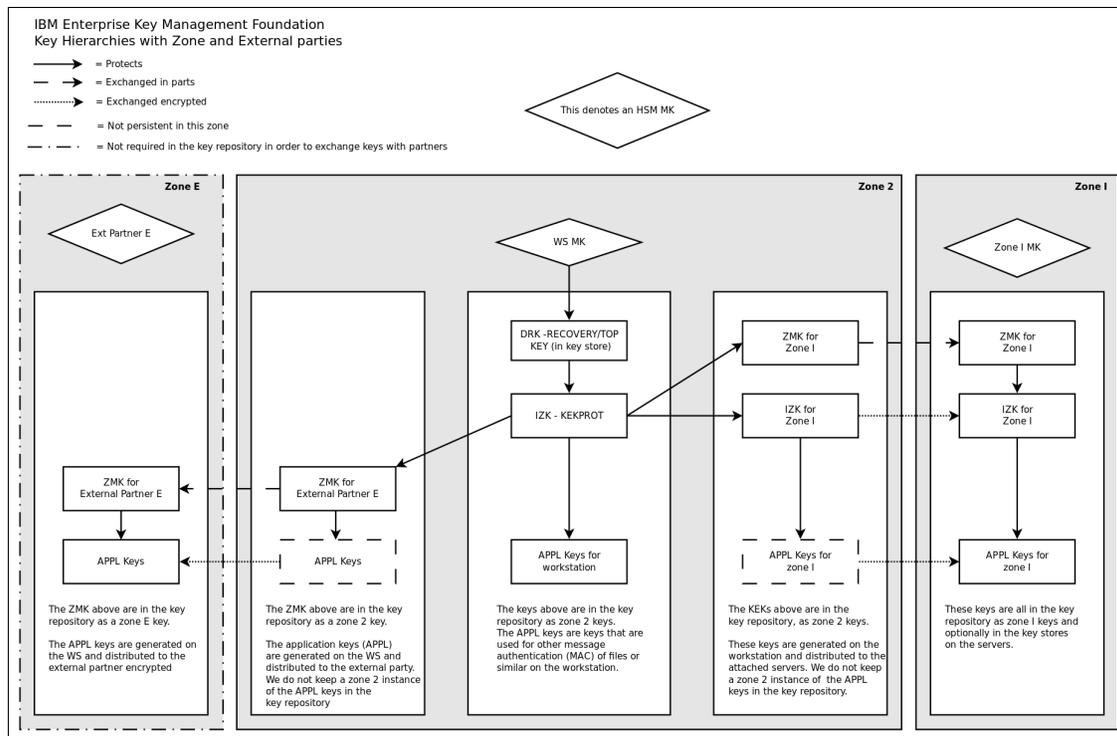


Figure 5-5 IBM Enterprise Key Management Foundation key hierarchy overview

The same hierarchy structure is used for both DES and AES keys. Asymmetric keys are exchanged through a symmetric key exchange key, and therefore appear in the hierarchy as one of the application keys. Key are organized in zones, where each zone has its own hierarchy of keys ending in a top-level key. Each zone is assigned one alphanumeric character as identification. The business needs of Fictional Bank are covered by planning for three zones:

- Zone 2** Traditionally, the key hierarchy that is used at the Key Management Workstation is referred to as 'zone 2'. The top-level key is known as the *disaster recovery key*.
- Zone I** Zone I holds all keys for the Issuing system at Fictional Bank. There is a zone A for the authorization system, which is not shown in Figure 5-5.
- Zone E** This is a zone for an external partner.

Each key in Figure 5-5 on page 122 is explained in Table 5-1.

Table 5-1 Types of keys

| Key type | Description |
|--|---|
| Workstation master key | This is the master key of the IBM 4765 on the Key Management Workstation. It protects any keys that are stored in the local key storage. The master key itself is protected by the IBM 4765. |
| DRK - Disaster recovery key or top key | This is the top-level key in the zone 2 hierarchy. Having a backup of this key available (by keeping it in parts on paper) ensures that all keys in the key repository across zones can be recovered from backups. The key is stored in the key repository and in the zone 2 keystore, where it is enciphered by the workstation master key. Recovery also can be based on the workstation master key, which is kept in parts on smart cards. We recommend (but do not require) to both print the recovery key and keeping the workstation master key on smart cards. |
| IZK KEKPROT KEK protection key | An intermediate KEK in the workstation zone that is used to protect all exchange keys with other zones plus keys that are used in the workstation zone (zone 2). |
| ZMK for zone I | This is the top-level key for zone I. This key is shared with the workstation key zone and exchanged in parts. It is loaded into the keystore of the issuing system. Possession of this key provides access to all keys in the zone. |
| IZK for zone I | To facilitate frequent changes of the used exchange keys, an intermediate KEK is introduced in the key hierarchy. It is exchanged encrypted under the ZMK for Zone I. |
| ZMK for external partner | For each external party or entity with which you exchange keys, you need a ZMK. The initial key is exchanged in parts if you use symmetric mechanisms. |
| Application keys | These are the keys that are used by the applications in the organization. They are generated or received on the Key Management Workstation, and distributed to the keystores and possibly to the external partners. In general, these keys exist only in zone A and I and not in zone 2. There are situations where the actual use of the key takes place on the Key Management Workstation. |

5.1.4 Keys to be managed for the application

The sample application that we use in phase 1 is payment cards with PIN and CVV. The issuing system generates the PINs and sends them to an external party to be printed and mailed to the card holders. The PINs are stored encrypted on the issuing system so that the PIN can be carried forward when the payment cards are replaced.

The authorization system has master keys for each ATM, and a zone master key that is shared with each network, so that it can exchange session/working keys.

Table 5-2 shows the different key types that are needed on the issuing and authorization systems.

Table 5-2 Key types for a magnetic stripe card application

| Purpose of key | ICSF key type for issuing system (zone I) | ICSF key type for authorization system (zone A) |
|--|---|---|
| PIN generation | PINGEN | PINVER |
| CVV calculation | MAC | MACVER |
| For protection of encrypted PINs | IPINENC and OPINENC | - |
| PIN protection key that is used to send PINs to print at the print vendor (exchange key) | EXPORTER with NO-CV option (used to export an OPINENC key for each file to be sent) | - |
| Terminal master keys for ATMs | - | EXPORTER (NO-CV) |
| ZMK with payments network | - | EXPORTER (NO-CV) |

5.1.5 Key label naming convention

It is essential to choose a good naming convention for the keys. It is valuable to understand the purpose and capabilities of a key when you see its label. The applications must be able to build a label when addressing the key in ICSF, and an authorization should be able to build the label from a skeleton and the data it receives in a transaction. Furthermore, RACF on z/OS can be used to protect the keys, and to enable easy administration, a good naming convention is needed.

Users of IBM Enterprise Key Management Foundation have used a naming convention like the following one. The key label consists of 4 - 5 blocks, which are separated with a ':' (for example, A.B.C.D.E). Each block is constructed from one or more items called tags. A tag looks like this: *<tag>*.

The key label has the following structure before the tags are substituted with real values:

<hierarchy><key-category><algorithm>.<application>.<more-info-1>.<more-info-2>.<keytype><seqno>

The tags have the following meaning:

- <hierarchy>** A one-letter identification of the key hierarchy. An IBM Enterprise Key Management Foundation instance can be configured with a value for this tag. Its typical use is to separate production, system test, and test environments, for example, by assigning the values P, S, and T, respectively.
- <key-category>** This tag is used to designate keys so that their high-level purpose is clear. A three-letter value. A list of values is provided in Table 5-3.
- <algorithm>** The cryptographic algorithm with which the key can be used. Values are AES, DES, RSA, and ECC.
- <application>** The business application with which the key is used.
- <more-info-1>** This tag is used to provide additional information about the key.
- <more-info-2>** This tag is used to provide additional information about the key.
- <keytype>** A three-letter abbreviation of the CCA key type, such as EXP for EXPORTER. The system can do this tag automatically.
- <seqno>** The sequence number. This tag is used for keys where versioning is needed.

Table 5-3 List of key categories

| Key category | Description |
|---------------------|--|
| ZMK | Zone master key. Exchange key between two key zones. Usually exchanged in parts. |
| IZK | Internal Zone Key. An exchange between two zones internally in the enterprise. |
| IDM | Internal Data or MAC key. |

| Key category | Description |
|--------------|---------------------------------------|
| TMK | Terminal Master Keys for ATMs. |
| PGK | PIN generation Keys. |
| CVK | CVV generation keys. |
| DKG | Key derivation keys. |
| IPK/XPK | Internal/External PIN protection key. |

5.2 Implementation

This section focuses on the implementation details for the design that is described in 5.1, “Planning for deployment” on page 110.

5.2.1 System z

This section describes the details that are required for implementing the EKMF Agent on z/OS. The following areas are covered:

- ▶ z/OS Agent
- ▶ ICSF
- ▶ DB2
- ▶ Security rules that are required
- ▶ Sysplex distributor HotStandby

z/OS Agent

The IBM Enterprise Key Management Foundation Agent on z/OS runs as a TCP/IP daemon listening on a port. It runs as a started task. The IKJEFT01 program is running unattended. Because the TSO environment is used, and the Agent uses authorized services, additional services must be added to the TSO authorized services list in IKJTSOxx.

Make the following changes to the installation’s SYS1.PARMLIB(IKJTSOxx) data set:

1. Add these modules to the AUTHTSF table:
 - KMGPRACF
 - KMGPNOCV

- KMGPCCMD¹
 - CSFEUTIL²
2. After updating the data set, run **PARMLIB** to activate the changes:
PARMLIB UPDATE(XX)
xx is the suffix of the IKJTSoxx member. Typical installations use 00.
 3. Check for changes by running **PARMLIB LIST**:
PARMLIB LIST
 4. Look for the AUTHTSF section in the output for the following module names:
CURRENT PARMLIB SETTINGS FOR AUTHTSF:

```

AARSERVE  KMGPRACF  KMGPNOCV  KMGPCCMD  CSFEUTIL  IEBCOPY
EQQMINOR  IHVUUSD    IKJEFF76  DB2TSOUT  ITPENTER  CSFDAUTH
CSFDPKDS  IRLMFCMD    MXITSF    IOAMAIN   SMPPEEK   SPACE
ICQASLIO  KMGPRACF    KMGPNOCV  KMGPCCMD  CSFEUTIL  EQQMINOI

```

Parameters that are set in the data set are pointed to from the KMGPARM DD card in the z/OS Agent started task JCL.

Profiles in the FACILITY class control certain aspects of Agent operations:

CRYPTO.DKMS.AUDITOFF

Turns off SMF auditing on z/OS.

READ access allows operation without SMF audit records being requested.

Required by both the z/OS Agent user ID and users logging on to the z/OS Agent through the Key Management Workstation.

The profile should be permitted only during the initial setup of the Agent and Key Management Workstation.

The profile XCSFMVR of the class CSFSERV must be defined and permissions to DKMS Started Task User must be given before any write of AUDit Log records can be written to SMF.

Define with AUDIT(ALL(READ)).

¹ Required when using the RACF certificate and key ring management features.

² This module is used to refresh the in-storage copy of the CKDS for ICSF.

CRYPTO.DKMS.CCFNOCVF

Allows NOCV keys to be defined to ICSF from the Key Management Workstation.

READ access allows NOCV keys.

Required by both the z/OS Agent user ID and users logging on to the z/OS Agent through the Key Management Workstation.

CRYPTO.DKMS.KMGPCCMD

Controls access to RACF key rings.

READ access allows key ring browsing.

Required by the z/OS Agent user ID.

UPDATE access allows key ring administration.

Required by users logging on to the z/OS Agent through the Key Management Workstation.

CONTROL access allows the addition, alteration, and deletion of certificates in RACF.

Required by users logging on to the z/OS Agent through the Key Management Workstation.

CRYPTO.DKMS.KMGPRACF

Allows user IDs to start the z/OS Agent started task. READ access allows the task user to start the z/OS Agent.

Required by the user ID that is assigned to the z/OS Agent started task.

CRYPTO.DKMS.KMGPRACF.<task ID>

Allows user IDs to log on to a specific z/OS Agent through the Key Management Workstation.

READ allows the user to access the z/OS Agent.

Required by a user logging on to the z/OS Agent through the Key Management Workstation.

CRYPTO.DKMS.LNKCRYOFF

Manage link encryption.

NONE access turns on link encryption.

READ access allows unencrypted sessions.

Required by the z/OS Agent user ID.

The profile should be permitted only during the initial setup of the Agent and Key Management Workstation.

Also, the z/OS Agent requires ALTER access in the DATASET class to the HLQ that is specified in the &RACFCERT-HLQ parameter. This is the data set that is created when digital certificates are exported from RACF.

ICSF

You can configure the IBM Cryptographic Services Facility by using an options data set. That data set usually is associated with the SYS1.PARMLIB concatenation. It is pointed to from the ICSF started task. The relevant options for setting up the IBM Enterprise Key Management Foundation Agent on z/OS are listed below³:

| | |
|----------------------|--|
| CKDSN() | Names the symmetric key data set. Used to store DES and AES keys. |
| PKDSN() | Names the asymmetric keys data set. Used to store RSA and ECC keys. |
| SSM() | Specifies whether an installation can enable special secure mode (SSM) while running ICSF. SSM lowers the security of your system to let you enter clear keys and generate clear PINs. The default value is SSM(NO). |
| SYSPLEXCKDS() | ICSF joins the ICSF sysplex group SYSICSF and this system participates in sysplex-wide consistency for CKDS data. |
| SYSPLEXKDS() | ICSF joins the ICSF sysplex group SYSICSF and this system participates in sysplex-wide consistency for PKDS data. |

Fictional Bank uses two separate ICSF configurations across the enterprise. Two LPARs share their key data sets, and therefore have the same master keys set in the CEX3C hardware. Another LPAR uses separate key data sets. The master keys may be different or the same as the shared environment.

³ For more information about the ICSF Options data set and its contents, see the *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520-17.

ICSF data sets may be accessed directly by the z/OS Agent.

DB2

IBM Enterprise Key Management Foundation uses DB2 to store key-related material. This is referred to at the *Key Repository*.

During installation, tables spaces, tables, and plans are installed into an existing DB2 subsystem. The installed CNTL data set contains JCL to bind the programs to the DB2 plans.

It is necessary to edit the installation data sets to reflect the current DB2 environment.

All tables should be placed in the same DB2 database, and if this already true, then SKMGSQL(KMGJCSQL) must be changed so that the SKMGSQL(KMGIMDEF) DD statement becomes a comment instead of part of the SYSIN concatenation.

Here is the first-time installation procedure. All data sets are members of the installed HLQ.KMGnnnn.SKMGSQL data set.

1. Change member KMGISQLI to reflect the CURRENT SQLID.
2. Change member KMGIMDEF to match the storage group, database, and volume names that are *used for the DKMS tables*.
3. Change the following members to suit the local naming standard for the DB2 table space:
 - KMGCSPRM
 - KMGCSUK1
 - KMGCSUK2
 - KMGCSUK3
 - KMGCSUK4
 - KMGCSAUD
 - KMGCSSER
 - KMGCSPKA
 - KMGCSELT
4. Change the following members to match the local storage group names:
 - KMGCTPRM
 - KMGCTUK1
 - KMGCTUK2
 - KMGCTUK3
 - KMGCTUK4
 - KMGCTAUD
 - KMGCTSER

- KMGCTPKA
 - KMGCTELT
5. Change the member KMGISUB to the DB2 subsystem name that will be used.
 6. Change the member KMGITEP2 to point to the library where the DB2 sample program DSNTEP2 is installed.
 7. Edit the job in member KMGJCSQL to meet your installation requirement for jobs. Ensure the marked DD cards point to the appropriate data sets.
 8. Submit the KMGJCSQL job.

After KMGJCSQL runs successfully, the DB2 environment is set up and ready for the bind processes.

Security rules that are required

Implementing access controls in RACF is performed through the usage of profiles in classes. A profile is a description of the resource and its associated *access control lists* (ACLs). Classes are collections of profiles protecting similar resources. System users are defined by their user ID profile. Groups are collections of user IDs. User IDs and group IDs may be on an ACL.

Profiles are grouped according to resource type in to *classes*. In most cases, classes must be active in order for the profile rules to take effect. EKM uses only standard RACF classes, such as FACILITY, CSFSERV, and CSFKEYS, so no new RACF classes must be defined.

For the purposes of this book, profile management is performed through the usage of RACF TSO commands.

These commands take position sensitive parameters in the formats that are shown in the following list:

| | |
|----------------|---|
| RDEFINE | Defines general resource profiles to RACF. RDEFINE <i>class_name profile_name</i> UACC(NONE) |
| RALTER | Alters existing general resource profiles in RACF. RALTER <i>class_name profile_name</i> OWNER(<i>new_owner</i>) |
| RLIST | Lists existing general resource profiles in RACF. RLIST <i>class_name profile_name</i> |
| RDELETE | Removes general resource profiles from RACF. RDELETE <i>class_name profile_name</i> |

PERMIT

Provides access rules for the specified resource.

PERMIT *profile_name* CLASS(*class_name*) ID(*user or group id*) ACCESS(*level of authority*)

ISCF protection profiles

You can create RACF protection profiles in the CSFSERV class for the listed ICSF services, as shown in Table 5-4.

Table 5-4 ICSF services

| Service | Function |
|---------|--------------------------|
| CSFDEC | Decipher |
| CSFDSV | Digital Signature Verify |
| CSFENC | Encipher |
| CSFIQF | ICSF Query Facility |
| CSFKGN | Key Generate |
| CSFKIM | Key Import |
| CSFKRC | Key Record Create |
| CSFKRD | Key Record Delete |
| CSFKRR | Key Record Read |
| CSFKRW | Key Record Write |
| CSFKYT | Key Test |
| CSFMGN | MAC Generation |
| CSFMVR | MAC Verification |
| CSFOWH | One Way Hash |
| CSFPKI | PKA Key Import |
| CSFPKRC | PKA Key Record Create |
| CSFPKRD | PKA Key Record Delete |
| CSFPKRR | PKA Key record Read |
| CSFPKRW | PKA Key Record Write |
| CSFSYI | PKA Symmetric Key Import |
| XCSFMVR | SMF Auditing |

You might want to define discrete profiles for every resource. To do so, run the following command:

```
RDEFINE CSFSERV service_name UACC(NONE)
```

You might also want to define generic profiles for specific cases. To use generic profiles, **GENERIC** must be enabled for the class in the RACF options. Set **GENERIC** on for the CSFSERV class⁴:

```
SETR GENERIC(CSFSERV)
```

Create the generic CSFSERV class profile by using wildcard characters in the following command:

```
RDEFINE CSFSERV CSF* UACC(NONE)
```

This profile covers all CSF services because they all have “CSF” as the first three characters in the name.

Activate the class and put its profiles in storage by running the following command:

```
SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV)
```

If the CSFSERV class already is in use, run the following command instead:

```
SETROPTS RACLIST(CSFSERV) REFRESH
```

Started Task protection

The STARTED class allows you to assign RACF identities to started procedures and jobs dynamically by using the **RDEFINE** and **RALTER** commands. Unlike the started procedures table, the STARTED class does not require you to modify code or perform an IPL to add or modify RACF identities for started procedures. It provides, in effect, a dynamic started procedures table.

Use the RACF STARTED class to define profiles for the EKMF Agent, ICSF, and DB2.

Resource names in the STARTED class are of the form membername.jobname, for example, CICS.JOBA, CICS.REGION2, or IMS.PROD. The resource name is of the form membername.membername if no job name is provided.

User IDs for the started tasks should be *protected IDs*. They have no passwords that can expire or be used to set the user ID to a revoked status.

⁴ For more information about generic profiles and their uses, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683-15.

Let us use an example to add the user IDs for ICSF, the EKMF Agent, and DB2 subsystems. This example uses STCxxx for the user ID of the task. You should follow normal installation naming conventions.

Create the user IDs as protected IDs:

```
ADDUSER (STCCSF, STCKMS, STCDB2) NOPASSWORD
```

List the IDs to ensure that they have the correct attributes:

```
LISTUSER STCCSF
```

```
USER=STCCSF NAME=UNKNOWN OWNER=L3TS02 CREATED=13.175  
DEFAULT-GROUP=ZLP3 PASSDATE=N/A PASS-INTERVAL=N/A  
PHRASEDATE=N/A  
ATTRIBUTES=PROTECTED
```

Create the started task profiles:

1. Allow generic profiles for the class by running the following command:

```
SETROPTS GENERIC(STARTED)
```

2. Define the STARTED profile or the CSF task (ICSF) by running the following command:

```
RDEFINE STARTED CSF.* UACC(NONE)  
STDATA(USER(STCCSF)  
GROUP(STCGROUP))  
DKMS*.*
```

3. Define the STARTED profile or the IBM Enterprise Key Management Foundation Agent by running the following command:

```
RDEFINE STARTED DKMS*.* UACC(NONE)  
STDATA(USER(STCKMS)  
GROUP(STCGROUP))
```

4. Activate the class and put its profiles in storage by running the following command:

```
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
```

If the STARTED class already is in use, running the following command instead:

```
SETROPTS RACLIST(STARTED) REFRESH
```

Create an additional STARTED profile for DB2 if necessary.

Sysplex distributor HotStandby

The z/OS Communications Server⁵ provides sysplex distributor support for a hot-standby server through the use of a distribution method called HotStandby. You configure a preferred server and one or more hot-standby servers.

In a sysplex distributor environment, the sysplex is assigned an IP address, as is each LPAR in the sysplex. The z/OS systems are configured as a subplex for TCP/IP workload distribution.

Configure the hot-standby function by specifying the following parameters and options on the **VIPADISTRIBUTE** statement of the **VIPADYNAMIC** statement in the TCPIP.PROFILE data set, as shown in Figure 5-6.

```
VIPADYNAMIC
  VIPADISTRIBUTE DISTMETHOD HOTSTANDBY AUTOSWITCHBACK
    shared_IP_address
    DESTIP local_IP_address_1 PREFERRED
        local_IP_address_2 BACKUP 100
    VIPAROUTE shared_IP_address local_IP_address1
ENDVIPADYNAMIC
```

Figure 5-6 Example VIPADYNAMIC statement for HotStandby

IPCONFIG SYSPLEXROUTING must be specified on all target systems for this distribution method to be used.

5.2.2 Key Management Workstation

This section describes the installation and configuration steps of the Key Management Workstation for Fictional Bank.

Installing and configuring the SLES operating system

This section provides an installation guide for SUSE Linux Enterprise Server (SLES) Version 11 Service Pack 2.

Unless otherwise noted, SLES is installed with the default options, except for the keyboard layout and network settings.

As Linux is a case-sensitive operating system, any entries regarding paths and file names must be typed in the same case as at entry time.

⁵ For more information about Sysplex distributor and virtual IP addressing, see *z/OS Communications Server: IP Configuration Reference*, SC31-8776-21.

Additional information: Obtain the information about your network configuration before you start an installation. Check with your network administrator to get the proper configuration settings for your workstation.

Additional instructions for installing and upgrading the operating system can be found at the following location:

<http://www.novell.com/documentation/sles11/>

To install SLES on your System x, complete the following steps:

1. Boot the machine from CD and click **Installation** on the start menu.
2. In the Welcome window, select **English (US)** language and the keyboard layout that is applicable for the connected keyboard. Read and agree to the license terms and click **Next**.
3. In the Media Check window, click **Start Check** to verify the installation media. After the check is successful, click **Next**.
4. In the Installation Mode window, verify that the mode is set to **New Installation** and click **Next**.
5. In the Clock and Time Zone window, select the correct Region and Time Zone for the current location and click **Next**.
6. In the Server Base Scenario window, verify that the scenario is set to **Physical Machine** and click **Next**.
7. In the Installation Settings window, review the choices that are made in the installation wizard and click **Install** to begin the installation.
8. Wait for the installation to finish (this might take a while).
9. In the Root Password window, enter a password for the root user, which is the administrator of the SLES operating system. Click **Next** to continue.
10. In the Host name and Domain Name window, enter the host and domain name according to your network configuration.
11. In the Network Configuration window, configure the network settings according to your network configuration.⁶ Click **Change** → **Network Interfaces** to configure the common IP settings. Click **Next**.
12. In the Test Internet Connection window, skip the test, as the Key Management Workstation is not supposed to be connected to the internet.
13. In the Network Services Configuration window, keep the default settings and click **Next**.

⁶ A faulty network configuration can prevent the Key Management Workstation from running.

14. In the User Authentication Method window, keep the default settings and click **Next**.
15. In the New Local User window, enter the user name *dkms*⁷ and the password in the User Name and User Password fields and click **Next**. More local users can be created after SLES is installed.
16. In the Release Notes window, optionally review the release notes and click **Next**.
17. In the Hardware Configuration window, review the configuration and click **Next**.
18. In the Installation Complete window, click **Finish** to complete the installation.

After completing this process, the system should be ready to use with the root and dkms users with their corresponding passwords.

Installing more software: Unless otherwise specified, any further software you are installing on the Key Management Workstation should be installed by the root administrator user without using the **sudo** command.

Installing the SLES utilities and additional packages

The following additional packages are needed for the installation of the CCA driver that is described in “Installing the IBM 4765 CCA software” on page 138:

- ▶ The gcc compiler
- ▶ Kernel-source and kernel-syms
- ▶ Java version 1.6.0 and above

To install these packages, complete the following steps:

1. On the taskbar of SLES (similar to a Windows operating system), click **Computer** → **Install Software** and enter the package name that is listed above in to the Search field, for example, gcc for the gcc compiler.
2. Select to install any further associated packages suggested by the wizard and proceed with the installation.
3. Repeat steps 1 and 2 for all packages to install.

Installing a printer

Install the USB printer by using the installation manual that comes with the printer.

⁷ User names in Linux can be no longer than eight characters, contain only lowercase letters, numbers, and underscore (_), and cannot begin with IBM, SQL, SYS, or a number.

Note: The installed printer name must be no longer than 12 characters.

Special printers on serial or parallel ports might need to be set up in SUSE through a *Raw Queue* [raw] print driver to avoid unwanted formatting.

Installing the IBM 4765

To install the IBM 4765 hardware, follow the instructions in Chapter 2 of the *4765 PCIe Cryptographic Coprocessor Installation Manual*, which can be found at the following website:

<http://www.ibm.com/security/cryptocards/pciicc/library.shtml>

Handle the IBM 4765 correctly according to the instructions in the manuals. In particular, pay attention to the handling of batteries, as described in the Battery Maintenance section in Chapter 3 of the *4765 PCIe Cryptographic Coprocessor CCA Installation Manual*.

Connecting the smart card readers

Connect the two OMNIKEY Cardman 3821 smart card readers to the USB ports of the Key Management Workstation.

The readers can be connected at any time, but do not function until after you complete the IBM Enterprise Key Management Foundation installation.

Installing the IBM 4765 CCA software

The IBM 4765 CCA software package Version 4.3.5 that is supported on SLES 11 SP2 is on the DKMS CD in the `ibm4765` directory. More information about the installation of this software can be found in *4765 PCIe Cryptographic Coprocessor CCA Support Program Installation Manual Release 4.4*, which can be found at the following website:

<http://www.ibm.com/security/cryptocards/pciicc/library.shtml>

To install the IBM 4765 CCA software with smart card support, complete the following steps:

1. Insert the IBM Enterprise Key Management Foundation installation CD (Version 8.3.1).
2. On the taskbar, click **Computer** → **More Applications** → **GNOME Terminal** to start the GNOME Terminal, similar to the DOS command window.
3. Enter the following command or use Midnight Commander⁸ to copy the contents of the CD, mounted under the `/media/IBM DKMS 8.3.1` folder:

```
cp /media/IBM DKMS 8.3.1 /tmp
```

4. Move to the folder containing the IBM 4765 CCA software installation files by running the following command:

```
cd /tmp/IBM DKMS 8.3.1/ibm4765
```

5. Verify the access rights and the owner of the installation files⁹ by running the following command:

```
ls -al
```

6. To start the IBM 4765 CCA software installer, run the following command:

```
./setup4765_4.3.5.bin
```

Note: If the GUI-based installer does not run, verify and change the access rights of the installation files to `r-xr-x---` and the owner to `root:root`.

7. Follow these installation steps and leave the default values unchanged:
 - a. Click **Custom** to choose the installation set.
 - b. Select **Device Driver Source** and the **Smart Card Utility Programs** check boxes to install all possible subprograms and click **Next**.
 - c. Leave the default installation folder unchanged (`/opt/IBM/4765`) and click **Next**.
 - d. Leave the default key storage directory unchanged (`/opt/IBM/4765/keys`) and click **Next**.
 - e. Leave the default keystore files names unchanged (`des.key`, `deslist`, `pka.key`, `pkalist`, `aes.key`, and `aeslist`) and click **Next**.
8. After the installer runs, click **Computer** → **Shutdown** → **Restart** to reboot the system.

The IBM 4765 CCA software installer installs and compiles the CCA driver that is needed for the PCe cryptographic coprocessor to be recognized by the system, the CNM Utility program, which is used to configure the coprocessor and the master keys, and also the Smart Card Utility Program (SCUP), which is needed to set up the coprocessor for smart card support.

⁸ Midnight Commander is similar to the Norton Commander, Total Commander, or Free Commander programs. In our example installation, we use it to move through folders and manage files. The two panels of this program can be used to copy or move files from one source folder (to be chosen in one panel) to another destination folder (to be chosen in the other panel). Type the command `mc` to start Midnight Commander.

⁹ The installation files must have the owner `root:root` and at least the Read and Execute access rights for the user or group (for example, `r-xr-x---`). Use the Linux commands `chown` or `chmod` to change the owner or the access rights of the files and folders.

Be patient: Even if the smart card support is installed, the smart cards are not available for use in the IBM 4765 software and you cannot load the IBM 4765 master keys from smart cards until the IBM Enterprise Key Management Foundation installer is run.

Installing the IBM 4765 firmware

The IBM 4765 contains firmware that must be updated to be compatible with the Key Management Workstation. This firmware is occasionally updated along with new IBM Enterprise Key Management Foundation versions, which is noted in the release notes. The firmware is loaded in three memory segments in the IBM 4756, where each segment has different responsibilities. More information about the segments is available in Chapter 4 of the *4765 PCIe Cryptographic Coprocessor CCA Support Program Installation Manual*.

Updating the 4765 firmware is done through the CLU utility that is installed with the IBM 4765 CCA software package. The usage of CLU is documented in Chapter 4 of the *4765 PCIe Cryptographic Coprocessor CCA Support Program Installation Manual*. Using CLU, it is possible to get a status of the current firmware levels of the IBM 4765 through the **ST** command.

In our scenario, the cryptographic coprocessor is delivered to the customer, so segment 1 is in the Factory state and segments 2 and 3 are OWNER: UNOWNED. If you change any of these states, consult Chapter 4 of the *4765 PCIe Cryptographic Coprocessor CCA Support Program Installation Manual*.

Reading the IBM PCIe 4765 Cryptographic Coprocessor status

To read the IBM PCIe 4765 Cryptographic Coprocessor status, complete the following steps:

1. On the taskbar, click **Computer** → **more Applications** → **GNOME Terminal** to open a GNOME Terminal.
2. To move to the installation folder of the IBM 4765 CCA software, run the following command:

```
cd /opt/IBM/4765/clu
```
3. To read the coprocessor status, enter the following command¹⁰:

```
./csulclu log20130709.txt ST 0
```

¹⁰ You can enter any log file name. In our example, we use the actual date for better traceability in case more log files are created over time.

Example 5-1 shows the output of this command.

Example 5-1 Read status of the IBM 4765 before loading any segments

```
CSULCLU V4.1.2 log20130709.txt ST 0   begun Tue Jul  9 10:25:08 2013
***** Command ST started. ---- Tue Jul  9 10:25:08 2013

*** VPD data; PartNum = 41U9986
*** VPD data; EC Num = N44178B
*** VPD data; Ser Num = 12345678
*** VPD data; Description = IBM 4765-001 PCIe Cryptographic Coprocessor
*** VPD data; Mfg. Loc. = 91
*** ROM Status; POST0 Version 1, Release 43
*** ROM Status; MiniBoot0 Version 1, Release 20
*** ROM Status; INIT: INITIALIZED
*** ROM Status; SEG2: UNOWNED , OWNER2: 0
*** ROM Status; SEG3: UNOWNED , OWNER3: 0
*** Page 1 Certified: YES
*** Segment 1 Image: 41U9986 N44178B 41U9987 201107260918c41 Factory
40000000110000000000000000000000
*** Segment 1 Revision: 0
*** Segment 1 Hash: 0FC6 FEC1 C1E2 5058 54C1 ED4A 9915 93BD E960 C147 5033 3135 5256
9B45 0000 0000
*** Query Adapter Status successful ***
Obtain Status ended successfully!
***** Command ST ended. ---- Tue Jul  9 10:25:55 2013
```

Still in factory state: Segment 1 of the coprocessor is in the Factory state.

Loading segment 1 in to the IBM PCIe 4765 Cryptographic Coprocessor

You can use the same GNOME Terminal that was opened in “Reading the IBM PCIe 4765 Cryptographic Coprocessor status” on page 140. Ensure that you are in the `/opt/IBM/4765/clu` folder. Complete the following steps:

1. To load segment 1 with Version 4.3.5 into the IBM PCIe 4765 Cryptographic Coprocessor, run the following command:

```
./csulclu log20130709.txt PL 0 cr14.3.5.clu
```

2. You can read the coprocessor status, as described in “Reading the IBM PCIe 4765 Cryptographic Coprocessor status”, to verify that segment 1 was loaded appropriately. Example 5-2 shows the read status after loading segment 1.

Example 5-2 Read status after loading segment 1

```
CSULCLU V4.1.1.2 log20130709.txt ST 0    begun Tue Jul  9 11:02:42
2013
***** Command ST started. ---- Tue Jul  9 11:02:42 2013

*** VPD data; PartNum = 41U9986
*** VPD data; EC Num = N44178B
*** VPD data; Ser Num = 12345678
*** VPD data; Description = IBM 4765-001 PCIe Cryptographic
Coprocessor
*** VPD data; Mfg. Loc. = 91
*** ROM Status; POST0 Version 1, Release 43
*** ROM Status; MiniBoot0 Version 1, Release 20
*** ROM Status; INIT: INITIALIZED
*** ROM Status; SEG2: UNOWNED , OWNER2: 0
*** ROM Status; SEG3: UNOWNED , OWNER3: 0
*** Page 1 Certified: YES
*** Segment 1 Image: 4.3.5 E P1v060C M011D P2v0708 F5540
201208281457403A000022000000000000
*** Segment 1 Revision: 40305
*** Segment 1 Hash: 722D F07C 6C6B 4939 5FFC 5B6F 777C 5B88 A35B
F368 BB73 3F49 9164 6D49 8B9E 5107
*** Query Adapter Status successful ***
Obtain Status ended successfully!
***** Command ST ended. ---- Tue Jul  9 11:03:48 2013
```

Note: Segment 1 is not in the Factory state anymore.

Loading segments 2 and 3 in to the IBM PCIe 4765 Cryptographic Coprocessor

IBM Enterprise Key Management Foundation uses user-defined extensions, which are known as UDX, to the standard IBM 4765 firmware, which must be loaded in to segments 2 and 3 of the IBM PCIe 4765 Cryptographic Coprocessor. The extensions are placed in the `/ibm4765/firmware` folder on the installation CD.

Load the IBM Enterprise Key Management Foundation extensions by using a CLU file named `load_seg2_seg3-dkmsudx-version.clu`. For more information about which version number to use in the loading file, check the IBM Enterprise Key Management Foundation installation guide on the installation CD.

In our scenario, segment 1 of cryptographic coprocessor was installed with Version 4.3.4 and the ROM Status of segments 2 and 3 indicates OWNER: UNKNOWN. Therefore, the extensions to be loaded in to segments 2 and 3 must have Version 4.3.5.

In the following steps, you can use the same GNOME Terminal that was opened in “Reading the IBM PCIe 4765 Cryptographic Coprocessor status”. Ensure that you are in `/opt/IBM/4765/clu` folder.

1. To load the IBM Enterprise Key Management Foundation extensions Version 4.3.5 in to segments 2 and 3 of the IBM PCIe 4765 Cryptographic Coprocessor, run the following command:

```
./csulclu log20130630.txt PL 0 tmp/IBM DKMS
8.3.1/ibm4765/firmware/load_seg2_seg3-dkmsudx-4.3.5.clu
```

2. You can read the coprocessor status, as described in “Reading the IBM PCIe 4765 Cryptographic Coprocessor status”, to verify that segments 2 and 3 were loaded appropriately.

Example 5-3 Read status after loading segments 2 and 3

```
CSULCLU V4.1.2 log20130709.txt ST 0   begun Tue Jul  9 10:25:08 2013
***** Command ST started. ---- Thu Jun  6 12:02:20 2013
```

```
*** VPD data; PartNum = 41U9986
*** VPD data; EC Num = N44178B
*** VPD data; Ser Num = 12345678
*** VPD data; Description = IBM 4765-001 PCIe Cryptographic
Coprocessor
*** VPD data; Mfg. Loc. = 91
*** ROM Status; POST0 Version 1, Release 43
*** ROM Status; MiniBoot0 Version 1, Release 20
*** ROM Status; INIT: INITIALIZED
*** ROM Status; SEG2: RUNNABLE , OWNER2: 243
*** ROM Status; SEG3: RUNNABLE , OWNER3: 72
*** Page 1 Certified: YES
*** Segment 1 Image: 4.3.4 E P1v060C M011D P2v0708 F5540
201207261613403A000022000000000000
*** Segment 1 Revision: 40304
*** Segment 1 Hash: C97D 2F06 B7A7 FOAF F458 C6DF 122F 34E1 E884 4429
741B 58AB 2C16 4C2F C932 D926
```

```
*** Segment 2 Image: 4.3.5    y4_13-lnx-2012-03-02-21
201208281509403A000000000305030500
*** Segment 2 Revision: 40305
*** Segment 2 Hash: 5EA8 9396 A42C 74DB 3664 9C1F 3622 C418 435E 88FF
D574 C38A 5132 0322 DCFD BE2C
*** Segment 3 Image: UDX 0305 DKMS 4.3.3, CCA 43543
2013041615204305
*** Segment 3 Revision: 4
*** Segment 3 Hash: E22A 331C E047 BB53 6FE7 9DC5 E38D BEC9 7E35 9CA4
8B01 DF94 D95D 85C7 4CC8 5D0B
*** Query Adapter Status successful ***
Obtain Status ended successfully!
***** Command ST ended. ---- Thu Jun  6 12:03:02 2013
```

Note: The ROM status of segment 2 is OWNER: 243, and the ROM status of segment 3 is OWNER: 72.

Further SLES11 related configurations

Here are a few related configuration items.

Users

Changes in the user or group configuration should be done carefully, as these changes can influence the access rights of the users to the programs that are installed.

Space

There are some specific folders that are used while the installation is done, such as /tmp, /opt, and /var. Check that these folders have enough space and eventually increase the space capacity.

Access rights

To configure access rights, complete the following steps:

1. Use a USB stick.

If you plan to export or import keys encrypted from and into your IBM Enterprise Key Management Foundation key repository, you eventually will use a USB stick for exchanging the export / import files. By default, SLES asks you to enter the root user's password to use the USB stick. Click **Remember authorization** to allow the dkms user to later use the USB stick without needing the root password.

2. Verify and eventually change the owner and access rights of the keystore files in the /opt/IBM/4765 directory, as shown in Example 5-4, by completing the following steps.

Example 5-4 Result of the "ls -al" command before setting the access rights

```
sles11sp1:~ # cd /opt/IBM/4765/keys
sles11sp1:/opt/IBM/4765/keys # ls -la
total 80
drwsrws--- 5 root users 4096 2013-07-03 12:47 .
drwxr-xr-x 11 root users 4096 2012-03-27 15:58 ..
-rw-r----- 1 root users 384 2013-05-21 08:53 aes.key
-rw-r----- 1 root users 164 2013-05-21 08:53 aes.key.NDX
drwsrws--- 2 root users 4096 2013-05-21 08:53 aeslist
-rw-rw---- 1 root users 4992 2013-07-03 12:47 des.key
-rwxrwx--- 1 root vboxsf 1988 2013-07-03 12:47 des.key.NDX
drwsrws--- 2 root users 12288 2013-05-21 08:53 deslist
-rw-r----- 1 root users 16784 2013-05-21 08:53 pka.key
-rw-r----- 1 root users 1532 2013-05-21 08:53 pka.key.NDX
drwsrws--- 2 root users 12288 2013-05-21 10:50 pkalist
sles11sp1:/opt/IBM/4765/keys #
```

- a. To move to the /opt/IBM/4765/keys folder, enter the following command:

```
cd /opt/IBM/4765/keys
```

- b. To show the file names of the folder keys and their related information, like the owner and access rights, run the following command:

```
ls -al
```

The result is shown in Example 5-4.

The owner is displayed in the third and fourth column in the format `username usergroupname` and should be set to `root users`.

The file access rights are shown in the first column in the format `rwXrwxrwx` or `rwsrwsrws` (if a sticky bit is used¹¹), where missing access is replaced by a "-" (dash). The access right should show either `rwXrwx---` or `rwsrws---`.

- c. To change the owner of all files, move to the parent folder /opt/IBM/4765 by running `cd` and then run the following command:

```
chown root:users -R keys
```

- d. To change the access rights for the files in the /opt/IBM/4765/keys folder, run the following command:

```
chmod 770 -R keys
```

¹¹ For more information about using the sticky bit, see the SLES administration manuals.

- e. To set the sticky bit at the group and user levels for the files in the /opt/IBM/4765/keys folder, run the following commands:
- ```
chmod g+s -R keys
chmod u+s -R keys
```
- f. To verify the owner and access right of the files in the keys folder, move to the keys folder and run the following command:

```
ls -al
```

Example 5-5 shows the results of the command.

*Example 5-5 Result of the "ls -al" command after setting the access rights*

---

```
sles11sp1:/opt/IBM/4765/keys # cd ..
sles11sp1:/opt/IBM/4765 # chown root:users -R keys
sles11sp1:/opt/IBM/4765 # chmod 770 -R keys
sles11sp1:/opt/IBM/4765 # chmod g+s -R keys
sles11sp1:/opt/IBM/4765 # chmod u+s -R keys
sles11sp1:/opt/IBM/4765 # cd keys
sles11sp1:/opt/IBM/4765/keys # ls -la
total 80
drwsrws--- 5 root users 4096 2013-07-03 12:47 .
drwxr-xr-x 11 root users 4096 2012-03-27 15:58 ..
-rwsrws--- 1 root users 384 2013-05-21 08:53 aes.key
-rwsrws--- 1 root users 164 2013-05-21 08:53 aes.key.NDX
drwsrws--- 2 root users 4096 2013-05-21 08:53 aeslist
-rwsrws--- 1 root users 4992 2013-07-03 12:47 des.key
-rwsrws--- 1 root users 1988 2013-07-03 12:47 des.key.NDX
drwsrws--- 2 root users 12288 2013-05-21 08:53 deslist
-rwsrws--- 1 root users 16784 2013-05-21 08:53 pka.key
-rwsrws--- 1 root users 1532 2013-05-21 08:53 pka.key.NDX
drwsrws--- 2 root users 12288 2013-05-21 10:50 pkaclist
sles11sp1:/opt/IBM/4765/keys #
```

---

## Installing the Key Management Workstation software

Before you can install the software, you must look at some preliminary steps.

### **Preliminary steps**

The installation software program is available on the IBM Enterprise Key Management Foundation CD as `dkmsws/install.sh`.

Make sure that the IBM Enterprise Key Management Foundation license code file `features.dat` is available before running the software installer.

As your Key Management Workstation will be set up to work in *online* mode and connect to a DB2 key repository database on the mainframe, make sure that the DB2 Connect license file `db2jcc_license_cisuz.jar` is also available.

Your smart card readers must be connected to the Key Management Workstation by now.

### ***Installation***

Proceed with the next steps to install the Key Management Workstation software, but do *not* start the software after the installation, as other steps must be done first:

1. Insert the installation CD in the Key Management Workstation CD drive and verify the expected software version with the version label on the CD.
2. Log on to the graphical desktop environment if you are not already logged on through the root user. Do not use the `su` command to start the installation executable file, as the access rights of this command are not sufficient.
3. Click the DVD on the desktop, open the `dkmsws` folder, and double-click the `dkms-installer.sh` file.
4. Follow the instructions that are given by the installer:
  - Read the instructions carefully and agree to the license terms.
  - Enter a new environment name, depending on your naming convention, for example, `test` for a test environment or `prod` for a production environment, and click **Add** and then **Next**.
  - Select the feature file `/tmp/dkms/features.dat` and click **Next**.

**Adding licensed features:** You can update your DKMS environment with additional licensed features later by copying the `features.dat` file to the `/var/opt/ibm/dkms/environment-name/table` folder of each `environment-name` environment being upgraded.

- If you are asked to install support for smart cards, click **Yes** and then click **Next**.
- Click **Yes** to perform the system hardening and click **Next**.

**Basic security system hardening:** If chosen, the installer performs a limited amount of security hardening, such as disabling the network system services (in particular, the ssh server, the network file system service, and the rpcbind service (sshd, nfs, and rpcbind)), and mounts the print spooler in the system's volatile memory.

The hardening ensures that a system that is installed with the default options presents a closed firewall allowing no incoming connections. For production systems, it is important to preserve this state, as any remote access might compromise the security of the Key Management Workstation.

System hardening at this step makes sense only if the printer is installed.

- Select the DB2 Connect license file `db2jcc_license_cisuz.jar` from the `/tmp/dkms` folder and click **Next**.

**Licenses:** The `db2jcc_license_cisuz.jar` file is not needed for working with UKDS2 keys. If you do not choose a license file now, you can copy the file later into the current EKMF version installations folder:

```
/opt/ibm/dkms/current-version-number/bin
```

- Click **Install** to start the installation.
5. Check the installation folders:
    - The IBM Enterprise Key Management Foundation software is installed in the `/opt/ibm/dkms/current-version-number` folder.
    - All the configuration data pertaining to the environment entered in step 4 on page 147 is placed in the `/var/opt/ibm/dkms/environment-name` folder.
  6. When the installer finishes, reboot the system to complete the installation.

**Be patient:** Some System x hardware that is used for the IBM Enterprise Key Management Foundation might require that you wait a few minutes until the physically installed IBM PCIe 4765 Cryptographic Coprocessor also ends its start routine.

### ***Verifying the application files***

The IBM Enterprise Key Management Foundation installation CD contains a file with a verification code for the application files. This verification code can be used to check for modified program files. The verification codes are in SHA-1 format and can be checked with a built-in SLES tool.

To verify the files, complete the following steps:

1. Open an instance of GNOME Terminal.
2. Change the directory to the new installation by running the following command:

```
cd /opt/ibm/dkms/version-number
```

3. Run the verification by running the following command:

```
sha1sum -c DKMS_CD/dkmsws/dkms-version-number.sha
```

4. If any checks fail, the failures are noted in the last lines of the output.

### **IBM 4765 management**

The next sections contain the basic installation and configuration steps for the IBM 4765 using the CNM Utility and the SCUP, which need to be done before the Key Management application is started.

#### ***Initial setup of the IBM PCIe 4765 Cryptographic Coprocessor***

The following steps must be done for an initial setup of the IBM PCIe 4765 Cryptographic Coprocessor:

1. Initialize IBM 4765.
2. Load the function control vectors (FCVs).
3. Synchronize / set the time of the IBM 4765 with the local system time.
4. Set a random DES master key.
5. Initialize the DES, PKA, and AES keystores.

These steps can be done either manually through the CNM application or automatically through the CCA test initialization utility (CCA Init). Both applications are supplied as a part of the CCA 4.2 installation. For more information about these steps, see “IBM 4765 initialization” on page 298.

The CCA Init application created a profile TESTER in the IBM 4765, which has the DEFAULT role assigned. Now, the DEFAULT role is permitted to run all possible operations with the IBM 4765. As part of the initialization, a split passphrase profile “CNMADMIN” is created and the TESTER profile is removed. The allowed operations in the DEFAULT role are restricted to a limited set, so it is now ready for secure operations.

Use the CNMADMIN profile for the remaining setup tasks.

**Note:** Split the passphrase in to two parts and assign them to two administrators. The first part is the left half part and must be chosen, entered, and kept secret by the first administrator. The second part is the right half part, which must be chosen, entered, and kept secret by the second administrator. This situation is necessary until the ADMIN smart card group-of-group profile is created and can be used, as described in “Smart card management”.

### **Smart card management**

Use the SCUP to set up your IBM Enterprise Key Management Foundation environment to use smart cards.

To log on to the SCUP and create the CA and TKE card that are needed for later use, complete the following steps:

1. On the taskbar, click **Computer** → **More Applications** → **SCUP** to start the SCUP.
2. In Figure 5-7, select the profile ID to log on and click **OK**.



Figure 5-7 SCUP - logon window

3. In Figure 5-8, enter the passphrase for the selected profile ID and click **OK**.



Figure 5-8 SCUP - logon window - enter password for user ID

4. Create and back up the CA smart cards:

- To create a CA card, complete the steps that are described in “Initializing and personalizing a certificate authority smart card” on page 248.
- To create a backup of the CA card, complete the steps that are described in “Backup CA smart card” on page 256.

**Note:** You should create at least one backup of the CA smart card. For disaster recovery, create three additional backups, so that in the end you have four CA smart cards. You should keep two CA cards in the same safe and the other two CA cards in a different, geographically separate safe.

Use always the last created backup card for creating a backup card, which ensures that all the CA backup cards are fully functional.

5. To enroll the IBM 4765, complete the steps that are described in “Enrolling the IBM PCIe 4765 Cryptographic Coprocessor” on page 262.

6. Create the TKE smart cards:

- To initialize and enroll a TKE card, complete the steps that are described in “Initializing and enrolling a TKE smart card” on page 267.

**Initializing and enrolling the TKE card:** During this step, also consider initializing and enrolling the TKE card that you plan to use as backup. Initialize and enroll three additional cards to use as backups for each TKE card per user so that you have four TKE cards for each user.

- To personalize a TKE card, complete the steps that are described in “Personalizing a TKE smart card” on page 272.

**Note:** Repeat the personalization steps for all the backup TKE cards that you plan to use.

Use the CNM Utility to create the smart card roles and profiles / users to generate the IBM 4765 logon keys for these users, and to save and manage the IBM 4765 master keys, as described in the next sections.

### ***Roles and profiles management***

After the initialization of the IBM 4765 is complete, as described in “Initial setup of the IBM PCIe 4765 Cryptographic Coprocessor” on page 149, the initial TESTER passphrase profile no longer exist and the DEFAULT role is now restricted to a minimum set of access control points.

At this point any configuration change on the IBM 4765 can only be performed through the use of the CNMADMIN passphrase profile — that enforce dual control by using a two-part passphrase separated into a first part and a last part — that is associated with the CNMADMIN role that have all access control points enabled.

In our example, we use the CNMADMIN passphrase profile to log on to the IBM 4765 and create the necessary roles and profiles<sup>12</sup> by completing the following steps:

1. To create all the necessary roles (ADMIN, SO1, SO2, and MANAGER), repeat the steps that are described in “Using the CNM Utility to create, edit, or delete a role” on page 350 for each role.

Figure 5-9 shows a list of the roles of Fictional Bank.

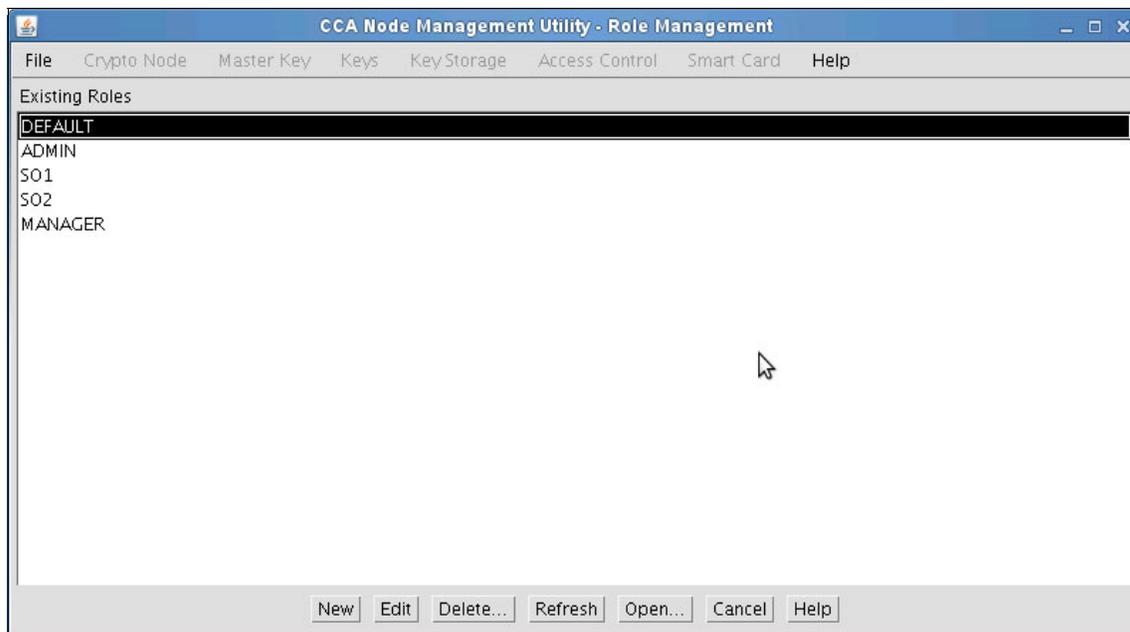


Figure 5-9 CNM - existing roles of Fictional Bank

<sup>12</sup> Profiles can be either users or group of users, that, depending on your setup, can log on to the IBM 4765 on a passphrase base or through smart cards. In our example, the bank is using smart card profiles.

2. To create all the smart card profiles that are needed, repeat the following steps for each profile:
  - Generate a logon key for the ADM11, ADM12, ADM21, ADM22, SO11, SO12, SO21, and SO22 profiles, as described in “Generating an IBM 4765 logon key on TKE smart card” on page 306.
  - Create the smart card profiles ADM11, ADM12, ADM21, ADM22, SO11, SO12, SO21, and SO22, as described in “Using the CNM Utility to create a smart card profile” on page 352.
  - To copy the contents and provide a backup of the TKE card that contains the IBM 4765 logon key generated previously, follow the steps that are described in “Backing up a TKE smart card” on page 312.
3. To create the smart card group profiles ADM1, ADM2, SO1, and SO2, repeat the steps that are described in “Using the CNM Utility to create a smart card group profile” on page 354 for each smart card group profile.
4. To create the smart card group-of-group profiles ADMIN and MANAGER, repeat the steps that are described in “Using the CNM Utility to create a group of groups profile” on page 356 for each smart card group-of-group profile.

Figure 5-10 shows a list of the existing passphrase and smart card profiles, group profiles, and group-of-group profiles of Fictional Bank.

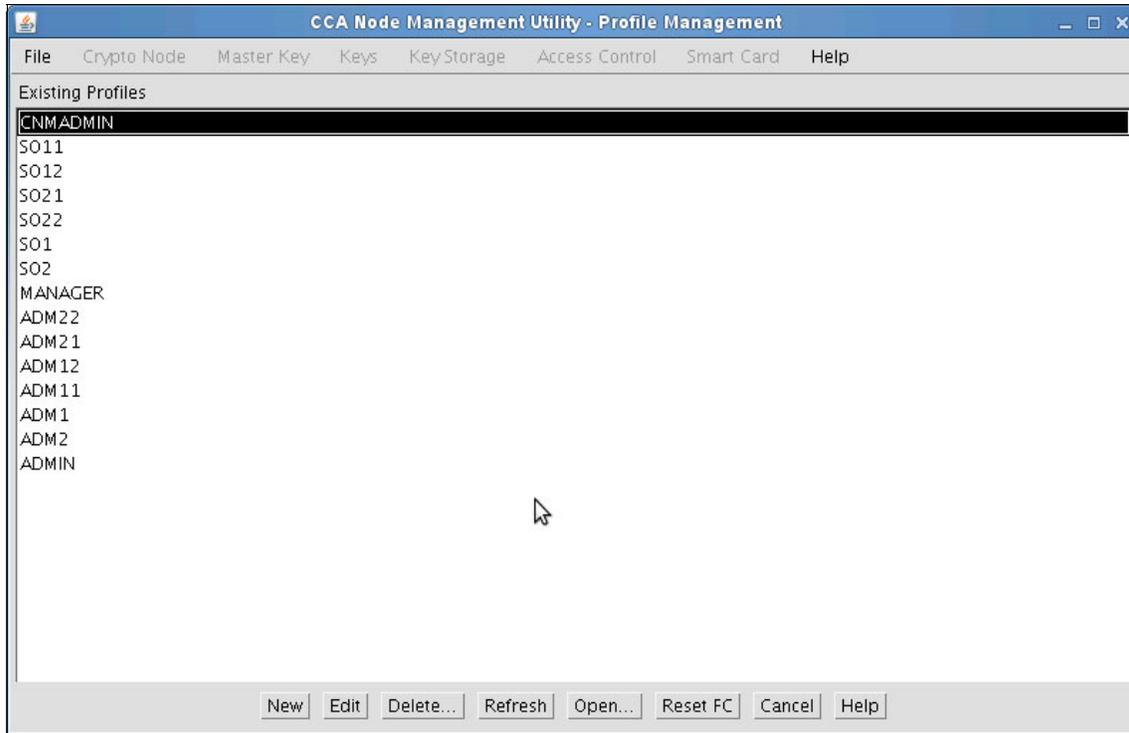


Figure 5-10 CNM - existing profiles of Fictional Bank

**Note:** You must create at least one ADMIN profile or group profile that has administrator permissions because this user is required for the initial configuration of the Key Management application.

### ***Master key management***

Depending on which keys you want to manage with the Key Management Workstation, you might need to generate and load the DES/PKA, AES, and ECC master keys.

In our example, we describe only the steps of generating and loading the DES/PKA master key using smart cards. Complete the following steps:

1. To generate the DES/PKA master key on the smart cards, complete the steps that are described in "Generating an IBM 4765 DES/PKA master key" on page 317. Use the TKE smart cards that you created earlier.

2. To load the DES/PKA master key that is generated in step 1 on page 154 from the smart cards, complete the steps that are described in “Loading an IBM 4765 DES/PKA master key” on page 326. The master key now is present in the new master key register.
3. To set the DES/PKA master key to the current master key register and activate it for further use, complete the steps that are described in “Setting the IBM 4765 DES/PKA master keys and re-enciphering the key storage” on page 334.

## **Starting and configuring the Key Management Workstation software**

Before you start the DKMS application for the first time, further configuration files must be prepared, as described in “Configuring a new DKMS environment”.

### ***Configuring a new DKMS environment***

For the bank scenario that is based on a remote setup of the DB2 key repository on the mainframe, you must provide additional configuration data for the new environment that is created at the installation time of the IBM Enterprise Key Management Foundation application.

This configuration data is either received from IBM or is on the installation CD in the `config` folder. This folder contains several subfolders containing configuration data for different licensed features.

The `/config/Basic Host with ICSF` subfolder contains the basic configuration files, which are identified by the file extension `.del`. These files must be copied to the `/var/opt/ibm/dkms/environment-name/table` folder of each installed IBM Enterprise Key Management Foundation environment in which they are required.

Also, copy the contents of the `/config/Basic Host with ICSF/form` subfolder in to the `/var/opt/ibm/dkms/environment-name/form` folder of each new IBM Enterprise Key Management Foundation environment.

When IBM Enterprise Key Management Foundation runs the first time, an initialization procedure loads the configuration data that is present in the `/var/opt/ibm/dkms/environment-name/table` folder in to Key Management Workstation configuration.

The different configurations of the purchasable features of IBM Enterprise Key Management Foundation are placed in separate folders and identified with their prefixed feature number. The subfolders contain different file types, as shown in Table 5-5.

*Table 5-5 Import files on the IBM Enterprise Key Management Foundation installation CD*

| <b>File extension</b> | <b>Description</b>                                                                                                                                                                                                                                      |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .elt                  | Files containing element table data have this extension. These files are imported in to IBM Enterprise Key Management Foundation through the CONF0055 Element Table Maintenance menu, as described in “Importing the element table entries” on page 168 |
| .kdt                  | Files containing key definition data have this extension. These files are imported into IBM Enterprise Key Management Foundation through the CONF0041 Key Templates menu.                                                                               |

### ***Starting the Key Management application***

Log on to SLES as the *dkms* user. The first time that the DKMS application is started, an initialization procedure runs and loads the copied configuration data and licensed features into the Key Management application configuration database. As the configuration data is unique for an environment, you must repeat the configuration steps for every additional environment that is hosted on the EKMF Workstation. Complete the following steps:

1. On the taskbar, click **Computer** → **More Applications** → **DKMS <version> <environment-name>** to start the Key Management application.

2. Click **Start** to display the list of suitable profiles and group profiles that are available<sup>13</sup> on the IBM 4765 that can be used to log on to the Key Management application, as shown in Figure 5-11.

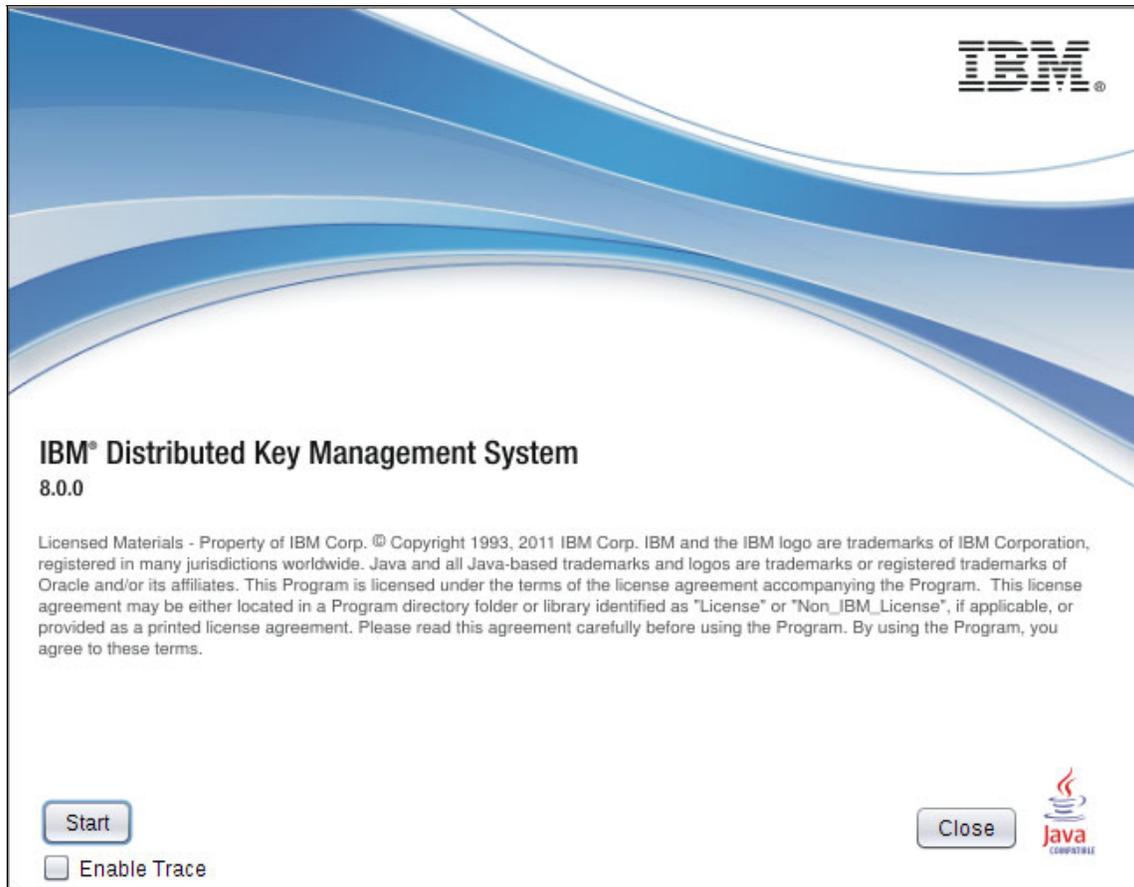


Figure 5-11 IBM Enterprise Key Management Foundation - Start window

<sup>13</sup> At least one profile or group profile called ADMIN must be available for logon to the Key Management application.

3. The initial logon must be done by the ADMIN profile. Select the ADMIN profile, as shown in Figure 5-12, and click **OK**.



Figure 5-12 Logon window

- As this ADMIN profile is in our group-of-group profile in our example, select which profile from the first group member ADM1 of the group ADMIN should log on, as shown in Figure 5-13, and click **OK**.

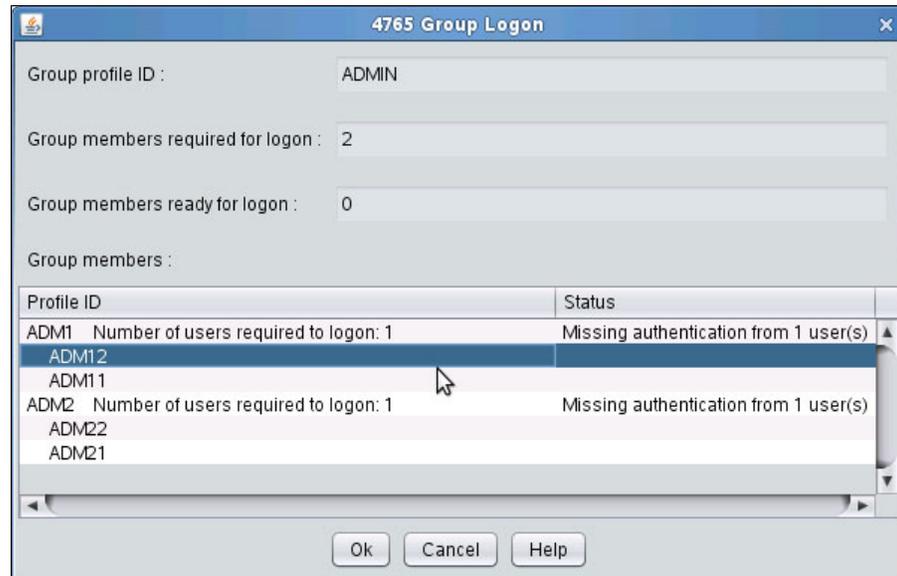


Figure 5-13 Group-of-group logon of the first group

- Insert the smart card that is associated with the selected profile and click **OK**, as shown in Figure 5-14.



Figure 5-14 Group-of-group logon - insert smart card

6. Enter the PIN for the smart card on the smart card reader, as shown in Figure 5-15.



Figure 5-15 Group-of-group logon - enter PIN on the smart card reader

7. After the first group is logged on, its status changes to ready for logon, as shown in Figure 5-16. Select which profile from the second group member ADM2 of the group ADMIN should log on next and proceed with the logon as described for the first group.



Figure 5-16 Group-of-group logon of the second group

8. The information that is contained in the \*.del files that is placed in to the /var/opt/ibm/dkms/environment-name/table folder, as described in “Configuring a new DKMS environment” on page 155, is used to create and populate the configuration tables. Not all entries are required to have data imported; the window that is shown in Figure 5-17 is an example of the most common setup for a new DKMS installation.

Click **Install** to import the configuration data from the \*.del files in the /var/opt/ibm/dkms/environment-name/table folder.

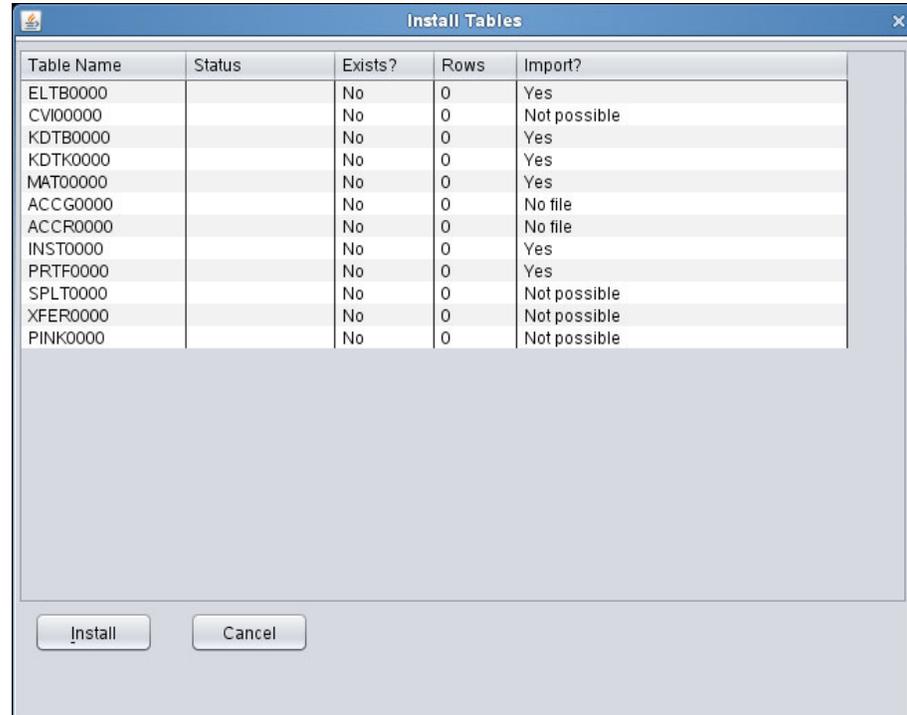


Figure 5-17 Start EKMF for the first time - Install Tables

9. Figure 5-18 shows an information message about the successful completion of the import. Click **OK** to proceed.



Figure 5-18 Start EKMF for the first time - tables successfully installed

### **Configuring the host network settings**

After you install the configuration tables successfully, use the IBM Enterprise Key Management Foundation host network settings window to configure the connection between the Key Management Workstation and the primary IBM Enterprise Key Management Foundation Agent on the mainframe.

**Note:** When the Key Management Workstation connects, it does so over a session that is secured with a key that is stored in the ICSF CKDS. This security is referred to as link encryption. The key that is used for link encryption cannot be established at this point, so link encryption must be disabled on both the workstation and z/OS, and not enabled until the key is established (see 5.4, “Link encryption configuration” on page 216).

Link encryption is configurable at the z/OS image by using the RACF FACILITY class profile CRYPTO.DKMS.LNKCRYOFF. To allow disabling link encryption, you must grant the z/OS Agent started task user READ level permission to this profile. In addition, you must set the **&KEY-EXCHANGE** parameter in the IBM Enterprise Key Management Foundation Agent option data set to **NONE**.

Complete the following steps:

1. In Figure 5-19 on page 163, complete all the necessary fields with the following information.

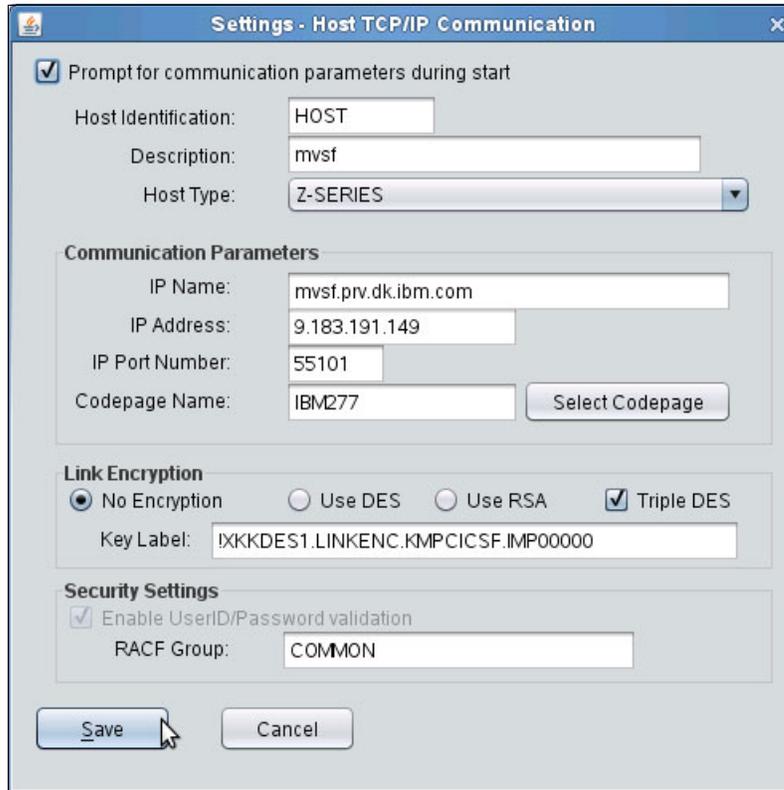


Figure 5-19 Host TCP/IP communication settings

**Prompt for communication parameters during start**

Select this check box only if you want to view or edit the next parameters at each EKMF start.

**Host Identification**

Enter an identifier to use for the primary EKMF DB2 host, for example, the short name of the LPAR that is running the EKMF Agent.

**Description**

Enter an optional host description.

**Host Type**

Select **Z-SERIES** from the Host Type item list.

**IP Name**

Enter the host name of the primary EKMF Agent.

**IP Address**

Enter the IP address of the primary EKMF Agent. If an IP name is entered, EKMF attempts to look up the IP address automatically.

**IP Port Number**

Enter the port number of the primary EKMF Agent.

|                          |                                                                                                                                                                                                                      |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Codepage Name</b>     | Enter a value for the code page name or click <b>Select Codepage</b> and select one from the list. The code page is used to translate text that is transferred between the Key Management Workstation and the Agent. |
| <b>Link Encryption</b>   | The link encryption setting remains as the initial No Encryption setting until it can be set up later.                                                                                                               |
| <b>Security Settings</b> | Enter a RACF Group name in to the RACF Group field to enable grouping of EKMF host tasks that share the same RACF database. This setting is used to allow single sign-on to multiple EKMF host tasks.                |

**Note:** The RACF Group name in this field does not have to be the name of a RACF group in the RACF database.

2. Click **Save** to finish the network configuration.
3. Because link encryption is not established, a warning message is issued, as shown in Figure 5-20.



Figure 5-20 Action message noting that link encryption is not active

Click **Yes** to continue to the Key Management Workstation main menu.

4. The application attempts to connect to the specified host. On successful connection to an IBM Enterprise Key Management Foundation host, a logon window opens, as shown in Figure 5-21 on page 165. Enter your TSO user ID and password in the **Userid** and **Password** fields and click **Logon**.

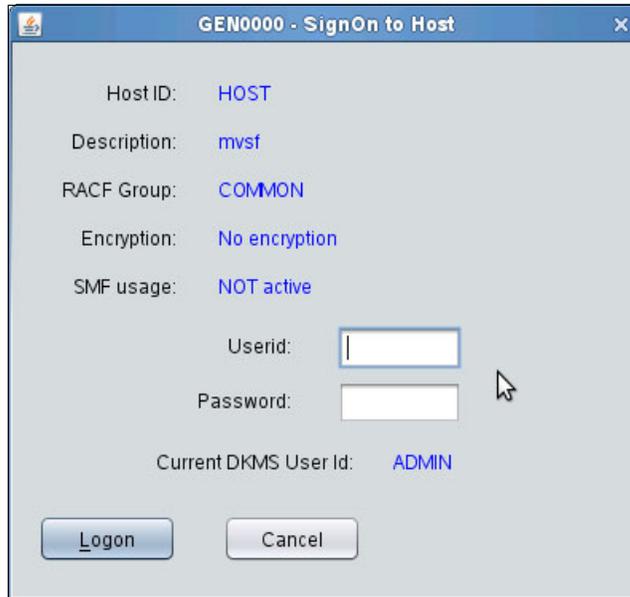


Figure 5-21 Log on to the host

### ***Configuring the workstation's miscellaneous parameters***

The next step is to set the basic configuration for the DKMS application. For our example remote key repository setup, we start the IBM Enterprise Key Management Foundation application without any configuration parameters, so it shows an error message.

Complete the following steps:

1. Accept the error message and continue to the configuration window that is shown in Figure 5-22.

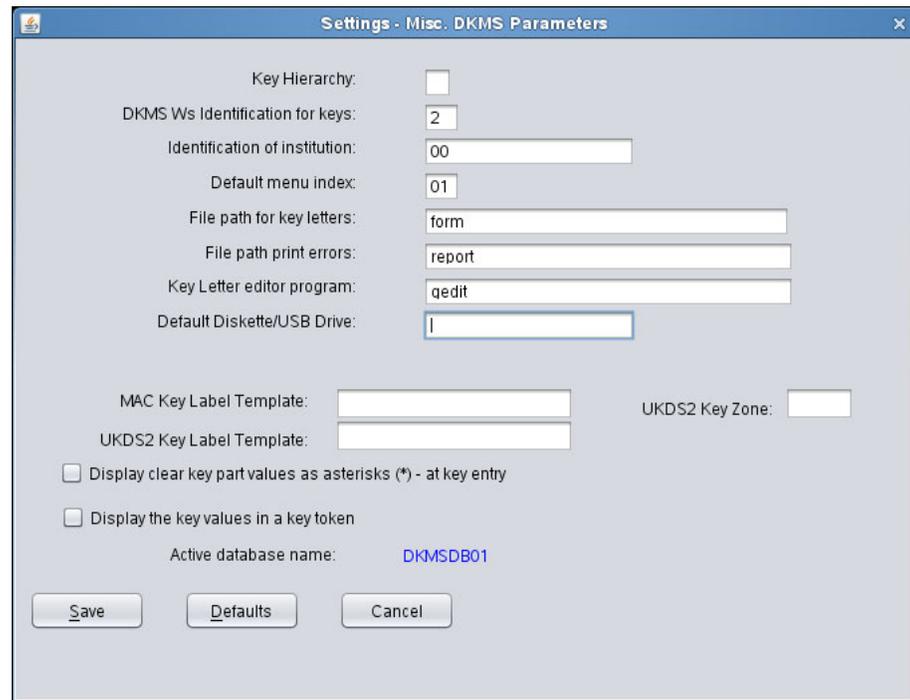


Figure 5-22 Key Management Workstation miscellaneous parameters

2. Complete the necessary fields with the following information:

**Key Hierarchy**

Enter a single character value that will be inserted into the labels of generated keys if the key label in the key template contains the tag <hierarchy>. For example, enter T for the key hierarchies of a test environment.

**DKMS Ws Identification for keys**

Enter the value 2, which corresponds to a standard Key Management Workstation.

**Identification of institution**

Enter the Institution Id that is defined in the Institution Table Maintenance for Fictional Bank. The default value is 00.

**Default menu index** Enter the top menu hierarchy index number 99 so that the Key Management Workstation autobuild menu opens after you start the application.<sup>14</sup>

**File path for key letters**

Enter the file path to look for key letters. The default is form, which is a subfolder of the current environment `/var/opt/ibm/dkms/environment-name/`.

**File path for print errors**

Enter the file path to place printer error reports. The default is report, which is a subfolder of the current environment `/var/opt/ibm/dkms/environment-name/`.

**Key letter editor program**

Enter the name of editor that is used for Key letter editing. The default editor is set to gedit (Gnome Editor).

**Default USB drive**

As USB sticks are automatically mounted on `/media/usb-label`, where `usb-label` is the label that is given to the USB stick, enter the path `/media/usb-label`.

**MAC Key Label Template**

Ignore this setting because it is used with UKDS2 keys only.

**UKDS2 Key Zone**

Ignore this setting because it is used with UKDS2 keys only.

**UKDS2 Key Label Template**

Ignore this setting because it is used with UKDS2 keys only.

**Display clear key part values as asterisks (\*) - at key entry**

Select this check box in a production environment to show asterisks instead of the key values when entering keys.

**Display the key values in a key token**

Clear this check box in a production environment to show asterisks instead of the encrypted key value in the key token.

---

<sup>14</sup> In our setup, we use the IBM Enterprise Key Management Foundation autobuild menu, as this menu cannot be changed by the user. After the Key Management Workstation is configured, you can create and assign your own menu structure at EKMF start through the CONF0033 program from the autobuild menu.

3. Click **Save** to proceed. IBM Enterprise Key Management Foundation starts and check the integrity values in the configuration data. Because this is the first run time, no integrity values exist. Accept the warnings and accept DKMS calculating the new values.

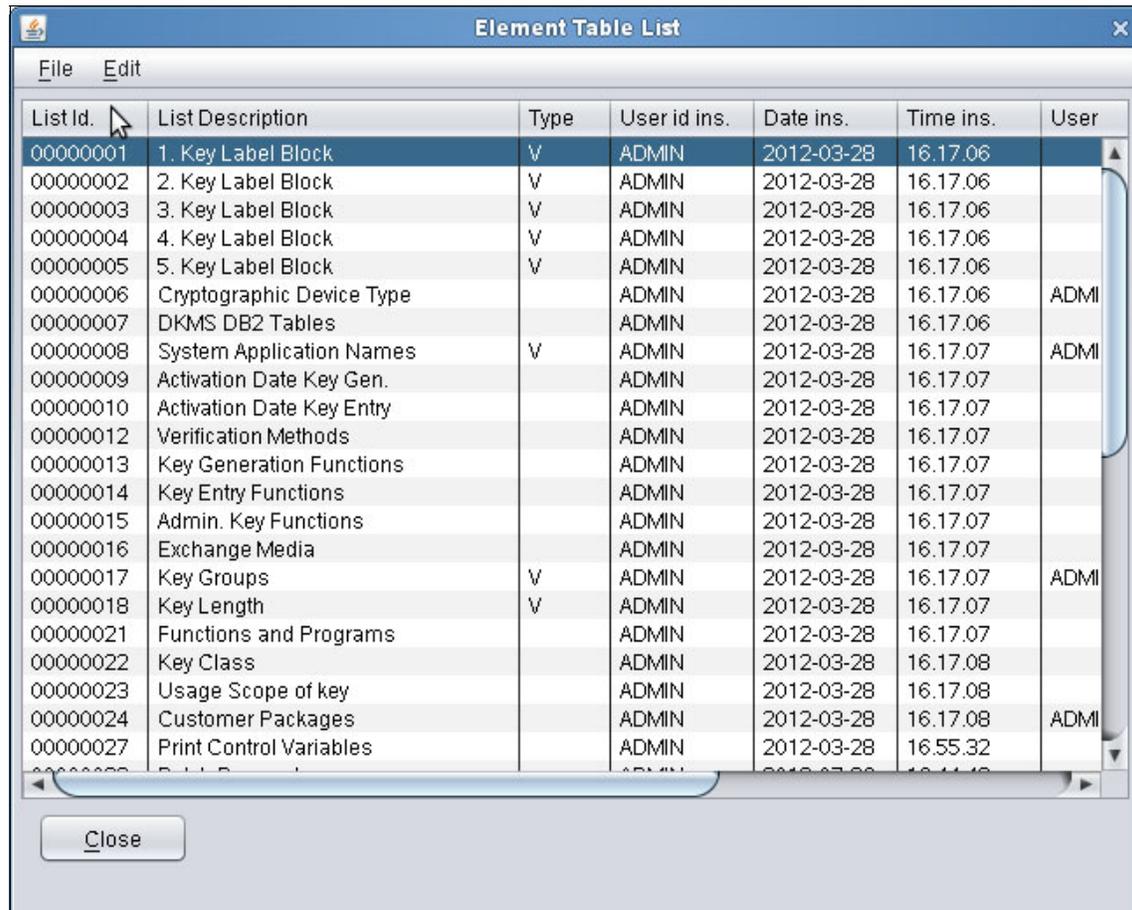
### ***Importing the element table entries***

For our basic setup in our example, we must import several files from the EKMF installation media/DVD, which contain the necessary configuration for the element table:

- ▶ `/config/Basic Host with ICFS/Basic Host with ICSF.elt`  
This file contains the basic configuration for using ICSF.
- ▶ `/config/0323 Symmetric Key Management/KEYTEMPLATEMANAGEMENT.elt` and `/config/0323 Symmetric Key Management/SYMMETRICKEYMANAGEMENT.elt`  
These files contain the basic configuration for using the symmetric key management.
- ▶ `/config/DKMS Dashboard/DASHBOARD.elt`  
This file contains the basic configuration for using the PROG0601 Dashboard.
- ▶ `/config/0070 Policies/POLICIES.elt`  
This file contains the basic configuration for using the CONF0070 System Policies.

Complete the following steps for each of these files:

1. From the autobuild menu of the DKMS application, start the **CONF0055 Element Table Maintenance** program. Figure 5-23 shows the contents of the element table list.



The screenshot shows a window titled "Element Table List" with a menu bar containing "File" and "Edit". The window contains a table with the following columns: List Id., List Description, Type, User id ins., Date ins., Time ins., and User. The table lists various system elements, including Key Label Blocks, Cryptographic Device Type, DKMS DB2 Tables, System Application Names, Activation Date Key Gen., Activation Date Key Entry, Verification Methods, Key Generation Functions, Key Entry Functions, Admin. Key Functions, Exchange Media, Key Groups, Key Length, Functions and Programs, Key Class, Usage Scope of key, Customer Packages, and Print Control Variables. The "User" column for many entries is "ADMIN", and for some, it is "ADM".

| List Id. | List Description          | Type | User id ins. | Date ins.  | Time ins. | User  |
|----------|---------------------------|------|--------------|------------|-----------|-------|
| 00000001 | 1. Key Label Block        | V    | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000002 | 2. Key Label Block        | V    | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000003 | 3. Key Label Block        | V    | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000004 | 4. Key Label Block        | V    | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000005 | 5. Key Label Block        | V    | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000006 | Cryptographic Device Type |      | ADMIN        | 2012-03-28 | 16.17.06  | ADMIN |
| 00000007 | DKMS DB2 Tables           |      | ADMIN        | 2012-03-28 | 16.17.06  |       |
| 00000008 | System Application Names  | V    | ADMIN        | 2012-03-28 | 16.17.07  | ADMIN |
| 00000009 | Activation Date Key Gen.  |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000010 | Activation Date Key Entry |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000012 | Verification Methods      |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000013 | Key Generation Functions  |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000014 | Key Entry Functions       |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000015 | Admin. Key Functions      |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000016 | Exchange Media            |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000017 | Key Groups                | V    | ADMIN        | 2012-03-28 | 16.17.07  | ADMIN |
| 00000018 | Key Length                | V    | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000021 | Functions and Programs    |      | ADMIN        | 2012-03-28 | 16.17.07  |       |
| 00000022 | Key Class                 |      | ADMIN        | 2012-03-28 | 16.17.08  |       |
| 00000023 | Usage Scope of key        |      | ADMIN        | 2012-03-28 | 16.17.08  |       |
| 00000024 | Customer Packages         |      | ADMIN        | 2012-03-28 | 16.17.08  | ADMIN |
| 00000027 | Print Control Variables   |      | ADMIN        | 2012-03-28 | 16.55.32  |       |

Figure 5-23 Element table maintenance

2. To import the Element Table Entries, click **File** → **Import** and select the first file that is listed from the specified subfolder on the installation CD, as shown in Figure 5-24. Click **Open** to import the selected file.

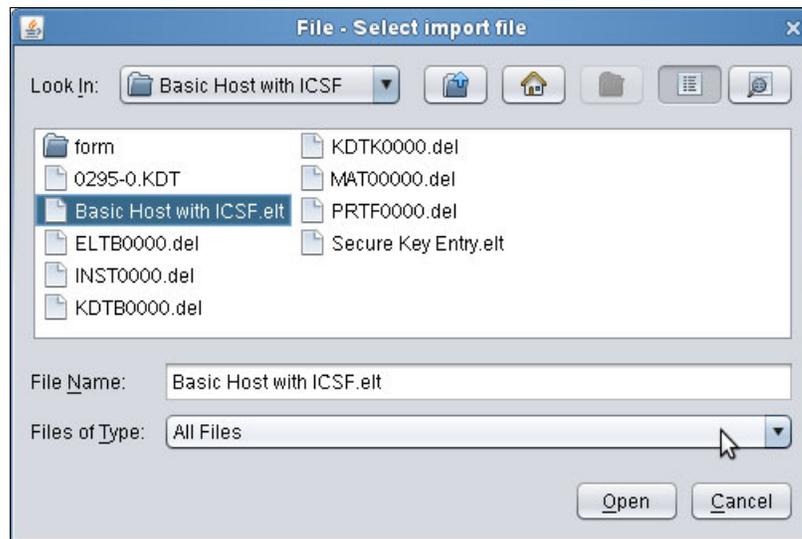


Figure 5-24 Element table - open file

- Figure 5-25 shows all the elements in the import file. The first column shows the status of these elements. Only the elements with status New must be imported.

Click **Remove Duplicates from List**, select all the remaining elements, and click **Install** to import them into the element table.

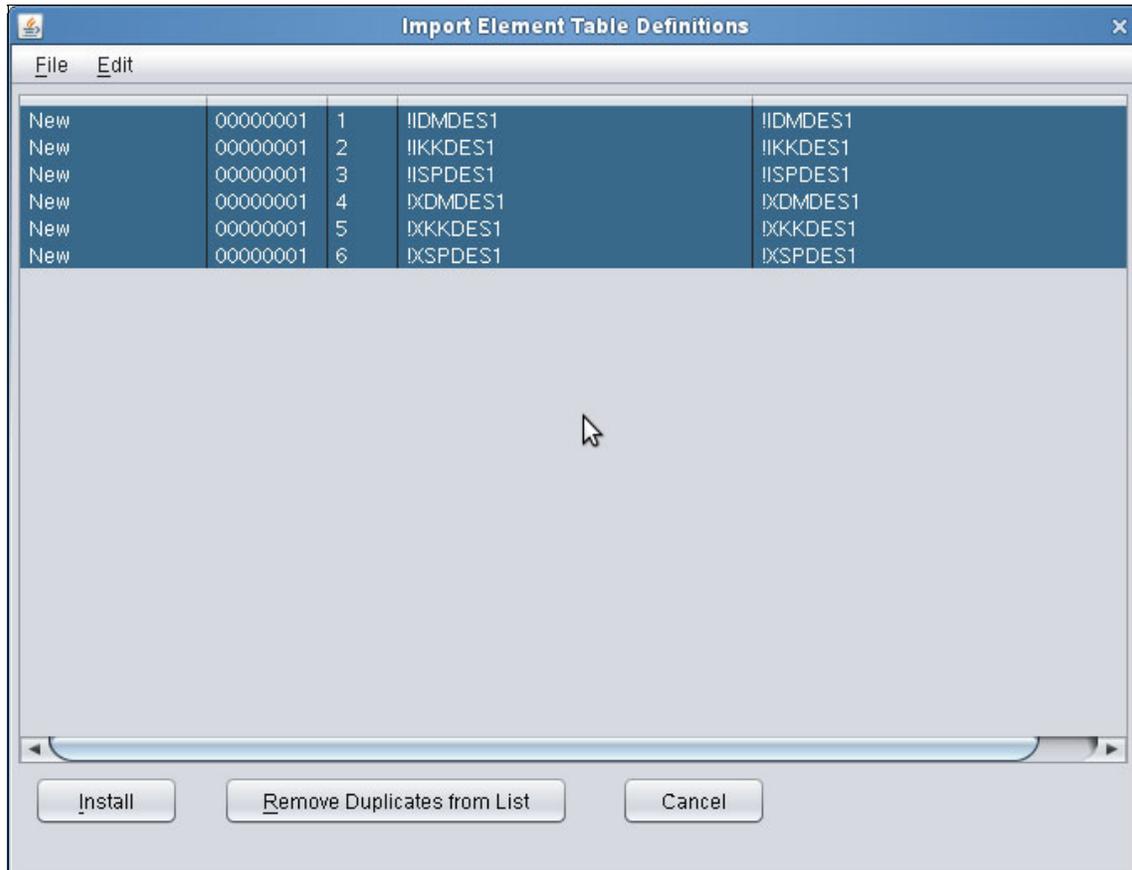


Figure 5-25 Element table - import elements

- After the import, the first column in Figure 5-26 shows a status of OK. Click **Cancel** to close the import window.

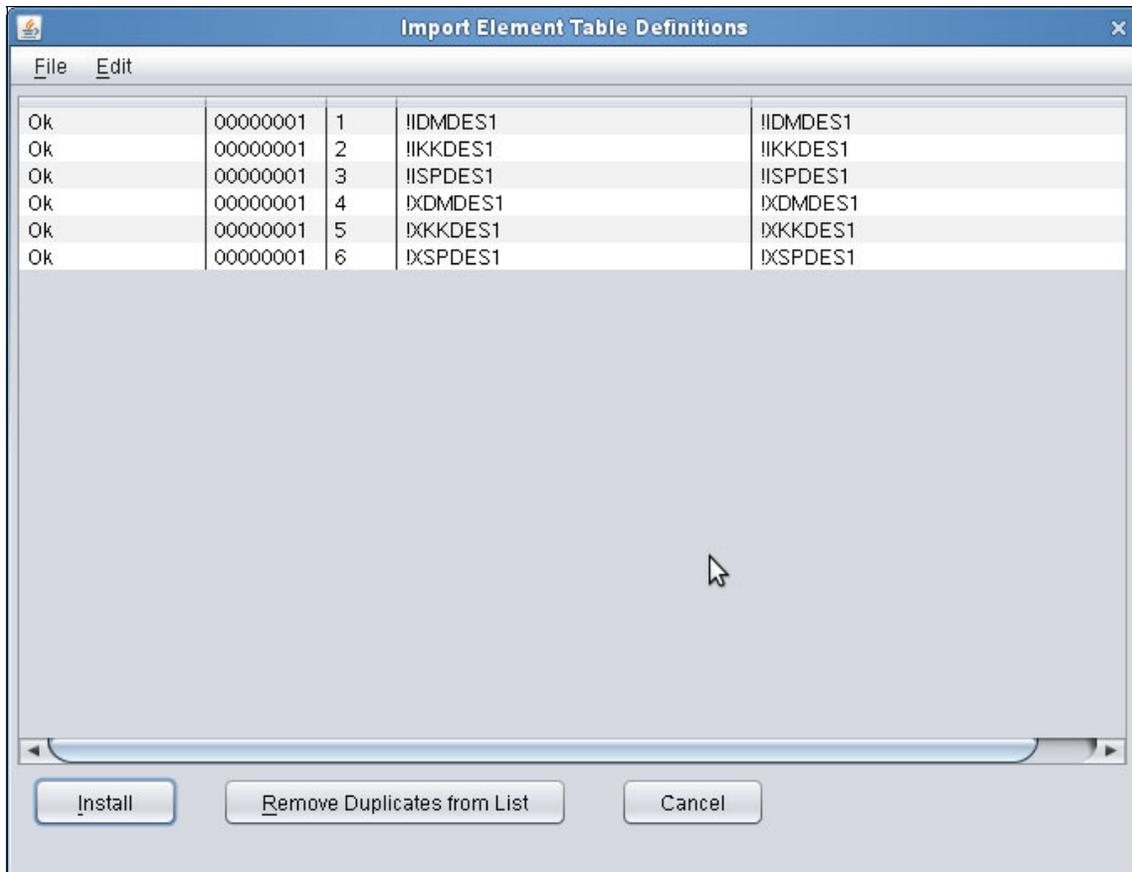


Figure 5-26 Element table - import elements - OK

- Repeat the previous steps for all the other import files.
- Click **Close** to close the Element Table Maintenance program and return to the autobuild main menu.

### **Configuring the UKDS7 settings**

The communication parameters of the DB2 database server on the mainframe are stored in the `/var/opt/ibm/dkms/environment-name/dbprefs.xml` file. To configure the database connection parameters, complete the following steps:

- Start the **PROG0323 Symmetric Key Management** program from the autobuild menu of the DKMS application.

- Click **Tools** → **Settings** and complete the fields that are shown in Figure 5-27 with the following information.

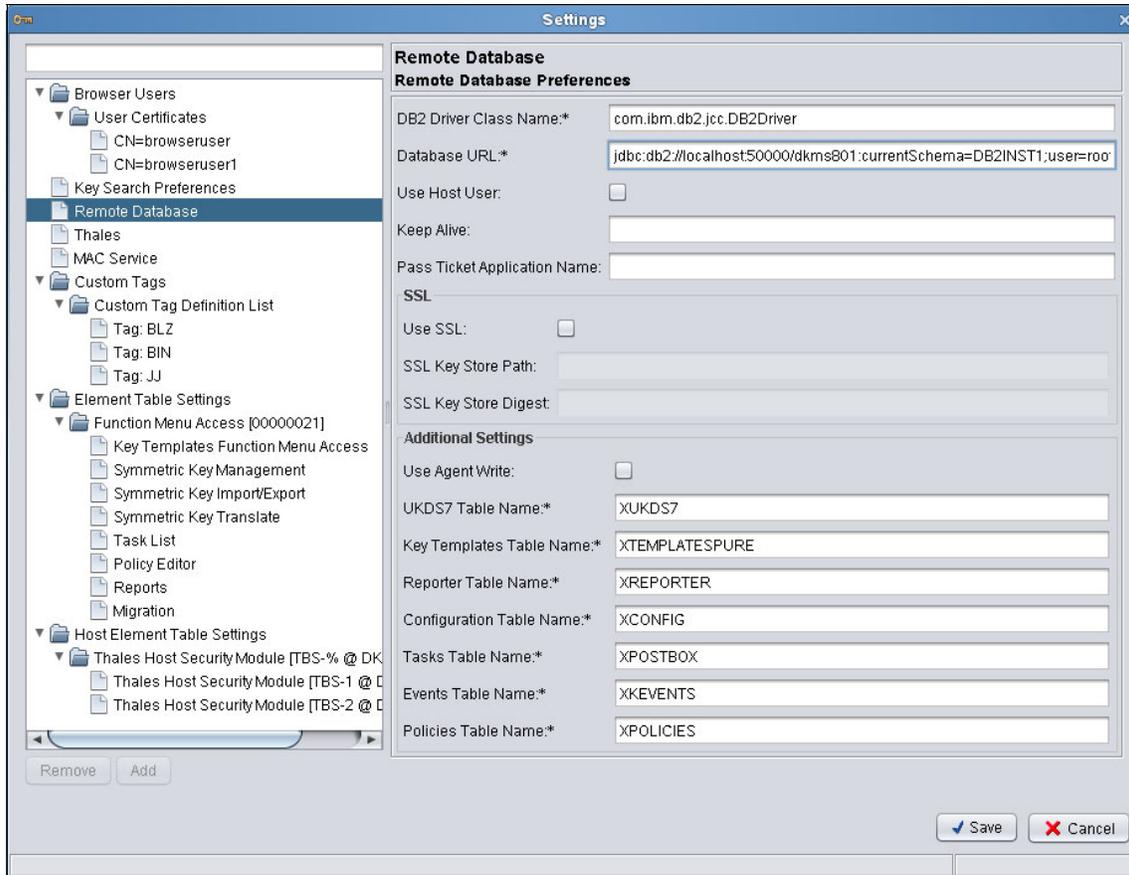


Figure 5-27 Settings - Remote Database

|                              |                                                                                                                                                                     |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DB2 Driver Class Name</b> | Enter the name of the JDBC driver to use. Defaults to com.ibm.db2.jcc.DB2Driver, which is the recommended setting.                                                  |
| <b>Database URL</b>          | The URL of the database server in the form<br>jdbc:db2://hostname:port/database[:option1;[option2;]...].                                                            |
| <b>Use Host User</b>         | When selected, this setting indicates that the host credentials should be used for the DB2 database authentication as well. Select this check box for our scenario. |

|                                    |                                                                                                                                                                                           |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Keep Alive</b>                  | An interval at which DKMS tries to communicate with the remote database to keep the connection alive.                                                                                     |
| <b>PassTicket Application Name</b> | Indicates the name of the application to use when requesting a ticket for database authentication. Leave empty for our scenario.                                                          |
| <b>Use SSL</b>                     | Indicates whether a connection to the database server should take place over SSL. Use SSL for JDBC connections. Clear the selection for our scenario.                                     |
| <b>SSL keystore Path</b>           | Denotes the path to a Java keystore file containing the certificates to use when verifying the database server's SSL identity. Leave empty for our scenario.                              |
| <b>SSL keystore Digest</b>         | Denotes an SHA-1 digest of the keystore file in the SSL keystore Path field. Leave empty for our scenario.                                                                                |
| <b>Use Agent Write</b>             | When checked, indicates that all database writes are performed by the Agent. If clear, JDBC is used for both reading and writing. Reading from the database always takes place over JDBC. |
| <b>Table Names</b>                 | Allows the user to configure table/view names on the database server. Leave the value at the defaults for our scenario.                                                                   |

3. In the navigation pane on the left side of the window, select **MAC Service**.
4. Select the **Insecure Mode** check box to deactivate MAC generation and verification on the database, as shown in Figure 5-28 on page 175. This check box must be cleared after the setup of the key hierarchy is done and the necessary MAC keys are installed in the system.

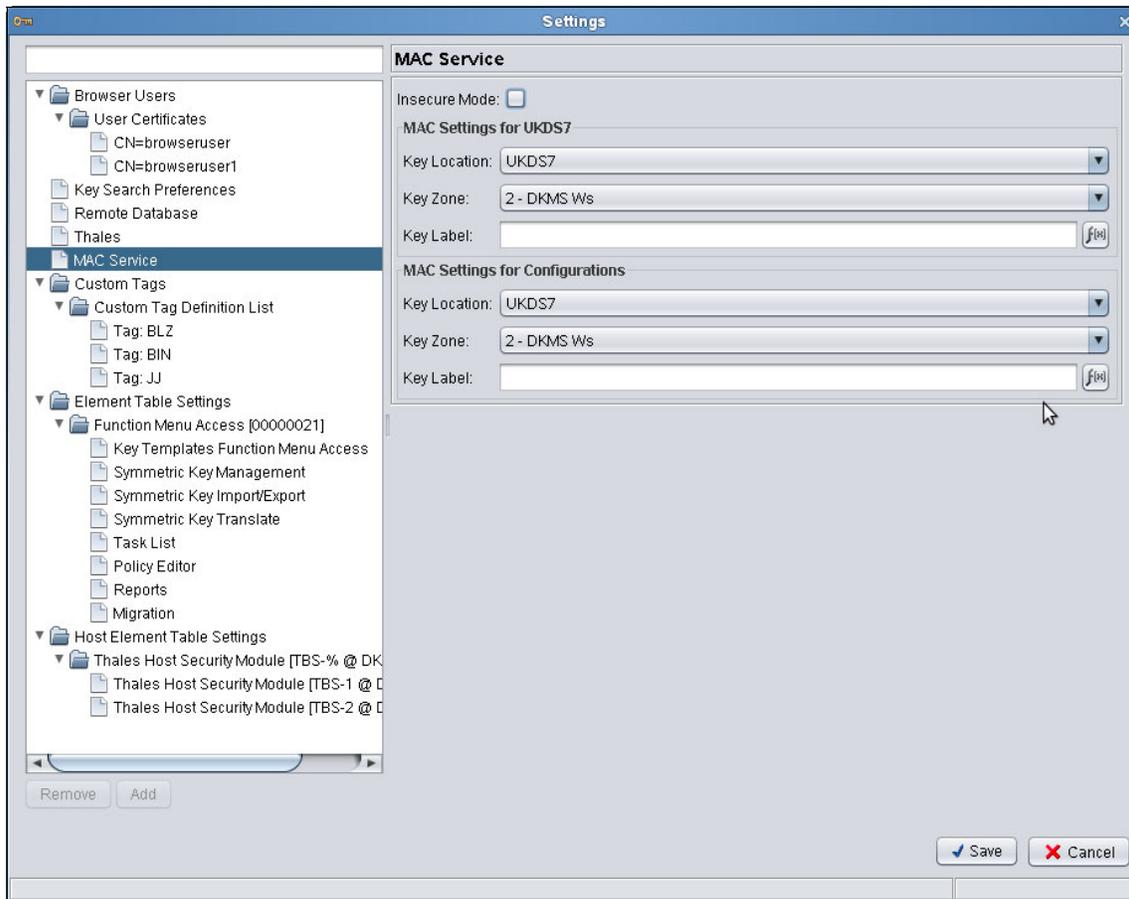


Figure 5-28 Settings - MAC Service

5. Click **Save** to save the changes.

### ***Configuring the DKMS application access groups and users***

You must configure the access groups in the DKMS application that group the users with the same role under the same group ID. The access group ID is used in “Access control maintenance” on page 180 to assign permissions to different resources within the DKMS application.

Also, you must define the same user names as in the smart card group profiles and smart card group-of-group profiles that you defined in the CCA Node Management Utility to log on to the IBM 4765.

Complete the following steps:

1. In the IBM Enterprise Key Management Foundation autobuild menu, start the **CONF0054 Access Control Maintenance** program.

Only the access group Administrator with the group ID 1 is present in IBM Enterprise Key Management Foundation, as shown in Figure 5-29.

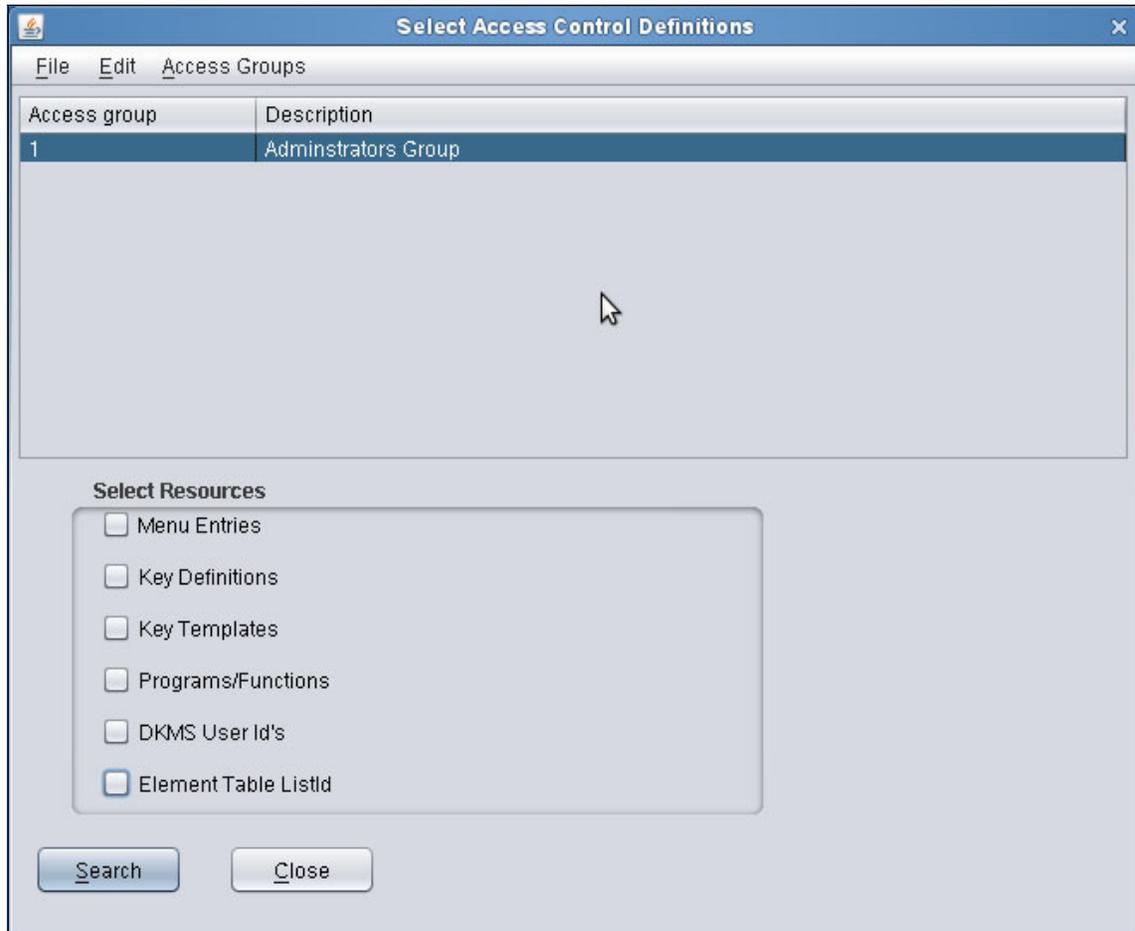


Figure 5-29 Access control maintenance

2. To create the access groups (2 for the Manager users, 3 for the Security Officer Group 1 users, and 4 for the Security Officer Group 2 users), click **Access Groups** → **List**, and then click **Access Groups** → **New**. Provide the ID and description for the access group and click **Create**, as shown in Figure 5-30. Repeat these steps for each of the access groups 2, 3, and 4.



The image shows a dialog box titled "New Access Group". It has a standard Windows-style title bar with a close button (X) in the top right corner. The main area of the dialog is light gray and contains two input fields. The first is labeled "Access group:" and has a small, empty text box next to it. The second is labeled "Description:" and has a larger, empty text box below it. At the bottom of the dialog, there are two buttons: "Create" on the left and "Cancel" on the right. Both buttons have a light blue gradient and a slight shadow.

Figure 5-30 Create New Access Group

3. To create logon users in EKMF with the same name as the IBM 4765 group profiles SO1 and SO2 and group-of-group profile names ADMIN and MANAGER, select the **DKMS User Id** check box and click **Edit** → **New DKMS user**. Enter the user ID and highlight the Access groups of which the user must be a member. Click **Create**, as shown in Figure 5-31.

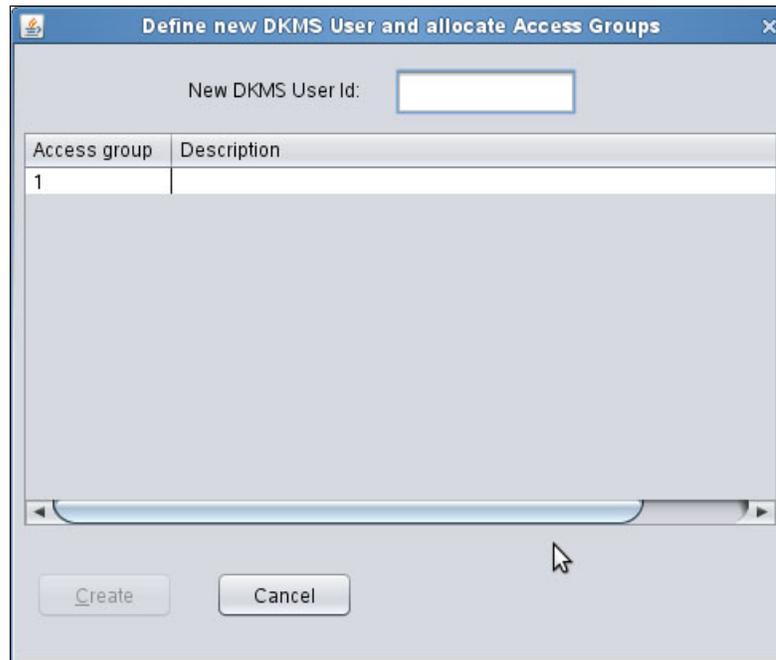


Figure 5-31 Define a DKMS user

### ***Customizing the menus***

IBM Enterprise Key Management Foundation provides simple mechanisms to define several customized menus of great value when instructions are written to operators. Special menus that are designed for the context of the key management operations make it easy for operators to perform their work without knowing DKMS in detail and without making too many mistakes.

The Key Management Workstation is delivered with a standard configuration of the menu system. A menu system consists of one or more top-level menus, related submenu entries, and related IBM Enterprise Key Management Foundation program entries.

Either use this standard-menu structure, create your own specific menu structure, or use the autobuild menu.

To create your own menu structure, see Chapter 5, “Menu Customization”, in *IBM Distributed Key Management System User’s Guide, Volume 1, Customization and basic DES key management*, DKMS-2200-10.

You can switch between the menu structure in the DKMS application by using the PROG0304 Changing Primary Menu program from the autobuild menu.

**Note:** The access groups that were created in “Configuring the DKMS application access groups and users” on page 175 must get the corresponding permission for each menu or submenu in the menu structure that is used. The permissions for each access group are defined in your role concept and must be given in the DKMS application, as described in “Access control maintenance” on page 180.

### ***Customizing the print forms***

When you start setting up the key hierarchy, as described in 5.3, “Managing keys” on page 185, you eventually need to print some keys or key parts on paper. For this purpose, sample key letter (print) forms are provided on the EKMF installation media in the /config/Basic Host with ICSF/form subfolder.

If you follow the instructions in “Configuring a new DKMS environment” on page 155, you should have these sample key letter print forms in the /var/opt/ibm/dkms/environment-name/form folder on your Key Management Workstation.

To customize these forms or create one, use the PROG0116 Define or Edit Letter program from the autobuild menu and follow the steps that are described in Chapter 8, “Generating, Printing and Entering Keys” → “Printing and Extracting Keys” → “Defining Forms for printing”, in *IBM Distributed Key Management System User’s Guide, Volume 1, Customization and basic DES key management*, DKMS-2200-10.

### ***Customizing the dashboard***

Use the dashboard to see an overview of pending actions and reports. It is possible to add and remove content and to move the content around by using either the mouse or the keyboard.

The example in Figure 5-32 shows a dashboard containing four widgets. With this dashboard, it is possible to see the status of importing clear parts, some recent reports, and a list of keys that are marked for translation.

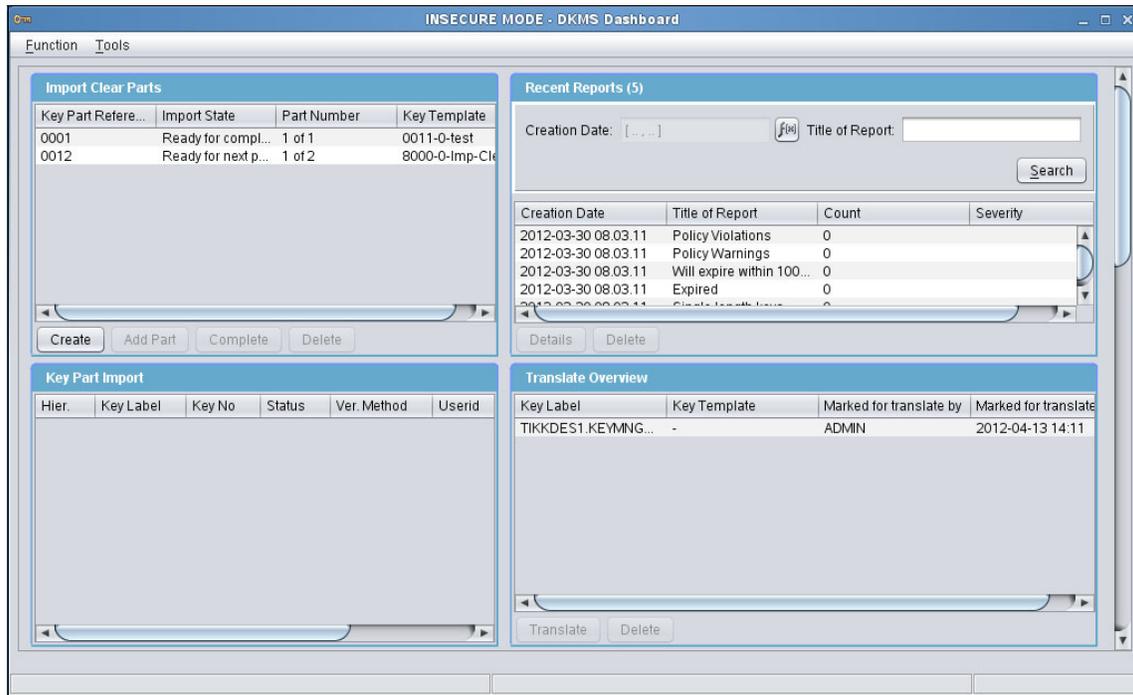


Figure 5-32 Dashboard

For more information about how to set up your own dashboard, see Chapter 13, “DKMS Dashboard” → “Customize your DKMS Dashboard”, in *IBM Distributed Key Management System User's Guide — Key Templates and Symmetric Key Management*, DKMS-2231-9, which can be found in the installation CD.

### **Access control maintenance**

After setting up the Key Management Workstation, you must verify and change the access group permissions for several resources so that they correspond to your role concept.

You can set permissions through the Access Control Maintenance for the following resources, as shown in Figure 5-33 on page 181:

- ▶ Menu entries
- ▶ Key definitions
- ▶ Key templates
- ▶ Programs and functions

- ▶ DKMS user IDs
- ▶ Element table entries

To set the permissions, complete the following steps:

1. Start the **CONF0054 Access Control Maintenance** program from the autobuild menu.
2. Select the check box that corresponds to the resource for which you want to change permissions. In our example, we select the **Programs/Functions** check box and click **Search** to continue, as shown in Figure 5-33.

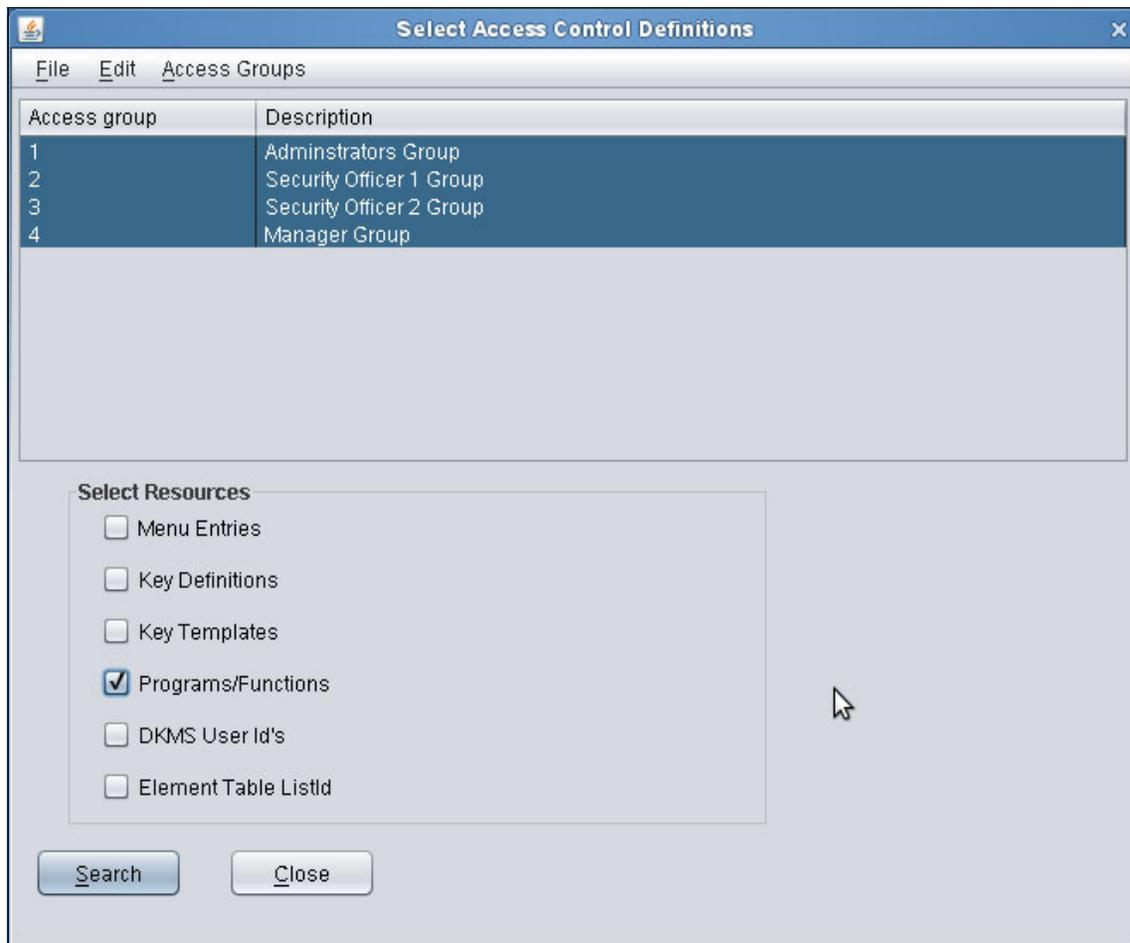


Figure 5-33 Access Control Maintenance - Program/Functions

3. In the initial setup, all resources can be used only by administrator access group 1. To give or take away permissions to the selected resources that are shown in the Figure 5-34, complete either of the following actions:
  - Add the access group or groups to the selected resources by clicking **Edit → Change**.
  - Delete the access group or groups from the selected resources by clicking **Edit → Delete**.

| Access groups | Resource Index | Description                |
|---------------|----------------|----------------------------|
| F 1           | CONF0030       | KeyDefinition Maintenance  |
| F 1           | CONF0033       | Menu Access Maintenance    |
| F 1           | CONF0034       | Comm. Settings             |
| F 1           | CONF0041       | Key Templates              |
| F 1           | CONF0054       | AccessControl Maintenance  |
| F 1           | CONF0055       | Element Table Maintenance  |
| F 1           | CONF0060       | DKMS Ws System Parameters  |
| F 1           | CONF0061       | Host Parm Maintenance      |
| F 1           | CONF0070       | System Policies            |
| F 1           | FUNC0001       | Generate key               |
| F 1           | FUNC0003       | Enter Key                  |
| F 1           | FUNC0004       | Enter all parts, same key  |
| F 1           | FUNC0005       | Extract all part, same key |
| F 1           | FUNC0007       | Inst. key from key parts   |
| F 1           | FUNC0008       | Inst. key from one part    |
| F 1           | FUNC0010       | Generate terminal keys     |
| F 1           | FUNC0012       | Key Translate              |
| F 1           | FUNC0013       | Change UKDS2 information   |
| F 1           | FUNC0014       | Ver&Gen KDT History Log    |
| F 1           | FUNC0015       | Browse audit Log - DB2     |
| F 1           | FUNC0017       | UKDS2 Delete Key in DB2    |
| F 1           | FUNC0018       | UKDS2 Restore Key          |

Figure 5-34 Access Control Maintenance - Programs / Functions list

Figure 5-35 shows how to add access group 4 (Manager Group) to the previously selected resources.

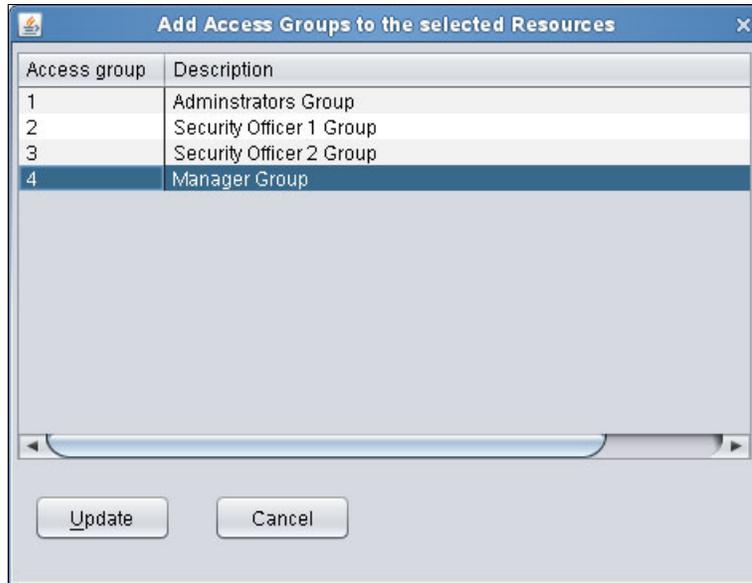


Figure 5-35 Access Control Maintenance - add access group

4. Click **Update** to save the configuration changes and return to the list of resources.
5. Click **Close** to return to the Access Control Maintenance window and repeat the previous steps for all the other resources.

**Note:** Any changes to these resources or the creation of new resources later, such as creating new key templates, sets the permissions to the access group 1 (Administrator Group). Therefore, you must verify and modify the permissions after each resource change.

## Small Linux guide

Table 5-6 shows the commonly used commands within the SLES operating system.

Table 5-6 Linux commands

| Linux command | Description                                                                                                                                                                                                                                                                                                                                                           | Example                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>chmod</b>  | Modifies access rights for the user, the group, and the other groups category. The following access rights are possible: <ul style="list-style-type: none"><li>▶ Read: Marked by 4 or r</li><li>▶ Write: Marked by 2 or w</li><li>▶ Execute: Marked by 1 or x</li></ul> A combination of these rights is possible by calculating the sum of their associated numbers. | <b>chmod 751 filename</b> changes the access rights of the <i>filename</i> file to Read+Write+Execute (4+2+1) for the user, Read+Execute (4+1) for the users group, and only Execute (1) for the other groups.<br><b>chmod 460 filename</b> changes the access rights to Read (4) for the user, Read+Write(4+2) for the users group, and gives no access(0) to the other groups. |
| <b>chown</b>  | Modifies the owner (user and group).                                                                                                                                                                                                                                                                                                                                  | <b>chown dkms:users filename</b> changes the owner of the <i>filename</i> file to the dkms user and group users.                                                                                                                                                                                                                                                                 |
| <b>ls</b>     | Lists directory contents. The <b>a1</b> option lists the owner and the access rights.                                                                                                                                                                                                                                                                                 | <b>ls -a1</b>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>cd</b>     | Moves between the folders.                                                                                                                                                                                                                                                                                                                                            | <b>cd folder-name</b> moves into the folder-name.<br><b>cd ..</b> moves into the parent folder.                                                                                                                                                                                                                                                                                  |
| <b>mkdir</b>  | Creates a directory                                                                                                                                                                                                                                                                                                                                                   | <b>mkdir dkms</b>                                                                                                                                                                                                                                                                                                                                                                |

| Linux command      | Description                                                       | Example                                                                                                                |
|--------------------|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>cp</code>    | Copies a source folder or files to a destination folder or files. | <code>cp /media/IBM DKMS 8.3.1 /tmp/dkms</code>                                                                        |
| <code>chown</code> | Modifies an owner (user and group)                                | <code>chown dkms:users filename</code> changes the owner of the <i>filename</i> file to the dkms user and group users. |

## 5.3 Managing keys

Before you can create the key templates for the keys to be managed, you must set up the device configuration in the DKMS application according to the environment of the bank.

Fictional Bank has three LPARs. Two of them are used for PIN authorization and one is used for PIN issuing. The two authorization LPARs are identical and can therefore be seen as one LPAR from the Key Management application application.

Fictional Bank uses key zone I and application name ISSUER the key to the issuing system. Fictional Bank uses key zone A and application name AUTHORIZ for the authorization system. This information must be entered in to the DKMS application.

### 5.3.1 Adding key zones

To add key zones, complete the following steps:

1. From the DKMS application main menu, open **CONF0055 - ELT Maintenance**, which shows the Element Table List shown in Figure 5-36.



The screenshot shows a window titled "Element Table List" with a menu bar containing "File" and "Edit". Below the menu bar is a table with the following columns: List Id., List Description, Type, User id ins., Date ins., Time ins., and User. The table contains 35 rows of data. The row with List Id. 00000006 and List Description "Cryptographic Device Type" is highlighted in blue. The "User" column for the last row (00000035) contains the text "ADM". At the bottom of the window is a "Close" button.

| List Id. | List Description          | Type | User id ins. | Date ins.  | Time ins. | User |
|----------|---------------------------|------|--------------|------------|-----------|------|
| 00000001 | 1. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000002 | 2. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000003 | 3. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000004 | 4. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000005 | 5. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000006 | Cryptographic Device Type |      | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000007 | DKMS DB2 Tables           |      | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000008 | System Application Names  | V    | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000009 | Activation Date Key Gen.  |      | ADMIN        | 2013-06-17 | 16.55.14  |      |
| 00000010 | Activation Date Key Entry |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000012 | Verification Methods      |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000013 | Key Generation Functions  |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000014 | Key Entry Functions       |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000015 | Admin. Key Functions      |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000016 | Exchange Media            |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000017 | Key Groups                | V    | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000018 | Key Length                | V    | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000021 | Functions and Programs    |      | ADMIN        | 2013-06-17 | 16.55.15  |      |
| 00000022 | Key Class                 |      | ADMIN        | 2013-06-17 | 16.55.16  |      |
| 00000023 | Usage Scope of key        |      | ADMIN        | 2013-06-17 | 16.55.16  |      |
| 00000024 | Customer Packages         |      | ADMIN        | 2013-06-17 | 16.55.16  | ADM  |
| 00000035 | Advanced KDT Functions    |      | ADMIN        | 2013-06-17 | 16.55.16  |      |

Figure 5-36 Element Table List

2. Double-click **00000006 Cryptographic Device Type**. Because the configuration for Basic host with ICSF was loaded during the installation of the the Key Management application, some of the cryptographic device types (called *key zones*) already are defined, as shown in Figure 5-37 on page 187.

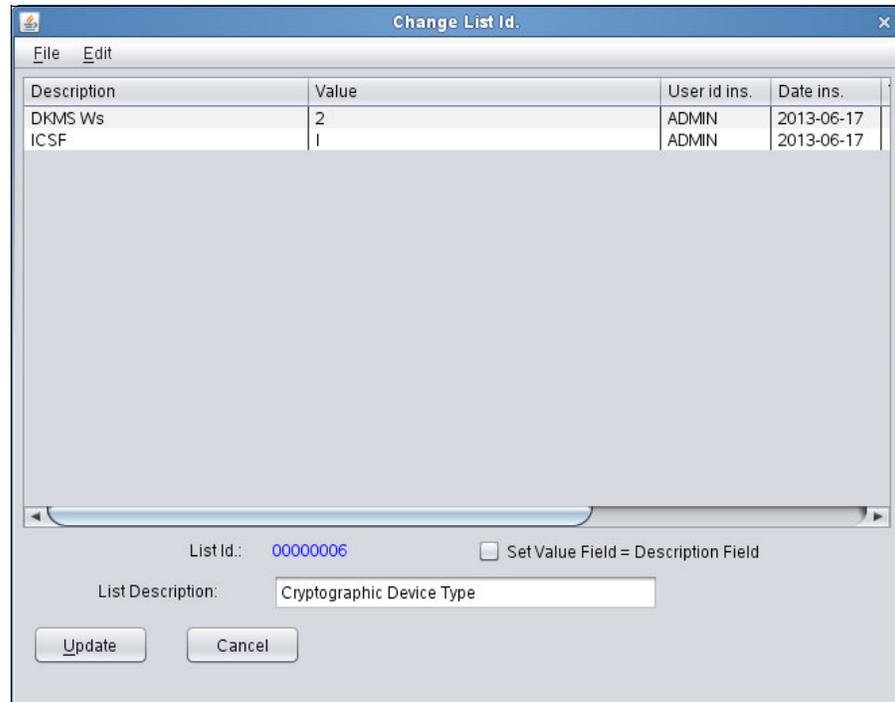


Figure 5-37 Pre-defined cryptographic device types

The DKMS Ws key zone is required to separate keys that are used only by the the DKMS application from other keys.

3. To add the key zone for the authorization system, click **Edit** → **New**, as shown in Figure 5-38.



The screenshot shows a dialog box titled "New List Id. Element". It contains the following fields and values:

- List Id.: 00000006
- List Description: Cryptographic Device Type
- Description: Authorization
- Value: A

At the bottom of the dialog are two buttons: "Create" and "Cancel".

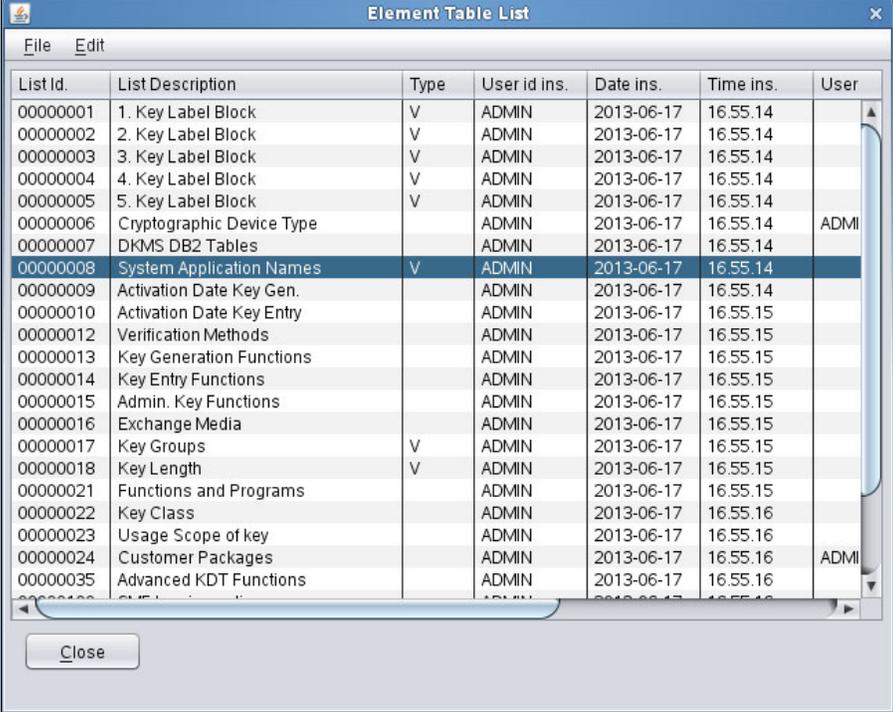
Figure 5-38 Create a key zone

4. Click **Create** and then **Update** in the Change List Id window.

## 5.3.2 Adding system application names

To add system application names, complete the following steps:

1. From the DKMS application main menu, open **CONF0055 - ELT Maintenance**, which opens the Element Table List shown in Figure 5-39.



The screenshot shows a window titled "Element Table List" with a menu bar containing "File" and "Edit". The window contains a table with the following columns: List Id., List Description, Type, User id ins., Date ins., Time ins., and User. The table lists various system application names, with "System Application Names" (List Id. 00000008) highlighted in blue. A "Close" button is located at the bottom left of the window.

| List Id. | List Description          | Type | User id ins. | Date ins.  | Time ins. | User |
|----------|---------------------------|------|--------------|------------|-----------|------|
| 00000001 | 1. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000002 | 2. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000003 | 3. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000004 | 4. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000005 | 5. Key Label Block        | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000006 | Cryptographic Device Type |      | ADMIN        | 2013-06-17 | 16:55:14  | ADMI |
| 00000007 | DKMS DB2 Tables           |      | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000008 | System Application Names  | V    | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000009 | Activation Date Key Gen.  |      | ADMIN        | 2013-06-17 | 16:55:14  |      |
| 00000010 | Activation Date Key Entry |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000012 | Verification Methods      |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000013 | Key Generation Functions  |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000014 | Key Entry Functions       |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000015 | Admin. Key Functions      |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000016 | Exchange Media            |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000017 | Key Groups                | V    | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000018 | Key Length                | V    | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000021 | Functions and Programs    |      | ADMIN        | 2013-06-17 | 16:55:15  |      |
| 00000022 | Key Class                 |      | ADMIN        | 2013-06-17 | 16:55:16  |      |
| 00000023 | Usage Scope of key        |      | ADMIN        | 2013-06-17 | 16:55:16  |      |
| 00000024 | Customer Packages         |      | ADMIN        | 2013-06-17 | 16:55:16  | ADMI |
| 00000035 | Advanced KDT Functions    |      | ADMIN        | 2013-06-17 | 16:55:16  |      |

Figure 5-39 Element Table List

2. Double-click **00000008 System Application Names**. The system application name KEYMNGNT is already defined. Keys that are used within IBM Enterprise Key Management Foundation must use this system application name, as shown in Figure 5-40.

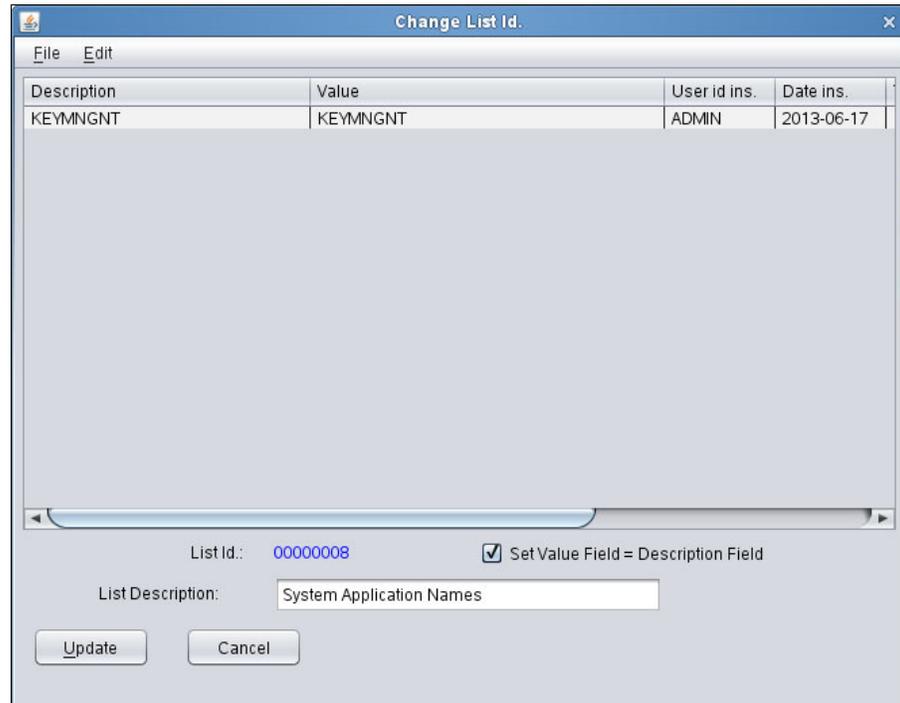


Figure 5-40 List of system application names

3. Click **Edit** → **New** to add the ISSUER system application name, as shown in Figure 5-41.



The screenshot shows a dialog box titled "New List Id. Element". It has a standard Windows-style title bar with a close button (X) on the right. The dialog contains three labeled text fields: "List Id." with the value "00000008", "List Description:" with the value "System Application Names", and "Description:" with the value "ISSUER". At the bottom of the dialog, there are two buttons: "Create" and "Cancel".

Figure 5-41 Add a system application name

4. Click **Create** and follow the same procedure to create a system application name named AUTHORIZ. Click **Update** in the Change List Id window. Click **Close** in the Element table list window.

### 5.3.3 Setting up the device configuration

The device configuration is used to map system application names and key zones to IBM Enterprise Key Management Foundation Agents. This information is used to determinate where to distribute keys.

The device configuration is maintained through the PROG0310 Device Configuration menu.

The first window, which is shown in Figure 5-42, in the device configuration shows the active configurations, which are empty when opened for the first time.

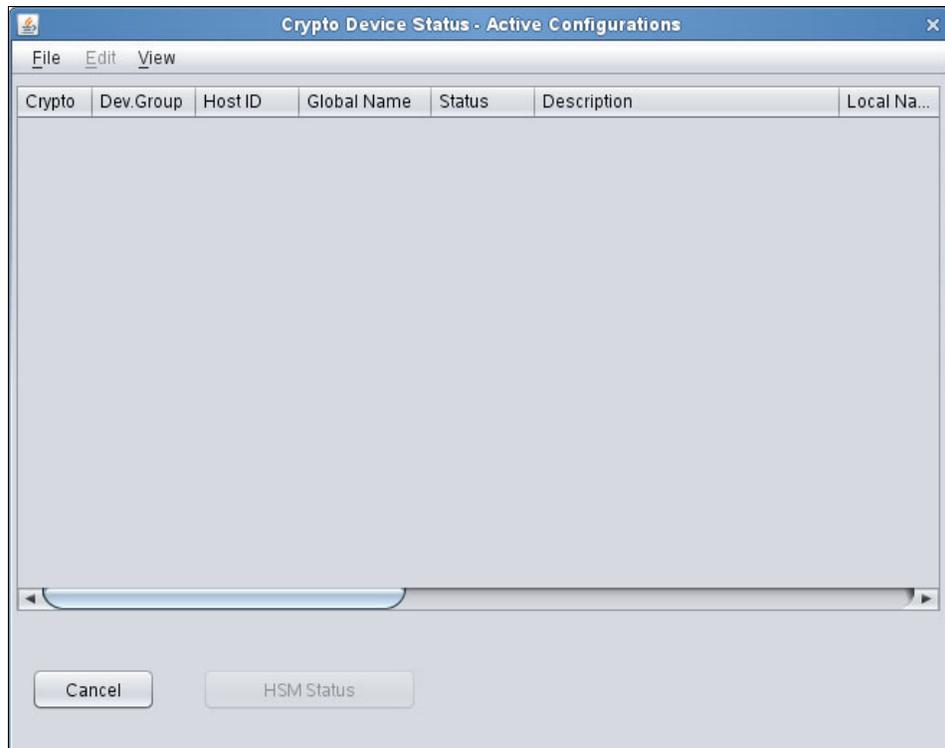


Figure 5-42 Empty device configuration

Complete the following steps:

1. Click **File** → **Hosts** to add information about how to communicate with the EKMF Agents. The list is empty when the Key Management application communicates only with the DB2 Agent that is defined during installation, as shown in Figure 5-43.

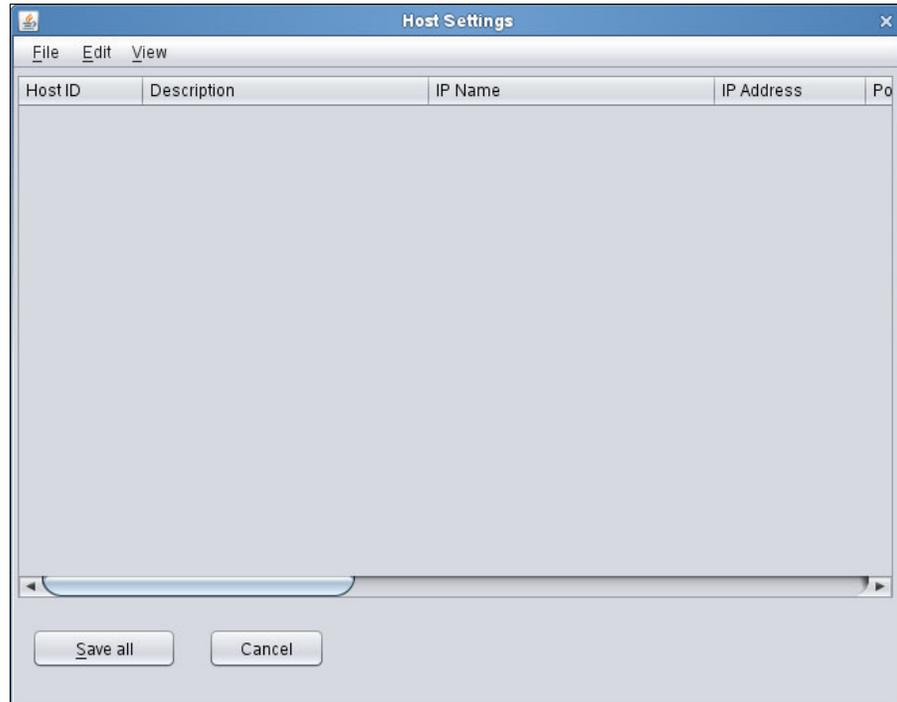
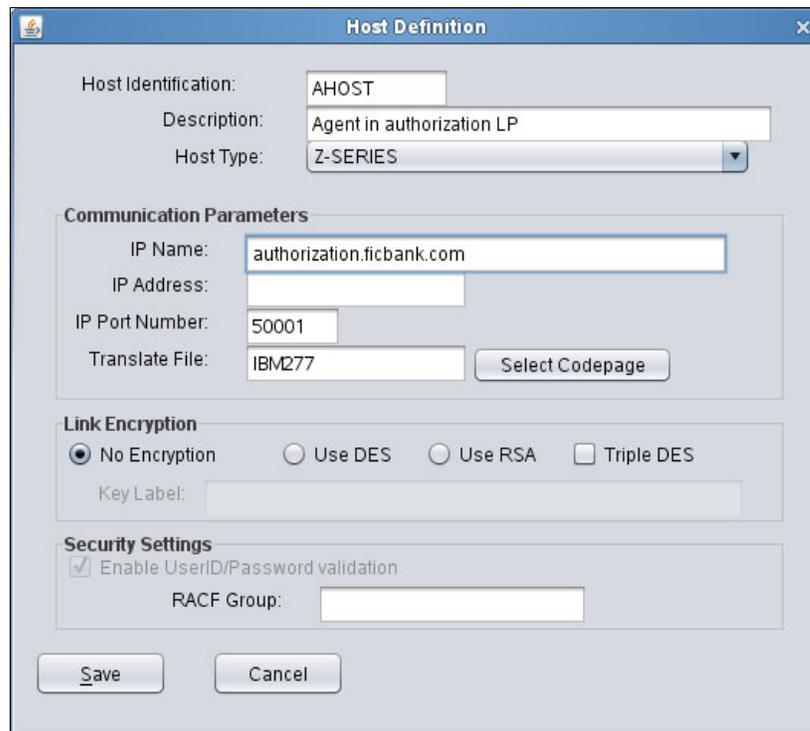


Figure 5-43 Empty list of hosts

2. Click **Edit** → **New** to add information about the IBM Enterprise Key Management Foundation Agent running on the authorization LPAR, as shown in Figure 5-44.



The image shows a 'Host Definition' dialog box with the following fields and options:

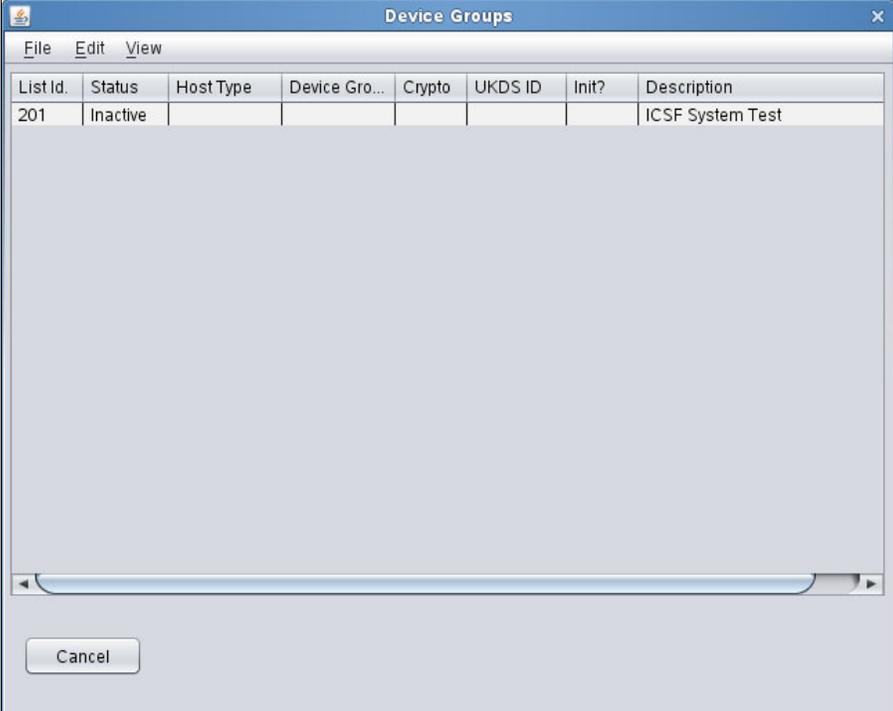
- Host Identification: AHOST
- Description: Agent in authorization LP
- Host Type: Z-SERIES
- Communication Parameters:
  - IP Name: authorization.ficbank.com
  - IP Address: (empty)
  - IP Port Number: 50001
  - Translate File: IBM277
  - Select Codepage button
- Link Encryption:
  - No Encryption
  - Use DES
  - Use RSA
  - Triple DES
  - Key Label: (empty)
- Security Settings:
  - Enable UserID/Password validation
  - RACF Group: (empty)
- Buttons: Save, Cancel

Figure 5-44 Add host communication parameters

3. Link encryption is disabled. The link encryption key must be generated before it can be enabled. Click **Save** and click **Save all** in the host settings window.

Fictional Bank does not need to add information about the EKMF Agent running in the issuer environment. This host is the DB2 host, and was configured during the first start of the DKMS application.

4. From the Crypto Device Status - Active configurations window, click **File** → **Configurations**. An empty, inactive device group is presented, as shown in Figure 5-45.



The screenshot shows a window titled "Device Groups" with a menu bar containing "File", "Edit", and "View". Below the menu bar is a table with the following columns: "List Id.", "Status", "Host Type", "Device Gro...", "Crypto", "UKDS ID", "Init?", and "Description". The table contains one row of data with the following values: "201", "Inactive", an empty cell, and "ICSF System Test". A "Cancel" button is located at the bottom left of the window.

| List Id. | Status   | Host Type | Device Gro... | Crypto | UKDS ID | Init? | Description      |
|----------|----------|-----------|---------------|--------|---------|-------|------------------|
| 201      | Inactive |           |               |        |         |       | ICSF System Test |

Figure 5-45 List of device groups

5. This group can be modified to fit Fictional Bank's environment. Double-click the device group and enter information about the issuing environment, as shown in Figure 5-46.

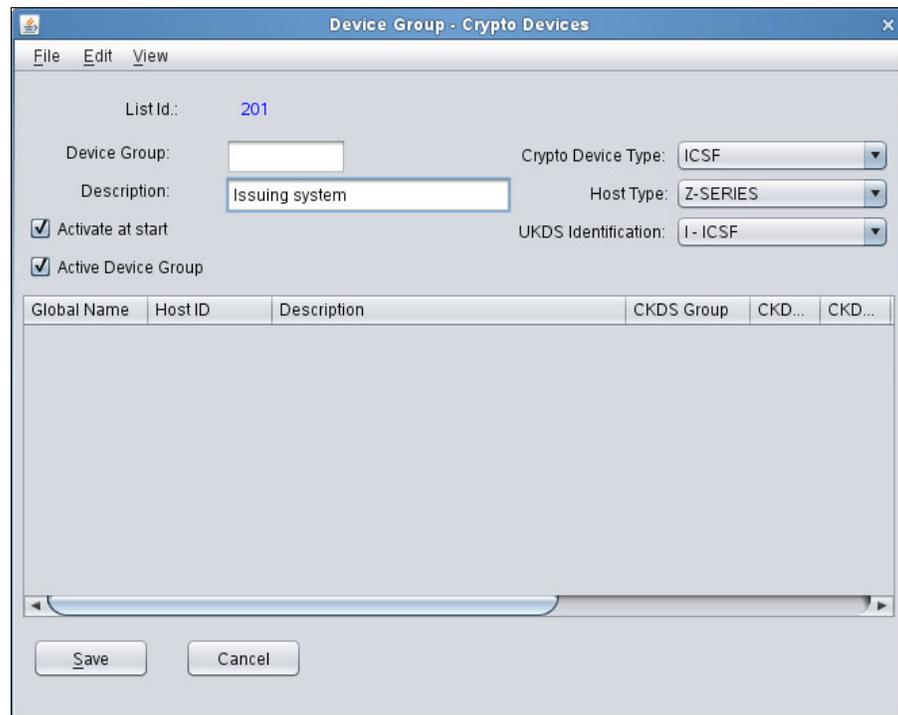


Figure 5-46 Add a device group

6. The device group must be active, and active from the start. It consists of ICSF devices on System z hosts, and is relevant for keys in key zone I. Click **Edit** → **New** to add a host to the device group, as shown in Figure 5-47 on page 197.

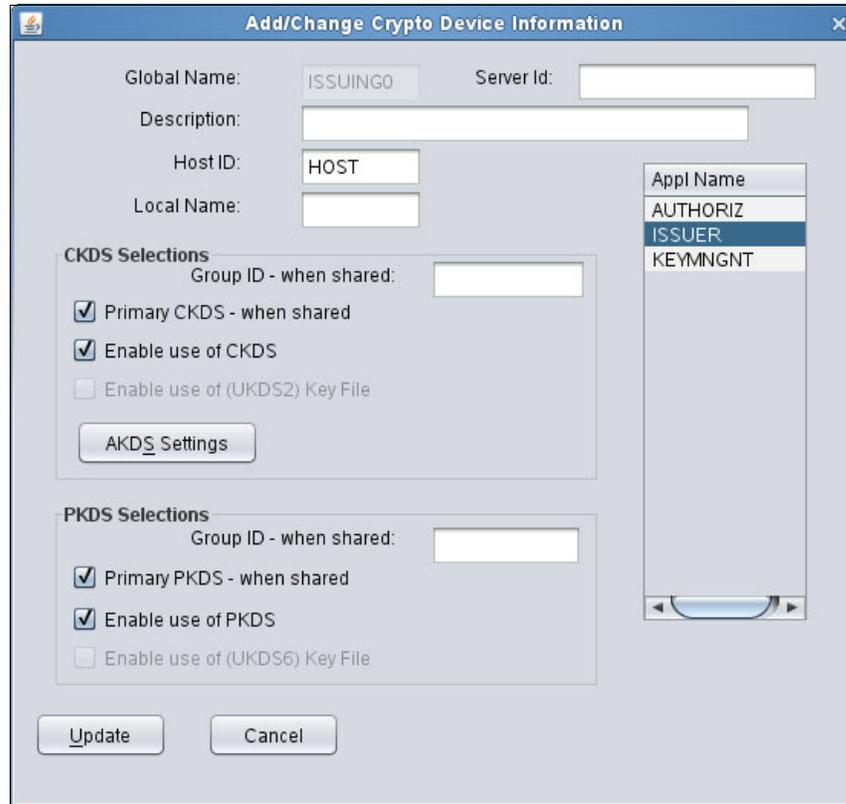


Figure 5-47 Add a host to a device group

The Global Name is a unique identifier that must be entered.

The Host ID is the name under which the host is known in the DKMS application. During the installation, the name HOST was selected for the primary DB2 Agent, which also must be used to distribute keys to the issuing system. In Appl Name, ISSUER is selected so that the keys that are used within the application are sent to the host. The selections that are made for CKDS/PKDS enable distribution of DES and RSA keys to this host. Fictional Bank does not need to distribute AES keys, so the AKDS settings are left blank.

7. Click **Update** and then **Save** in the Device Group - Crypto Devices window.
8. Follow the same procedure to add a device group for the authorization system. Select A - Authorization as UKDS Identification. Enter AHOST as the Host ID and select AUTHORIZ as the Appl Name.

The DKMS application must be restarted whenever changes are made to the device configuration.

## 5.3.4 Importing key templates

Fictional Bank decides to let IBM create their key templates.

To accomplish this task, complete the following steps:

1. Start **CONF0041 - Key Templates** and click **Function** → **Import**, as shown in Figure 5-48.

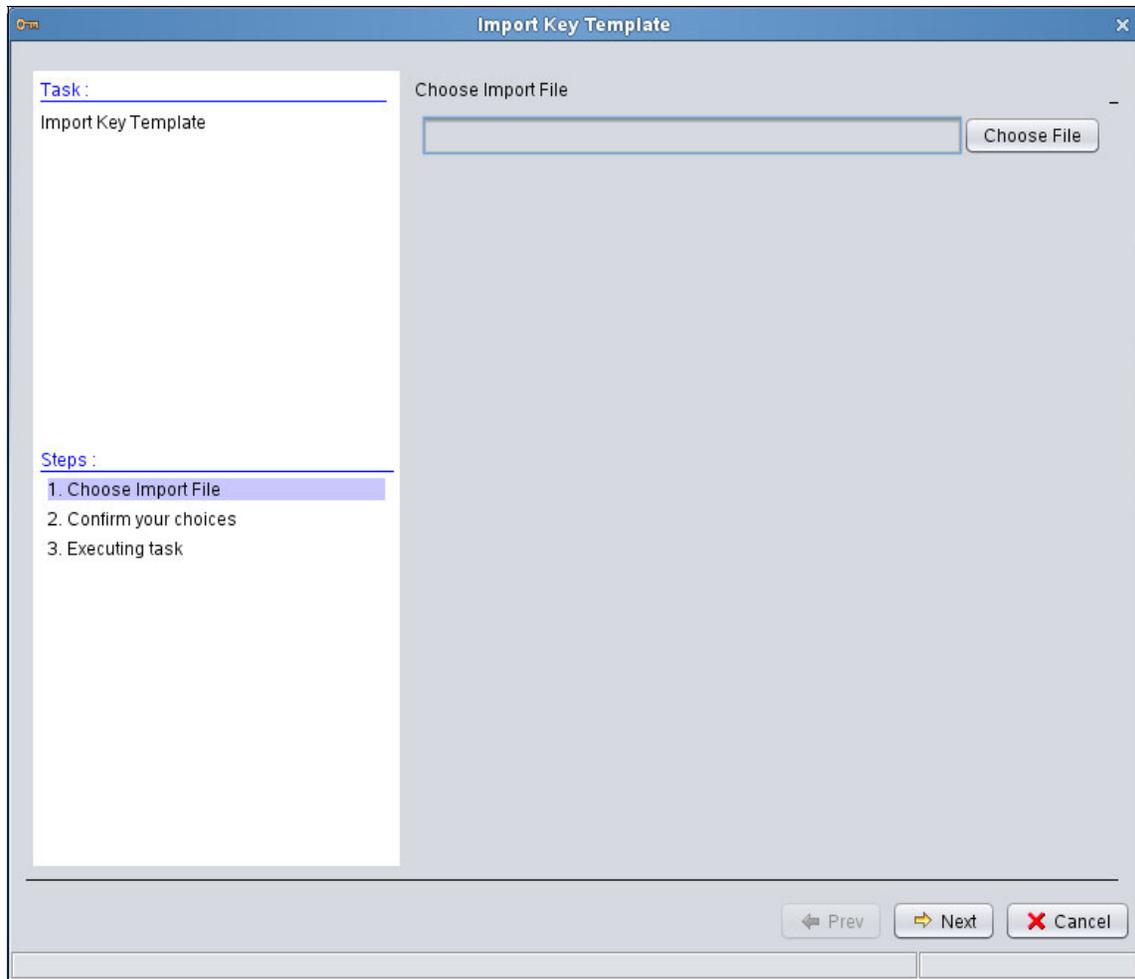


Figure 5-48 Import Key Template window

2. Click **Choose File**, select the file containing the key templates to import, and click **Next**. The content of the selected file is shown in Figure 5-49.

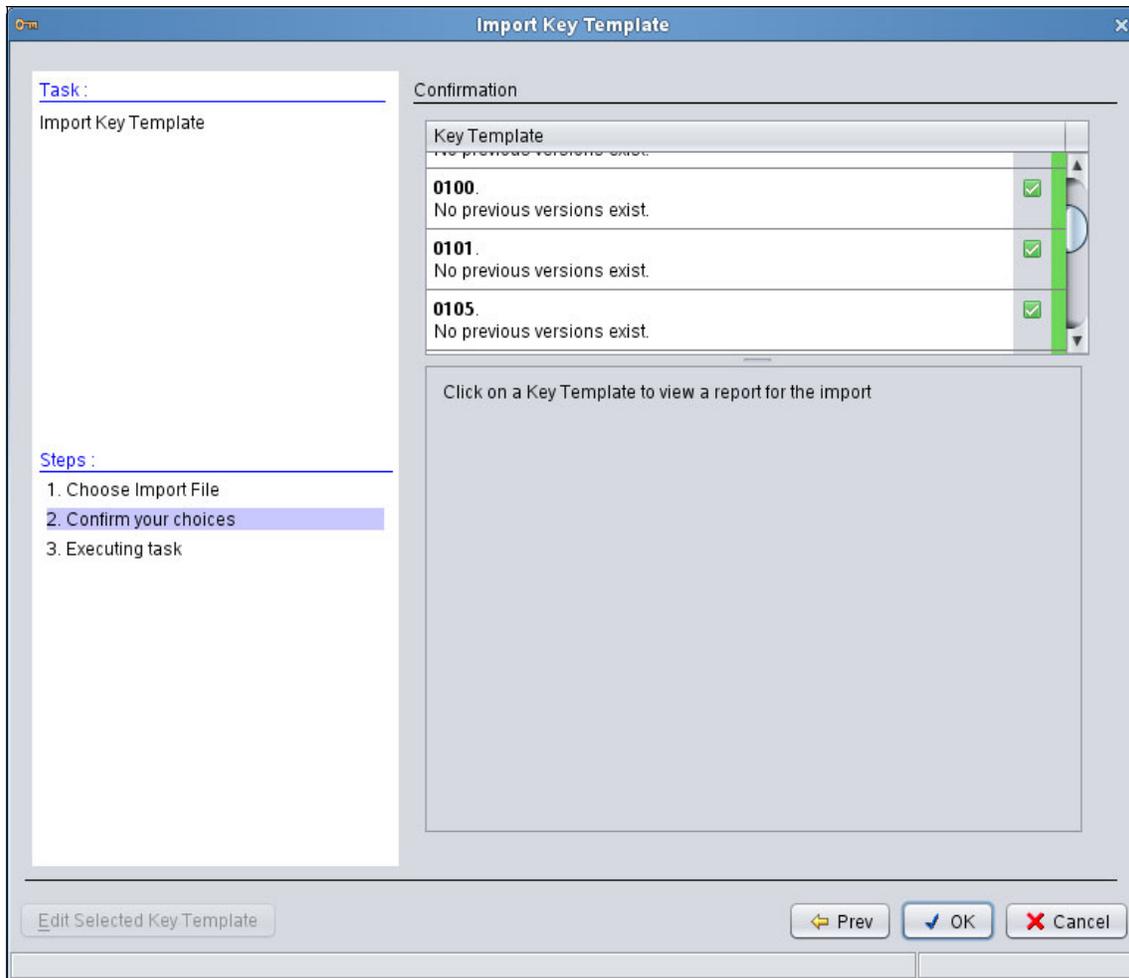


Figure 5-49 List of key templates to be imported

3. Scroll through the list to make sure all the key templates are marked green. If one or more key templates are marked red, it is not possible to import any of the key templates. A typical reason for a key template to be red is because it refers to a key zone or system application that is not defined on the importing system. Click **Edit Selected Key Template** if a key template must be modified before it can be imported. Click **OK** to import all the key templates.

A list of imported key templates is shown (Figure 5-50) when the operation is complete. The list can be compared to the key hierarchy that is described in Figure 5-5 on page 122.

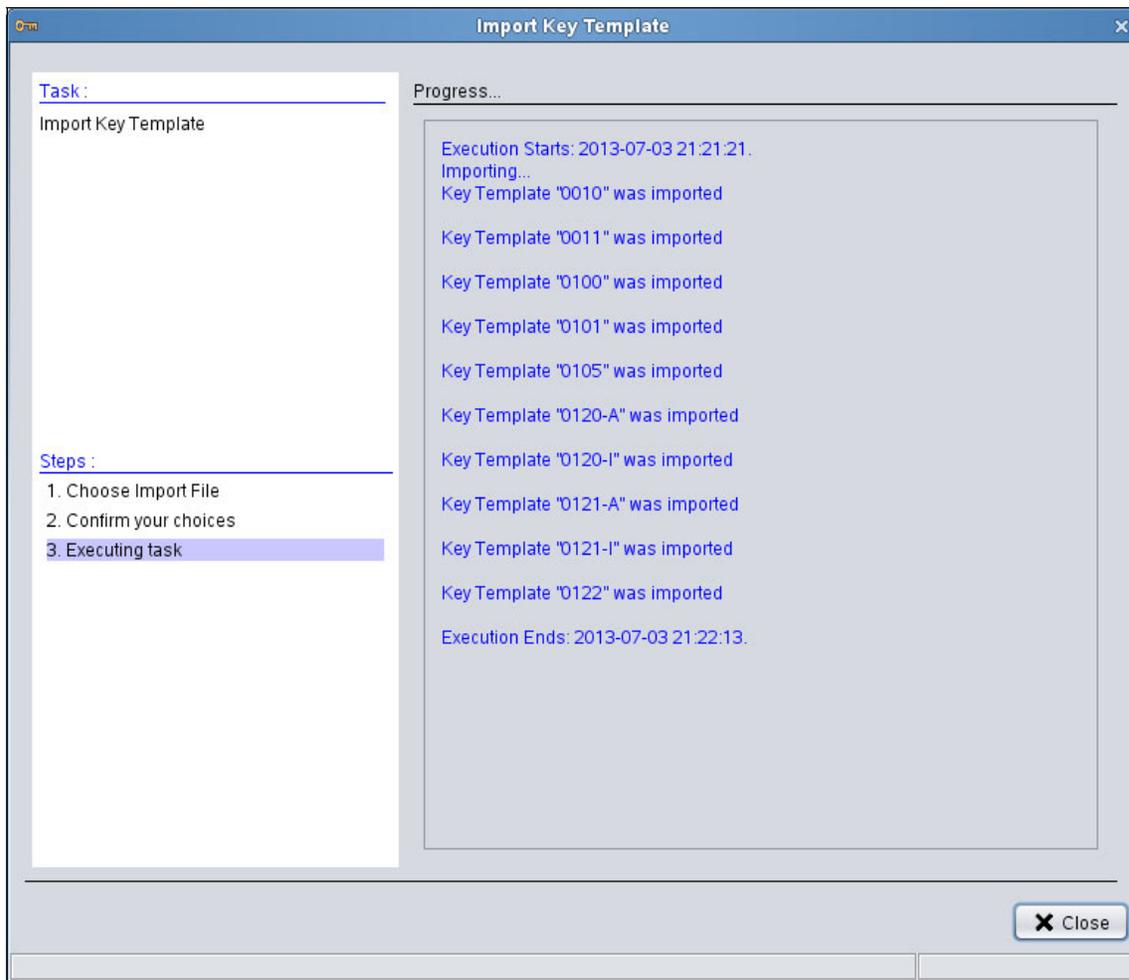


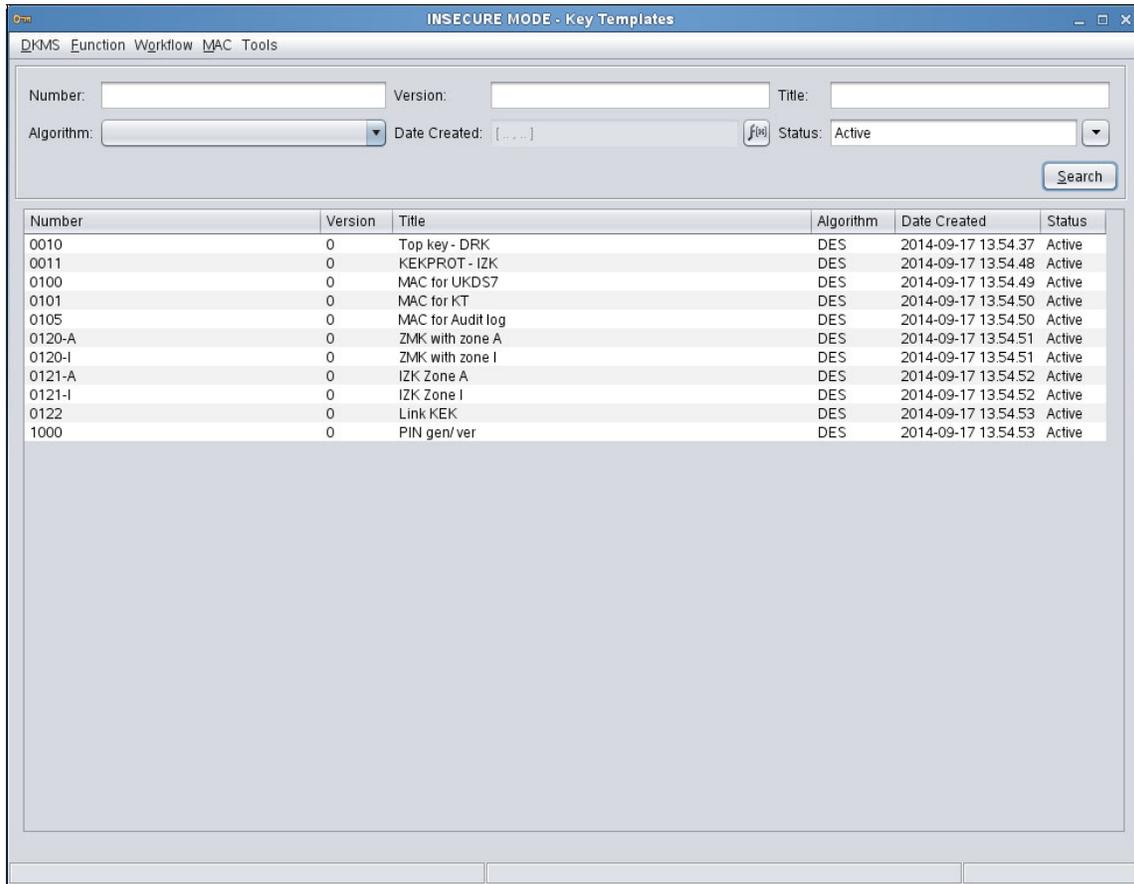
Figure 5-50 List of imported key templates

### 5.3.5 Verifying the key templates

To ensure that the imported key templates meet the requirements, each of them can be opened in the key template editor and compared to the infrastructure design. The key template editor can be used to create the key hierarchy from scratch, for example, to verify the 0121 key encryption key to the issuer system.

Complete the following steps:

1. Open **CONF0041 Key Templates**. Click **Search**. A list of all the key templates is shown in Figure 5-51.



The screenshot shows a software window titled "INSECURE MODE - Key Templates". At the top, there are menu options: "DKMS", "Function", "Workflow", "MAC", and "Tools". Below the menu is a search form with the following fields: "Number:" (text input), "Version:" (text input), "Title:" (text input), "Algorithm:" (dropdown menu), "Date Created:" (text input with a file icon), and "Status:" (dropdown menu with "Active" selected). A "Search" button is located to the right of the search fields. Below the search form is a table with the following data:

| Number | Version | Title             | Algorithm | Date Created        | Status |
|--------|---------|-------------------|-----------|---------------------|--------|
| 0010   | 0       | Top key - DRK     | DES       | 2014-09-17 13.54.37 | Active |
| 0011   | 0       | KEKPROT - IZK     | DES       | 2014-09-17 13.54.48 | Active |
| 0100   | 0       | MAC for UKDS7     | DES       | 2014-09-17 13.54.49 | Active |
| 0101   | 0       | MAC for KT        | DES       | 2014-09-17 13.54.50 | Active |
| 0105   | 0       | MAC for Audit log | DES       | 2014-09-17 13.54.50 | Active |
| 0120-A | 0       | ZMK with zone A   | DES       | 2014-09-17 13.54.51 | Active |
| 0120-I | 0       | ZMK with zone I   | DES       | 2014-09-17 13.54.51 | Active |
| 0121-A | 0       | IZK Zone A        | DES       | 2014-09-17 13.54.52 | Active |
| 0121-I | 0       | IZK Zone I        | DES       | 2014-09-17 13.54.52 | Active |
| 0122   | 0       | Link KEK          | DES       | 2014-09-17 13.54.53 | Active |
| 1000   | 0       | PIN gen/ ver      | DES       | 2014-09-17 13.54.53 | Active |

Figure 5-51 List of key templates

2. Double-clicking key template number 0121-I opens it in the key template editor, as shown in Figure 5-52.

The screenshot shows the 'Key Template Editor' window with the following configuration:

- Title:** IZK Zone I
- Number:** 0121-I
- Version:** 0
- Status:** Active
- Description:** (Empty text area)

**Key Creation Values:**

- Key Label:** <hierarchy>IZKDES.KEYMNGNT.IZK.ZONEI.KEK<seqno>
- Key State:** Active
- Algorithm:** DES
- Key Size:** DOUBLE
- Key Check Method:** 8: ENC-ZERO
- Active Date:** Today + 0d
- Expiry Date:** Today + 2y
- Origins:** Generate
- Comment:** KGN0,KGN7
- Allow keys of equal left and right halves:** No (selected)
- Prompt for Institution Number during Key Generation:** No (selected)

**Key Instances:**

| Application | Key Store Label                                 | Key Zone    | Key Store Type | Key Type | Install |
|-------------|-------------------------------------------------|-------------|----------------|----------|---------|
| ISSUER      | <hierarchy>IZKDES.KEYMNGNT.IZK.ZONEI.<keyTyp... | 1 - ICSF    | ICSF           | IMPORTER | Yes     |
| KEYMNGNT    | <hierarchy>IZKDES.KEYMNGNT.IZK.ZONEI.<keyTyp... | 2 - DKMS Ws | CCA            | EXPORTER | No      |

**Export Key Instances:**

| Export key | Export Key Label | Key Destination | Preferred Key Letter |
|------------|------------------|-----------------|----------------------|
|            |                  |                 |                      |

Buttons: Save, Cancel

Figure 5-52 Main window of the key template editor

- The key uses tags in the label to ensure that a unique label is assigned every time that the key is generated. It is a double-length DES key, which can enter the system only when it is generated. It has two key instances: One IMPORTER, which is installed in the ICSF on the issuer LPAR, and one EXPORTER, which is used by the DKMS application when generating keys for the issuer system. Double-click the issuer instance to see details about it, as shown in Figure 5-53.

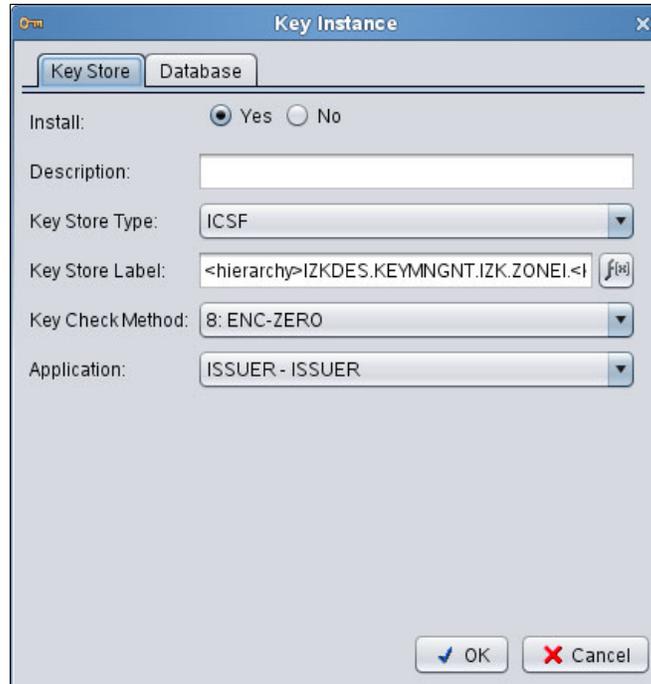


Figure 5-53 Keystore details for a key instance

- The key instance is installed in ICSF under a key label that is different from the key label it is stored under in the key repository. Click the **Database** tab to see more information about the key, as shown in Figure 5-54.



Figure 5-54 Database tab for a key instance

The control vector is used to specify the key type for DES keys. It can be modified directly, or a help dialog box can be opened if the magnifying glass icon is clicked.

In this example, the encrypting key is found at key generation time by finding the most current active key that is generated by the 0120-I key template.

### 5.3.6 Generating keys

The keys that are defined in the key templates must be generated, starting from the top of the key hierarchy. The top of the hierarchy is the master key of the IBM 4765, which is installed in the EKMF Workstation. This key was generated as part of the IBM 4765 configuration. The first key that is generated is the disaster recovery key, which is defined in key template number 0010.

Complete the following steps:

1. In PROG0323 Symmetric Key Management, click **Function** → **Generate Key**. The Generate Key wizard starts, as shown in Figure 5-55.

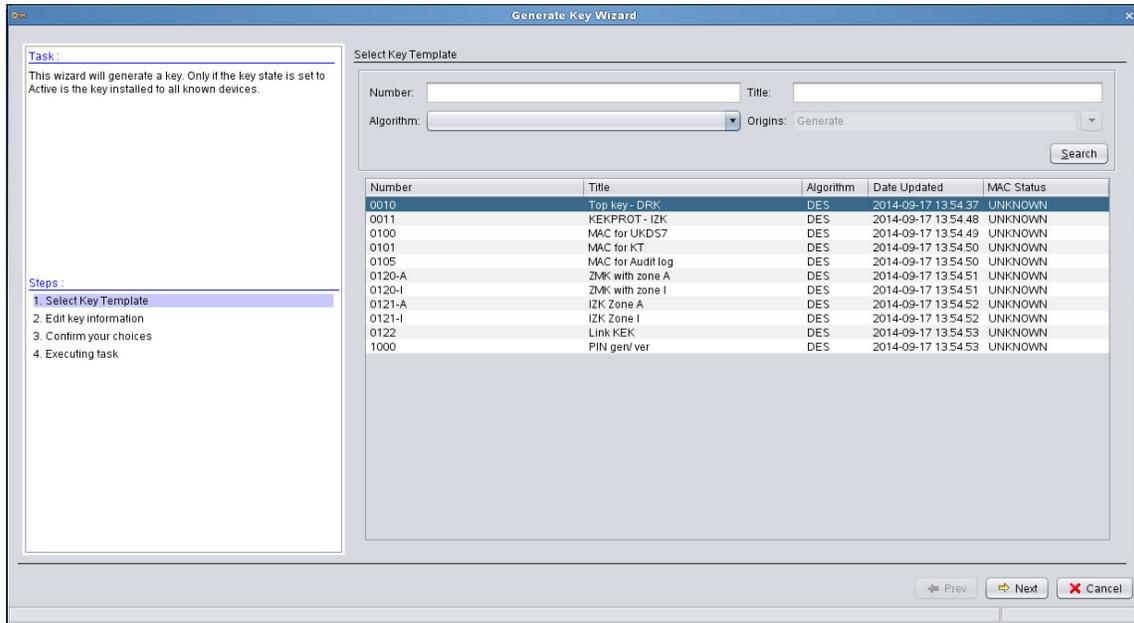


Figure 5-55 Select a key template to generate a key

- Click **Search**, select key template number 0010, and click **Next**. Information about the selected key template is shown in Figure 5-56.

**Task:**  
This wizard will generate a key. Only if the key state is set to Active is the key installed to all known devices.

**Steps:**

1. Select Key Template
2. Edit key information
3. Confirm your choices
4. Executing task

**Edit key information**

Key Label\* <hierarchy>DRKDES.KEYMNGNT.TOPKEY.IMP<seqno> f9

Key State\* Active Algorithm\* DES

Key Size\* DOUBLE Key Check Method\* 8. ENC-ZERO

Active Date\* Today Expiry Date\* Today + 10y f9

Comment: KGN0.KGNS

| Application | Key Store Label   | Key Zone    | Key Store Type | Key Type | Install |
|-------------|-------------------|-------------|----------------|----------|---------|
| KEYMNGNT    | Same as Key Label | 2 - DKMS Ws | CCA            | IMPORTER | Yes     |

| Export key | Export Key Label  | Key Destination | Preferred Key Letter |
|------------|-------------------|-----------------|----------------------|
| Yes        | Same as Key Label | Print           | IBMD010U.BF1         |

Prev Next Cancel

Figure 5-56 Show the selected key template

- Click **Next** to see a summary of the key that is about to be generated, as shown in Figure 5-57.

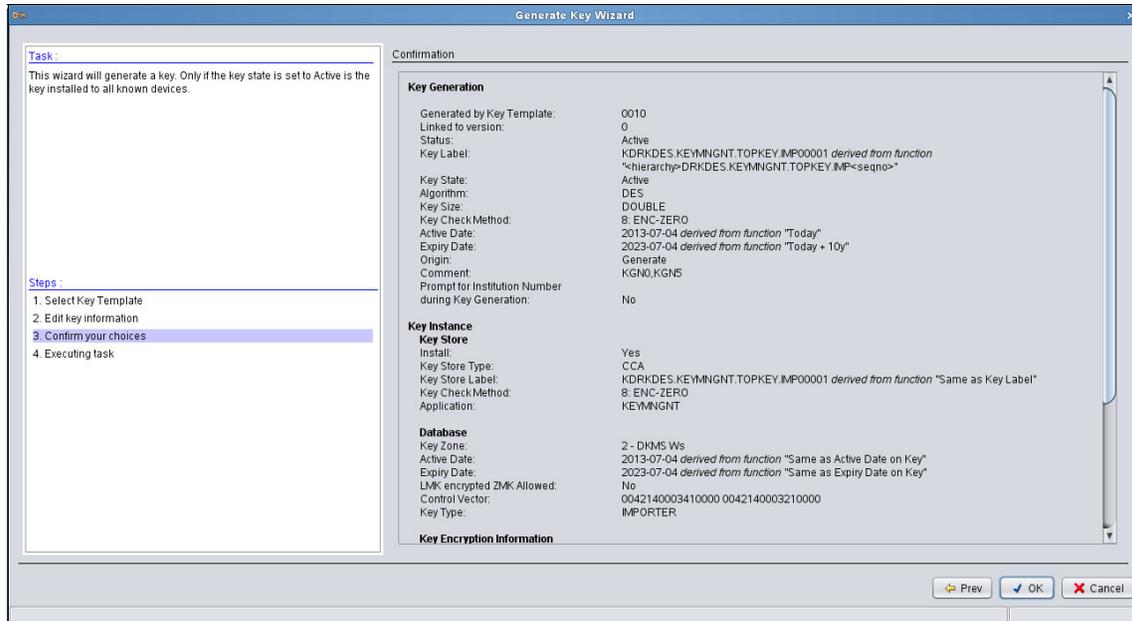


Figure 5-57 Details of the key that is about to be generated

4. Click **OK** to generate the key. The result is shown in Figure 5-58.

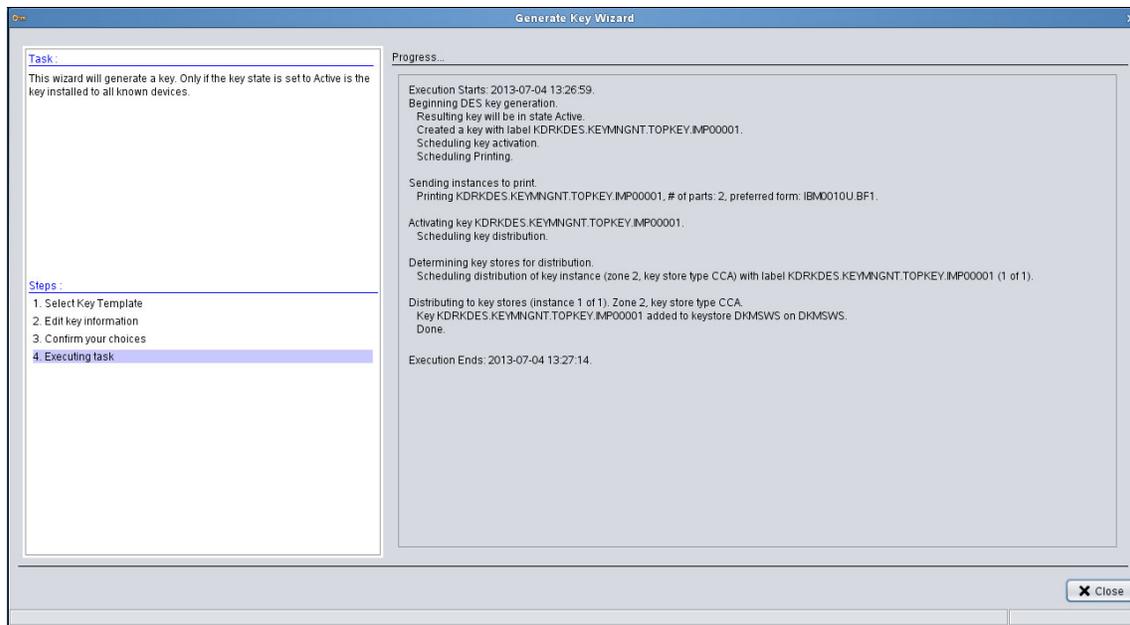


Figure 5-58 Result of the key generation

The key is generated, stored in the key repository, sent to be printed, and installed in the local keystore.

5. Repeat the key generation procedure for keys 0011, 0100, and 0101. These keys are used only locally on the EKM Foundation Workstation.
6. Generate the 0120-I key, which is the initial exchange key to ICSF on the Issuer LPAR. This key cannot be installed in ICSF because trust is not yet established between ICSF and the Key Management application. The key must be printed and entered in to ICSF through TKE.

- When the key is generated, start **PROG0330 - Print/Extract**, which opens the window that is shown in Figure 5-59.

The screenshot shows the 'PROG0330 - Print/Extract' window. The title bar indicates the user is 'ADMIN'. The window has a menu bar with 'File', 'Actions', and 'Sort'. Below the menu bar, there are several sections for configuration:

- Select keys depending on Types/Institution Id.**: Includes a 'First Key part' dropdown menu, an empty text input field, and checkboxes for 'All status types' (checked) and 'View by Institution' (unchecked).
- Print Form selection**: Includes a dropdown menu showing 'IBM0120U.BF1 Print ICSF Initial Exchange Key - First Key Component' and a 'Form Details' button.
- Output data type/Output media**: Includes a 'Key values' dropdown menu, a 'Single Key(info) on one Letter' dropdown menu, and checkboxes for 'Display warnings' (checked) and 'Remove from list when extracted' (unchecked).

At the bottom, there is a table with the following data:

| Key No | Key label                        | Ref.No. | S | Status      | Gen. Date  |
|--------|----------------------------------|---------|---|-------------|------------|
| 0010   | KDRKDES.KEYMNGNT.TOPKEY.IMP00001 |         |   | Not Printed | 2013-07-04 |
| 0120-I | KZMKDES.KEYMNGNT.ZONEI.IMP00001  |         |   | Not Printed | 2013-07-04 |

Figure 5-59 Keys ready to be printed

- Two keys are ready to be printed: The recovery key, and the initial exchange key to ICSF on the issuer LPAR. Select **First Key Part**, select print form IBM0120U.BF1, select key number 0120-I, and then click **Actions** → **Print/Extract**. The first key part is printed at the printer that is specified in the IBM0120U.BF1 print form.

If problems occur during printing, the settings for the print form might need to be changed. Use **PROG0116 - Define or Edit Letter** to change the printer. Print the last part of the key too.

Example 5-6 shows a sample of a key part that is printed on the IBM0120U.BF1 print form.

*Example 5-6 Printed key part*

---

I C S F I N I T I A L E X C H A N G E K E Y

Date Printed: 2013-07-04  
Key Letter Reference No.: 0001 (part two must have same no.)

This letter contains the first key component of the ICSF Initial Exchange Key with

Key Label: KZMKDES.KEYMNGNT.ZONEI.IMP00001

for Institution no. 00

Att.:

The key in this letter is to be used when setting up a secure path between DKMS WS and CCF/ICSF.

The key is entered into CCF/ICSF using TKE Workstation (recommended) or KGUP on host.

In both situation ICSF Verification Pattern is to be used as the verification method.

Key Value, First key component: 97B5 6794 D5A7 E0A1 0BFD D668 C849 A407

Key Verification values for first key component:

Verification Method: N  
Verification Key:  
Verification Code: 8432 4617 D568 EA02

Key Verification values for entire key:

Verification Method: N  
Verification Key:  
Verification Code: C392 D2F8 B1C9 3A95

Key Generate date: 2013-07-04  
Key Activate date: 2013-07-04  
Key Definition Number: 0120-I

End Key letter for first key component of the ICSF Initial Exchange Key.

9. Use both key letters to enter the key in to ICSF on the issuer LPAR.
10. Go back to the DKMS application to verify that the key also can be seen from there. Start the **Prog0323 - Symmetric Key Management** application and click **Search**. Locate and select the key in the list and double-click it. Click the **KeyStore Details** tab. Verify that the key is present in the issuer keystore, as shown in Figure 5-60.

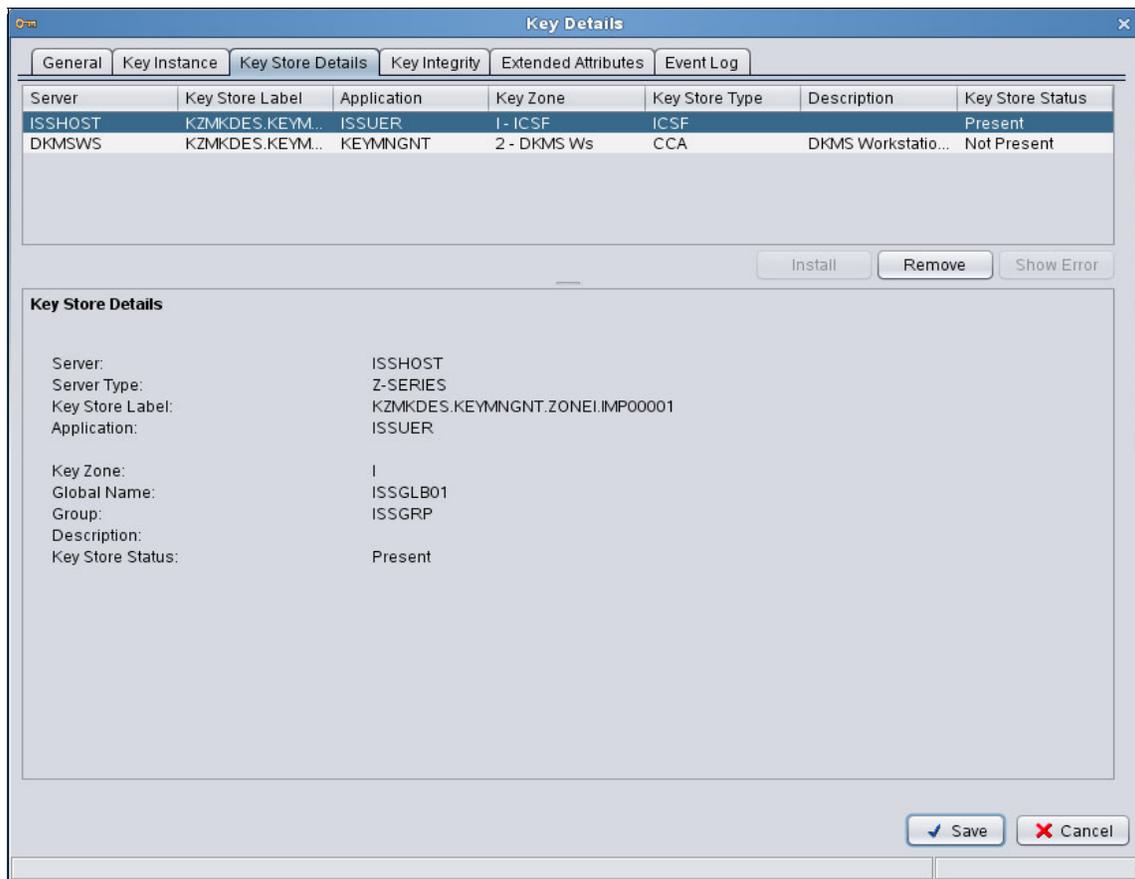


Figure 5-60 The key that is entered from TKE can be seen

The keystore status must have the Present status in the issuer keystore. If the status is Present (Wrong key), somebody entered a wrong key in TKE, and the key must be entered again. *Do not* click **Remove** if the correct key is present in the keystore because this action deletes the key from the keystore, and it must be entered manually from TKE again.

It is now possible to administer DES keys in the issuing LPAR. The same procedure must be performed by using key template number 0120-A for the authorization LPAR.

11. Now keys for the rest of the hierarchy can be generated. Generate the keys by using the 0121-I, 0121-A, 0105, and 0122 key templates. These templates create the key infrastructure, so a key that is used by external applications (for example, applications that can generate and verify PIN codes) can be managed. Before generating the application keys, the DKMS application must be switched in to secure mode and link encryption must be enabled.

### 5.3.7 Leaving insecure mode

When the DKMS application is set up for the first time, it is in insecure mode, which means that database records are not protected by a MAC. After the MAC keys are generated, it is possible to switch the system to secure mode, where MAC values are calculated on database records.

To switch to secure mode, complete the following steps:

1. Using CONF0041 - Key Templates, click **Tools** → **Settings** and select **MAC Service**. Clear **Insecure Mode** and select **UKDS7** as the key location and **DKMS WS** as key zone, as shown in Figure 5-61 on page 213.

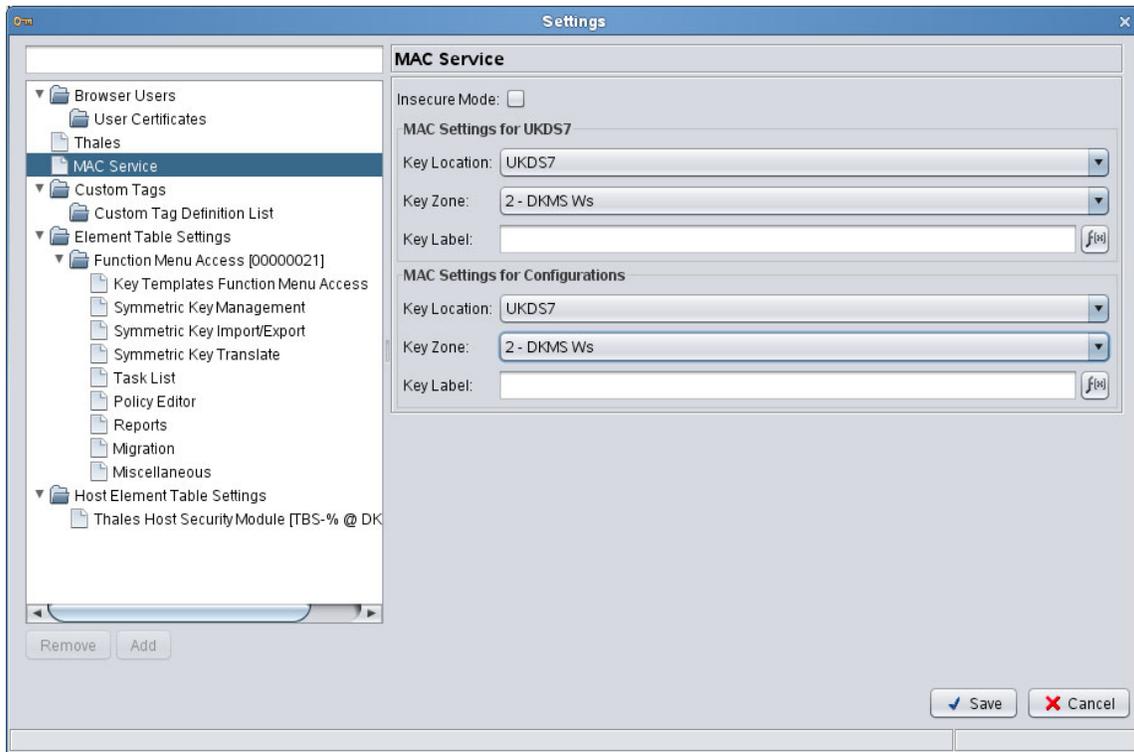


Figure 5-61 Set the MAC service to secure mode

2. Click **f(x)** for the Key Label field for the MAC for UKDS7 and select **Key of Key Template**. A dialog box opens in which you can select a key template (Figure 5-62).



Figure 5-62 Select a key template for the MAC key

3. Click the magnifying glass icon. Click **Search** and select key template **0100**, as shown in Figure 5-63.

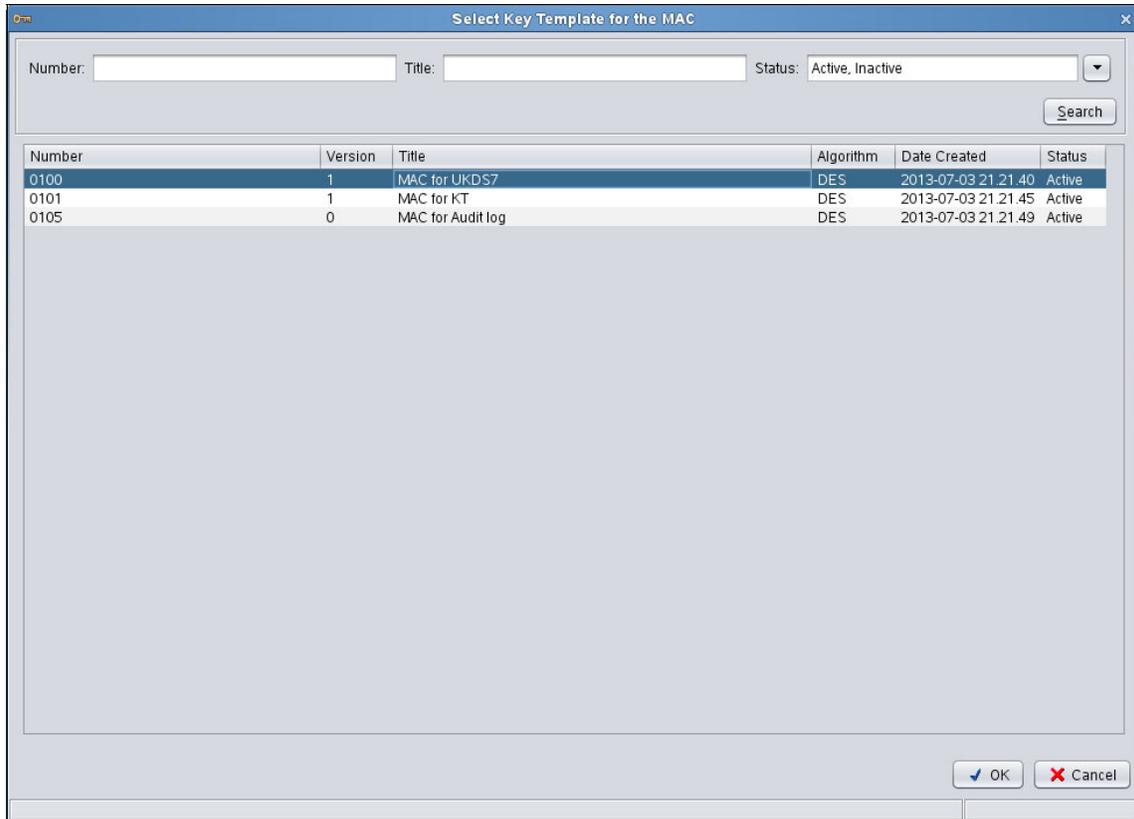


Figure 5-63 List of key templates usable for the MAC service

4. Click **OK**, and **OK** again in the parent dialog box.
5. Follow the same procedure to select the MAC key to protect the configuration. Select key template number 0101 for this task. Click **Save** in the settings dialog box. The system is still in insecure mode because the MAC keys that are selected do not yet have a valid MAC on their database records.
6. Start **PROG0323 - Symmetric Key Management**. Click **Search**. Click **MAC** → **Generate MAC** → **Keys in Search Result**. A dialog box opens and asks for confirmation, as shown in Figure 5-64 on page 215.

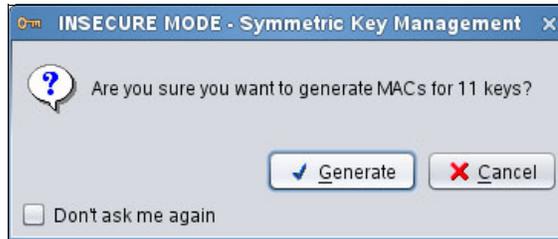


Figure 5-64 Confirm to generate a MAC for the keys

7. Click **Generate**. A dialog box opens when the operation is complete (Figure 5-65).

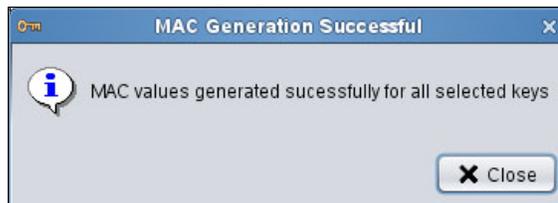


Figure 5-65 MAC generated for the keys

8. Click **Close**, click **Tools** → **Settings**, select **MAC Service**, clear **Insecure Mode**, and click **Save**. A dialog box asks whether the MAC must be calculated on items that are created in insecure mode, as shown in Figure 5-66.



Figure 5-66 Generate MAC for insecure items

9. Click **Generate MACs**. A dialog box opens when the operation is complete (Figure 5-67).

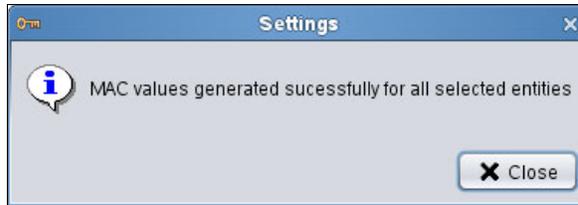


Figure 5-67 MAC successfully generated

The Key Management application now operates in secure mode.

## 5.4 Link encryption configuration

When link encryption is not enabled, IBM Enterprise Key Management Foundation communicates in the clear over the network. However, keys always are encrypted when transported outside the IBM 4765. Link encryption is recommended mainly because the credentials that are provided to RACF should be encrypted during transport.

The keys that are required to establish link encryption were generated by following the steps in 5.3.6, “Generating keys” on page 204 by using the key template number 0122. The key template placed a key encryption key in the Key Management Workstations local keystore, and the same key encryption key in each of the involved LPARs keystores.

Here is the key label that is specified in the key template for the link key:

```
<hierarchy>ZMDES.KEYMNGNT.LINKKEK.IMP<seqno>
```

### 5.4.1 Configuring the Agents

In the option data sets for the IBM Enterprise Key Management Foundation Agents running on all LPARs, the label must be specified by the &KEY-LABEL tag. The hierarchy letter and sequence number can be specified as \*s, so the option data set does not need to be modified if a new version of the link encryption key is created. Here is an example:

```
&KEY-LABEL (*ZMDES.KEYMNGNT.LINKKEK.IMP*****)
```

To avoid any communication in the clear, set &KEY-EXCHANGE to DES. The link encryption key is a double length DES key, so &KEY-LGT must be set to KEYLEN16.

```
&KEY-EXCHANGE (DES)
&KEY-LGT (KEYLN16)
```

Restart the KMG task when the option data set is changed.

## 5.4.2 Configuring RACF permissions

Reset the RACF FACILITY class profile CRYPTO.DKMS.LNKCRYOFF to enforce the use of link encryption by running the following command:

```
PERMIT CRYPTO.DKMS.LNKCRYOFF CLASS(FACILITY) ID(agent_ID) DELETE
```

## 5.4.3 Configuring the application

To configure the application, complete the following steps:

1. Start the DKMS application logon as a user profile starting with ADMIN. This action allows you to recalculate MAC values, which are required when the communication settings are changed.

When the EKM Agents are configured for encrypted communication only, the DKMS application fails to start because it is still set to communicate in the clear. Communication settings for the DB2 host are then shown, as shown in Figure 5-68.

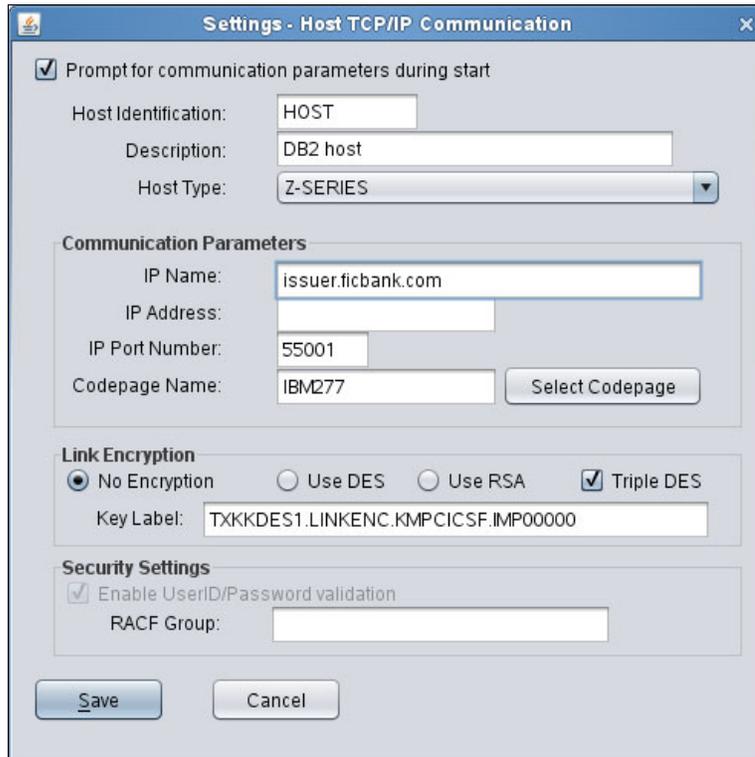


Figure 5-68 Communication parameters to the DB2 host

2. Change the parameters to use DES encryption.
3. Change the key label to the link encryption key in the local keystore. The first key that is generated with the link encryption key template has the following label:

`TZMDES.KEYMNGNT.LINKKEK.IMP00001`

Where T is the hierarchy letter for the installation.

4. Click **Save**. The DKMS application detects that the configuration changed, and shows an error (Figure 5-69).

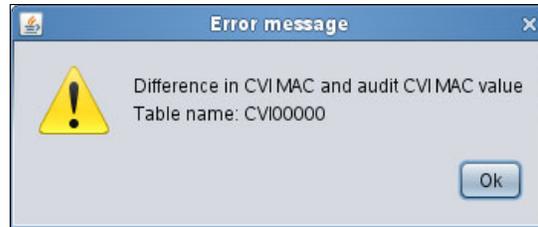


Figure 5-69 CVI MAC changed

5. Click **OK**. The DKMS application detects that the element table changed (Figure 5-70).

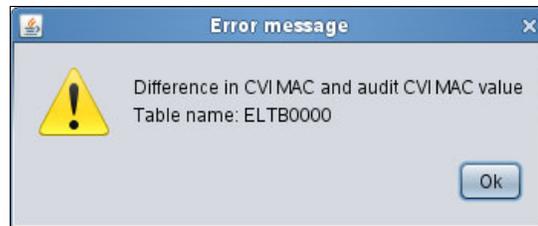


Figure 5-70 Element table changed

6. Click **OK**. The DKMS application cannot start when the MAC on the tables cannot be verified. Because the logged on profile starts with ADMIN, it is possible to recalculate the MAC. The Action message in Figure 5-71 shows this dialog box.



Figure 5-71 Updated MAC values on database tables

7. Click **Yes** to start the application. Start **PROG0310 - Device configuration** to change the communication parameters to the authorization LPAR. Click **File** → **Hosts**. The only other EKMF Agent that is defined in the system is shown in Figure 5-72.

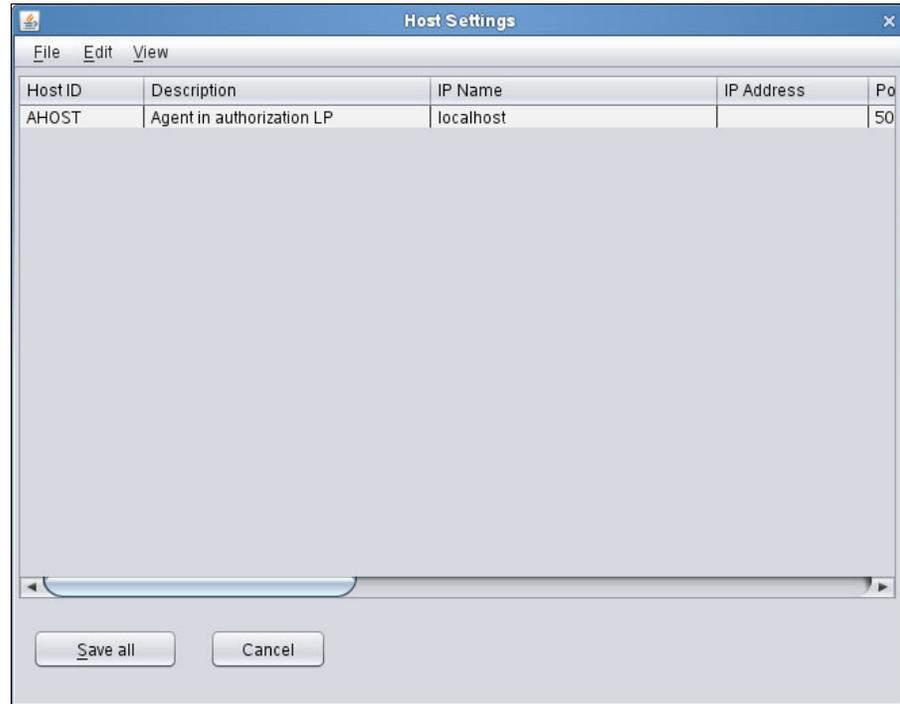


Figure 5-72 Non DB2 Agent that is defined in the system

8. Double-click **AHOST** and change it so that it can use link encryption the same way as it was done for the DB2 host during start.
9. Click **Save all**, and restart the DKMS application.

## 5.5 Application keys

Application keys are different from the system keys that have been generated until now because application keys must be usable only in applications running on the hosts. It is assumed that the people generating application keys have less authority on the Key Management Workstation than the people setting it up.

The workflow for generating application keys can differ considerably from that of the system keys because application key generation is often a recurring event that involves another group of people, and the system key generation is usually a one-time event involving the system administrator group.

## 5.5.1 Requesting key generation

The key generation can be split in to two operations, where the first is to request a key to be generated, and the second is to generate it. The request can be entered from both the DKMS application and the EKMF browser.

To request key generation, complete the following steps:

1. Start **CONF0041 - Key templates**. Click **Search** and select key template number 1000. Click **Workflow** → **Request Key Generation**. The Generate Key wizard opens, as shown in Figure 5-73.

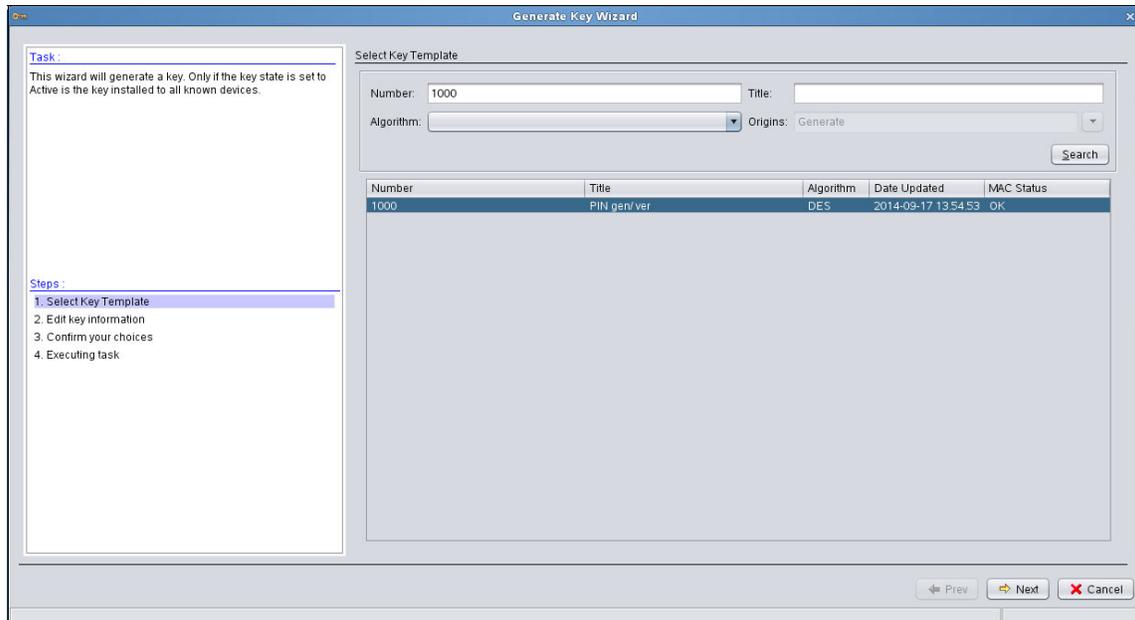


Figure 5-73 Request key generation

2. Click **Next**. Verify that the selected key template has the expected attributes, as shown in Figure 5-74.

**Task:**  
This wizard will generate a key. Only if the key state is set to Active is the key installed to all known devices.

**Steps:**

1. Select Key Template
2. Edit key information
3. Confirm your choices
4. Executing task

**Edit key information**

Key Label\*: <hierarchy>DES.APP.PIN.<seqno> [f9]

Key State\*: Pre-activation Algorithm\*: DES

Key Size\*: DOUBLE Key Check Method\*: 8\_ENC\_ZERO

Comment: [ ]

Active Date\*: Today [f9] Expiry Date\*: Today + 3y [f9]

Expiry Date Low: - - [f9]

| Application | Key Store Label                  | Key Zone         | Key Store Type | Key Type | Install |
|-------------|----------------------------------|------------------|----------------|----------|---------|
| ISSUER      | <hierarchy>DES.APP.PIN.PG<seqno> | I - ICSF         | ICSF           | PINGEN   | Yes     |
| AUTHORIZ    | <hierarchy>DES.APP.PIN.PV<seqno> | A - Authoriza... | ICSF           | PINVER   | Yes     |

Prev Next Cancel

Figure 5-74 Verify the key template attributes

3. The application key has key instances only in zones I and A. The key is not usable from the Key Management Workstation. Click **Next**. A summary of the key generation request is shown (Figure 5-75).

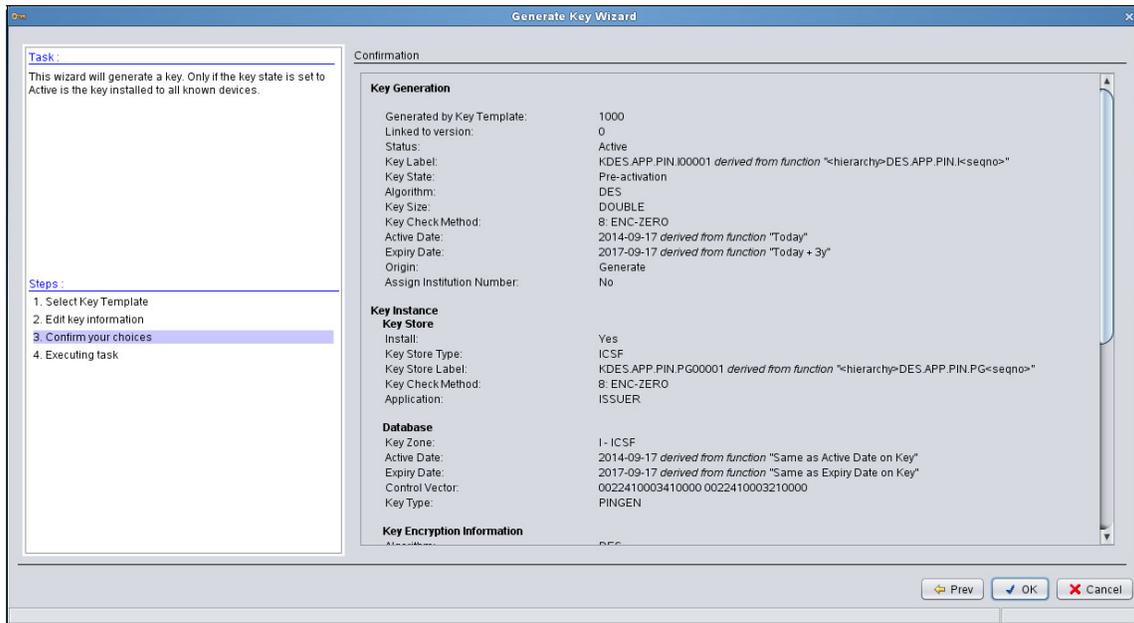


Figure 5-75 Summary of key generation request

4. Click **OK** to close the wizard.

## 5.5.2 Processing a key generation request

To process a key generation request, complete the following steps:

1. In the DKMS application, open **CONF0041 - Key Templates** and click **Workflow** → **Task List**. The task list is shown (Figure 5-76), containing the key generation request that was created in 5.5.1, “Requesting key generation” on page 221.

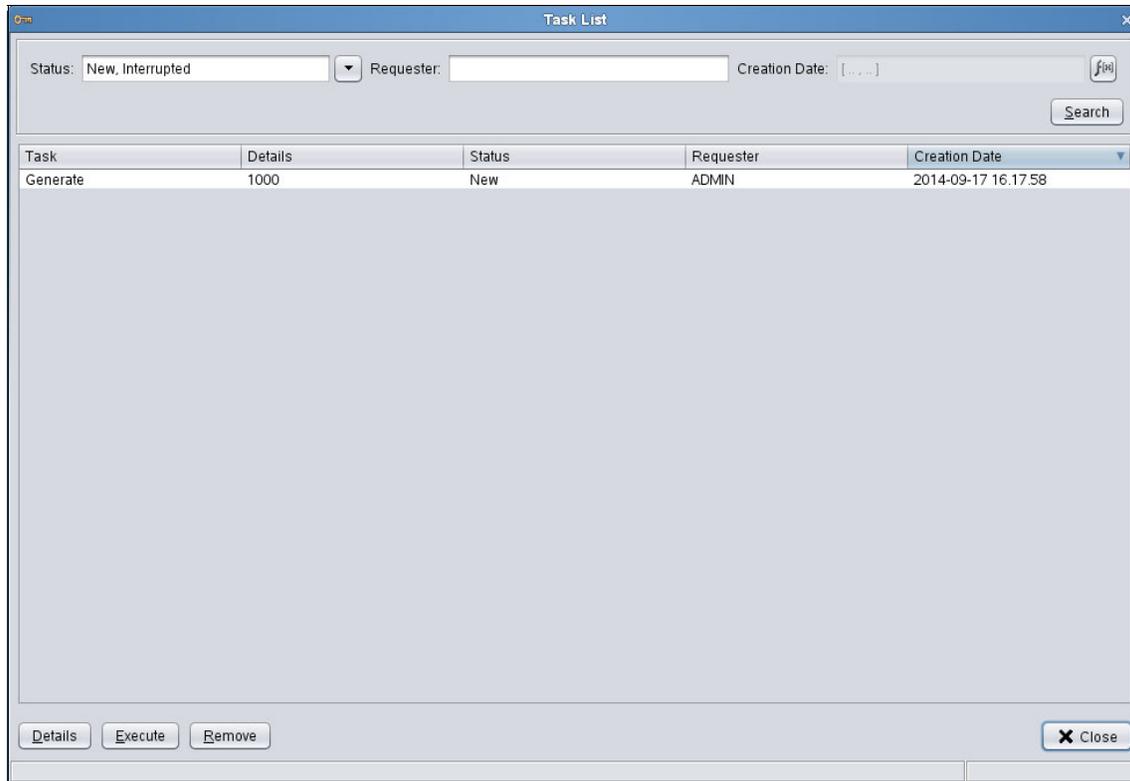


Figure 5-76 List of tasks to be performed

2. Select the generated task and click **Execute**. A summary of things to be executed is shown in Figure 5-77.

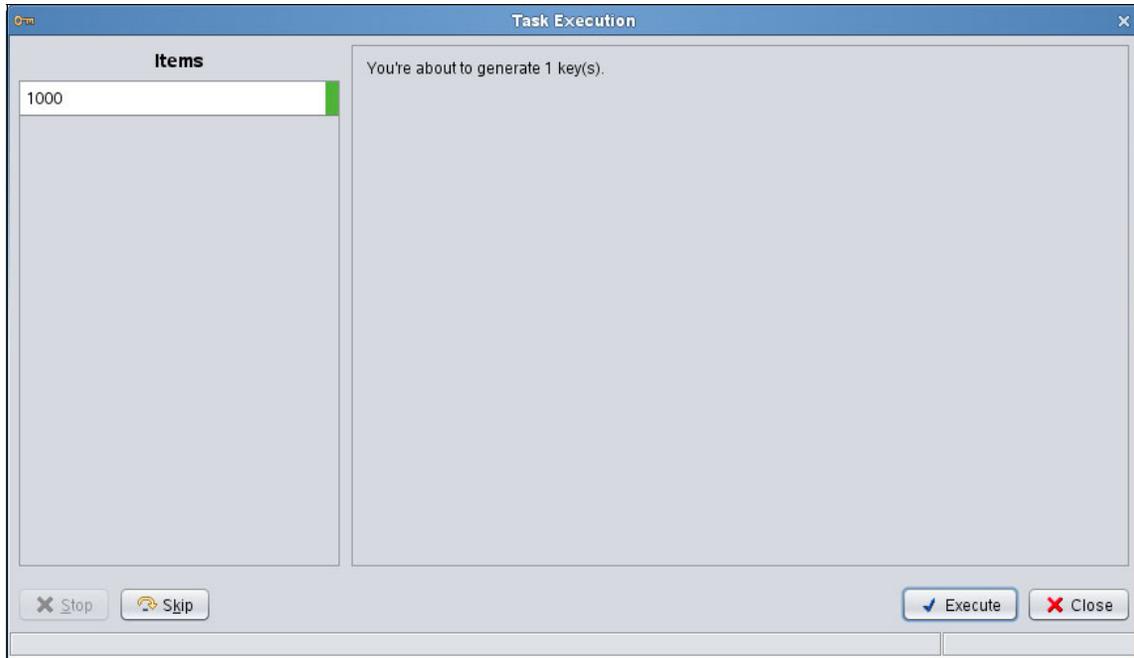


Figure 5-77 Items to be executed in a task

3. Click **Execute**. The result of the task is shown in Figure 5-78.

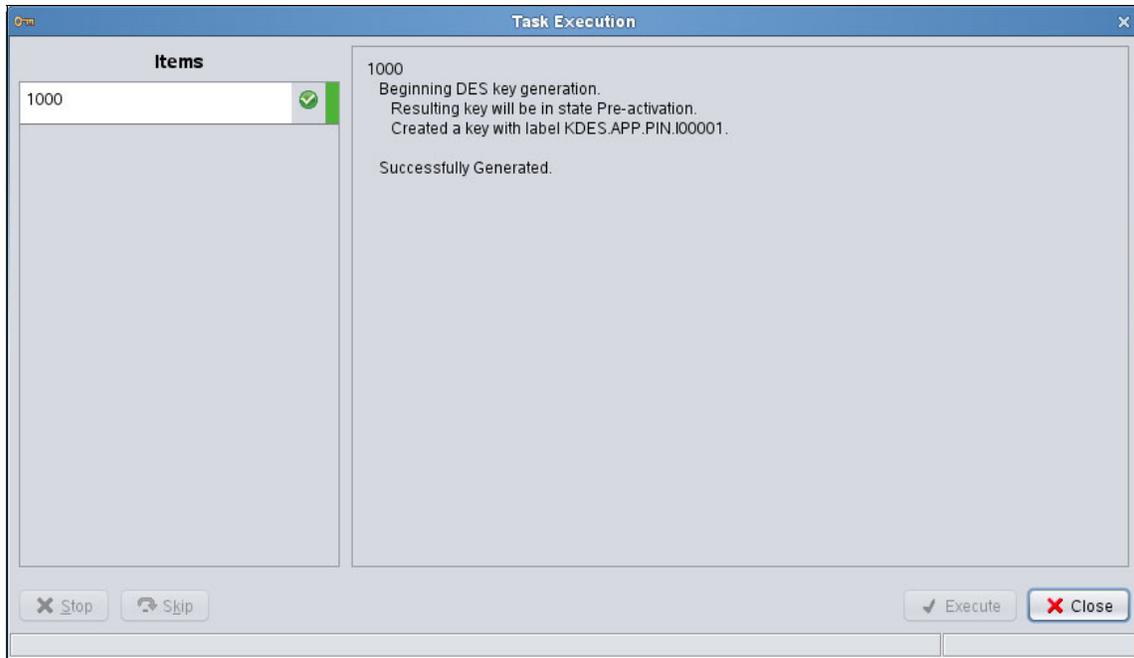


Figure 5-78 Result of executing a task

4. Click **Close**. The key is now generated.

## 5.6 Key lifecycle management

In IBM Enterprise Key Management Foundation, the key lifecycle must follow the NIST standard.

Complete the following steps:

1. Use **PROG0323 Symmetric key management** to control the lifecycle. Specify a search criteria to find the 1000 system key. Select **All** in Key State. Click **Search**. The search result is shown in Figure 5-79.

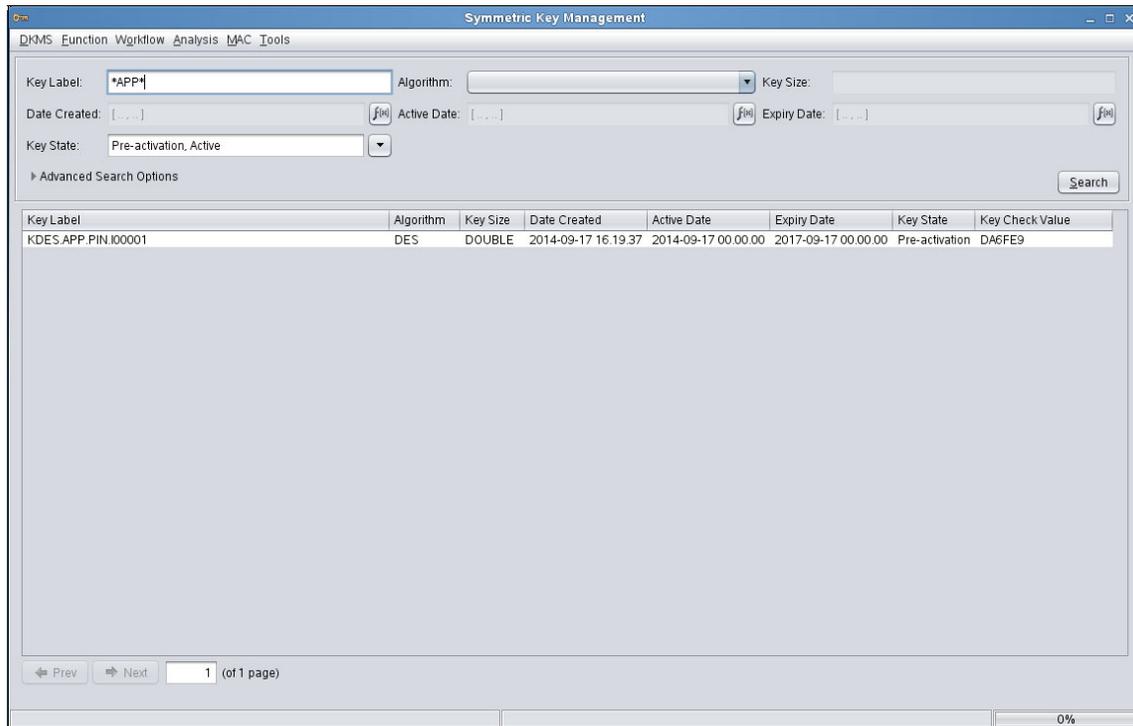


Figure 5-79 Result of the key search

- The Key State is Pre-Active. Double-click the key and click the **Key Instance** tab (Figure 5-80).

The screenshot shows the 'Key Details' dialog box with the 'Key Instance' tab selected. It displays a table of key instances and various configuration fields.

| Application | Key Store Label       | Key Zone          | Key Store Type | Install |
|-------------|-----------------------|-------------------|----------------|---------|
| ISSUER      | KDES.APP.PIN.PG00...  | I - ICSF          | ICSF           | Yes     |
| AUTHORIZ    | KDES.APP.PIN.PV000... | A - Authorization | ICSF           | Yes     |

Below the table, the following fields are visible:

- Application: ISSUER - ISSUER
- Key Store Label: KDES.APP.PIN.PG00001
- Key Zone: I - ICSF
- Key Store Type: ICSF
- Key Check Method: 8: ENC-ZERO
- Key Check Value: DA6FE9
- Active Date\*: 2014-09-17 00:00:00
- Expiry Date\*: 2017-09-17 00:00:00
- Control Vector: 0022410003410000 0022410003210000
- Key Type: PINGEN
- KEK: KIZKDES.KEYMNGNT.IZK.ZONEI2.KE
- KEK Location: UKDS7

Buttons for 'Save' and 'Cancel' are located at the bottom right.

Figure 5-80 Key instances of the 1000 key

3. The key is specified to be installed. Click the **KeyStore Details** tab, as shown in Figure 5-81.

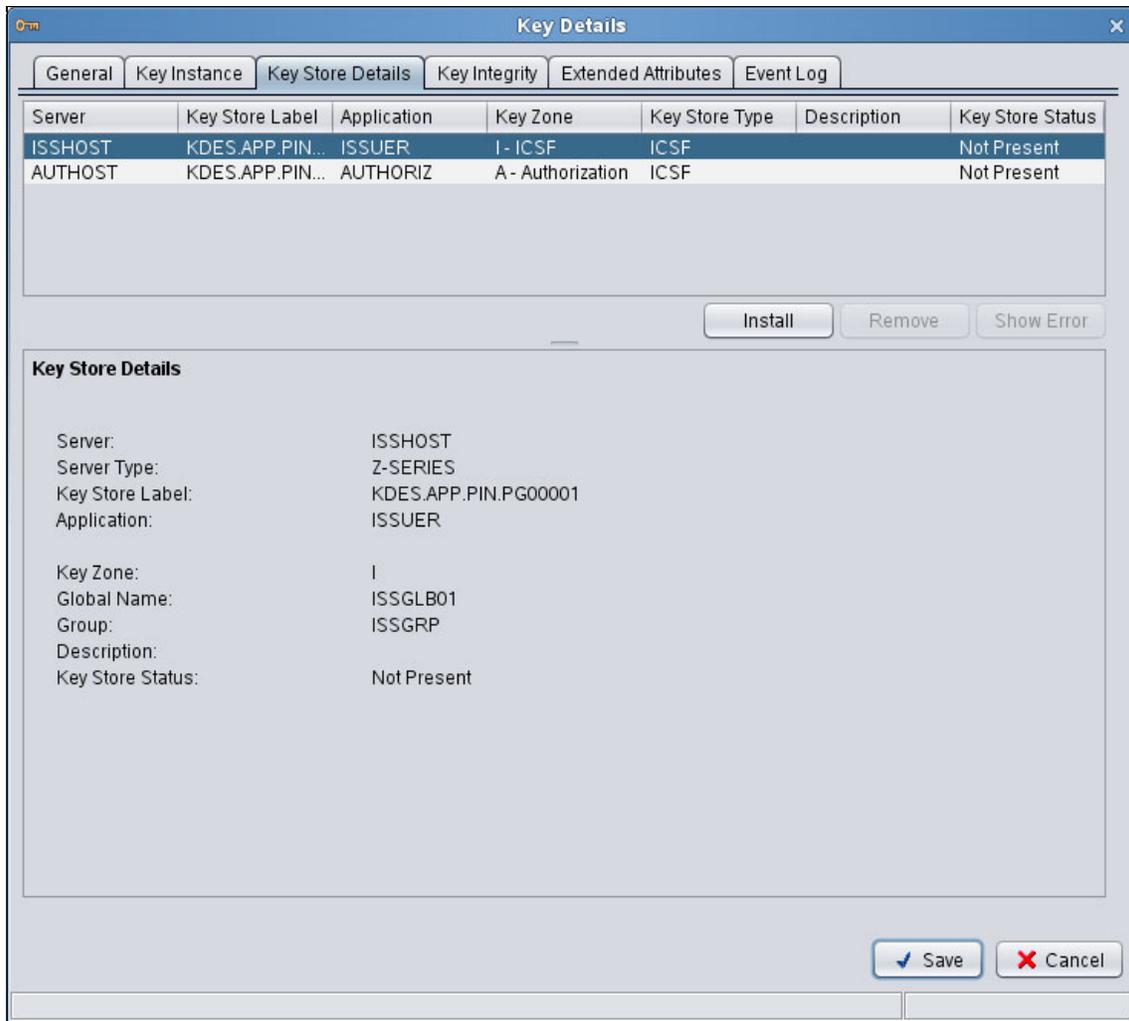


Figure 5-81 Keystore details for the 1000 key

The key is not present in the keystores because it is not active. It is possible to install the key without changing the key state by clicking **Install**.

4. Click **Cancel**. In Symmetric Key Management, click **Function** → **Activate Key** to activate the key. An activate report is shown in Figure 5-82.

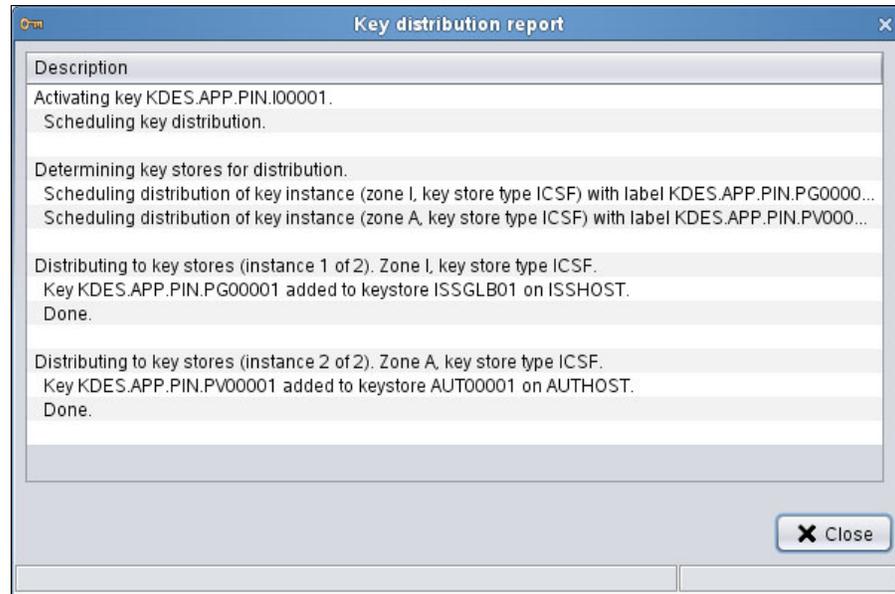


Figure 5-82 Activate report

5. Details for the key now show that the key is present in both keystores. Click **Function** → **Deactivate Key** to deactivate the key. A deactivate report is shown in Figure 5-83.

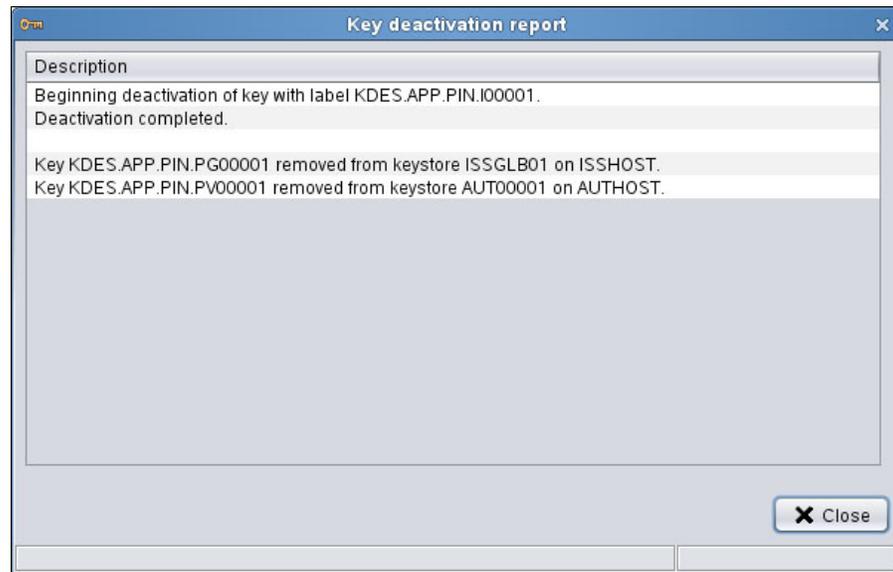


Figure 5-83 Deactivate report

- The key is now removed from both keystores. It is possible to manually install the key from the key details dialog box. Click **Function** → **Destroy key**. Click **Destroy** to confirm. A key destroy report is shown in Figure 5-84.

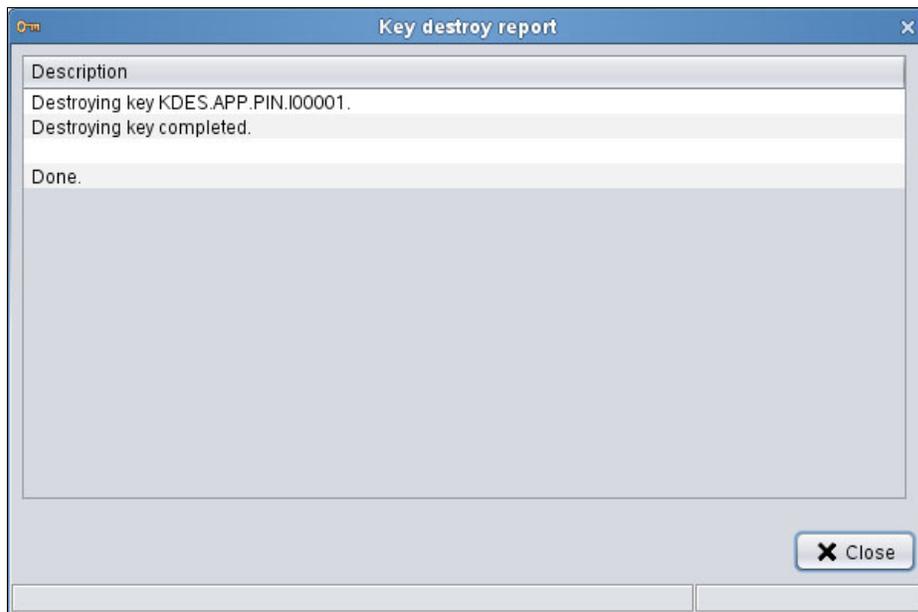


Figure 5-84 Key destroy report

The key value is now removed from the database, and it is no longer possible to install the key in the keystores from the key details.

## 5.7 Conclusion

We deployed the basic key management infrastructure at Fictional Bank, and a Key Management Workstation is in place, including a securely configured IBM PCIe 4765 Cryptographic Coprocessor and a basic DES key hierarchy. Host Agents are in place to enable distribution of keys to z/OS LPARs. Finally, we generated the first application key. The administrators can now turn over the system to the key managers so that they can manage keys for the applications.

The typical application that can be supported by this setup is magnetic stripe-based payment cards. This includes both issuing of cards and routing and verification of transactions.

The next step for Fictional Bank is to meet the extended key management requirements that are listed in 4.3.3, “Extended key management requirements” on page 96. To accomplish this task, the following specific features must be enabled:

- ▶ Configure RSA keys.
- ▶ Configure ATM key management.
- ▶ Issue EMV cards.
- ▶ Set up workflows and reporting.
- ▶ Configure certificate management.

A detailed explanation about implementing these features is not part of this book; such an explanation *might* be added to a future edition.





# A

## Troubleshooting

This appendix describes some troubleshooting issues in IBM Enterprise Key Management Foundation. Common problems and solutions for each component are described, and how to enable trace and what to expect from the trace output.

This appendix includes the following sections:

- ▶ EKMF workstation
- ▶ EKMF agents
- ▶ CCA Node Management Utility

## EKMF workstation

During the setup and usage of the Key Management application (DKMS), there are a few common situations where a wrong configuration leads to a blocking error. Table A-1 describes some of these situations, and how to solve them.

Table A-1 Common problems and solutions for the EKMF application

| Error description                                                                                                       | Reason                                                                                                | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| When starting the DKMS application, the logon dialog box shows an empty list of profiles                                | No profile exists in the IBM 4765, which has access to the Access Control Point (ACP) 8001 DKMS User. | <ol style="list-style-type: none"> <li>1. In the CCA Node Management Utility program, create a role that permits the ACP 8001 DKMS User.</li> <li>2. Create a profile that belongs to this role.</li> <li>3. Restart the DKMS application.</li> </ol>                                                                                                                                                                                           |
| When starting the DKMS application for the first time, an error is displayed: No functions accessible for current user! | No configuration is loaded in to the DKMS application configuration database.                         | <ol style="list-style-type: none"> <li>1. Stop the IBM Enterprise Key Management Foundation application.</li> <li>2. Delete all the content of the <code>/var/opt/ibm/dkms/&lt;environment-name&gt;/db</code> directory.</li> <li>3. Copy the configuration containing *.DEL files and the features.dat file to the <code>/var/opt/ibm/dkms/&lt;environment-name&gt;/table</code> directory.</li> <li>4. Start the DKMS application.</li> </ol> |

| Error description                                                                                                                                                                                                         | Reason                                                                                                                                                                                                                                           | Solution                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>When the DKMS application for the first time, it is not possible to select a host type, and an error is displayed:<br/>           Error Type: 84/DKMS-SOC TCP/IP DLL errors<br/>           Return code: 21</p>         | <p>Feature codes are not loaded for a database host, and it is possible that no feature codes are loaded at all.<br/>           The feature code for using a database on a mainframe is not loaded. Eventually, no feature codes are loaded.</p> | <ol style="list-style-type: none"> <li>1. Stop the DKMS application.</li> <li>2. Copy features.dat to the <code>/var/opt/ibm/dkms/&lt;environment-name&gt;/table</code> directory.</li> <li>3. Start the DKMS application.</li> </ol>                                                                                                                                                            |
| <p>When starting the DKMS application, an error is displayed:<br/>           Error Type: 41/Error from the DKMS Ws modules<br/>           Return code 5</p>                                                               | <p>The EKMF Agent agent is not running on the IP address and port that is specified in the settings.</p>                                                                                                                                         | <p>Do either of the following tasks:</p> <ul style="list-style-type: none"> <li>▶ Start the EKMF Agent on the IP address and port that are specified in the settings.</li> <li>▶ If the DKMS application is started under an ADMIN user, you can change the address/port settings in the <b>host TCP/IP Communication settings</b> menu to the ones to which EKMF Agent is listening.</li> </ul> |
| <p>When starting the DKMS application, an error is displayed:<br/>           Error Type: 11/DKMS/J<br/>           Return code: 6<br/>           or<br/>           Error Type: P2/DKMS/J<br/>           Return code: 6</p> | <p>The EKMF Agent closed the connection, most likely because a wrong code page was selected.</p>                                                                                                                                                 | <p>In the DKMS application <b>host TCP/IP Communication settings</b> menu, select the code page that is used on the server where the EKMF Agent runs. Select <b>UTF-8</b> if the server is not a System z server.</p>                                                                                                                                                                            |
| <p>Logon to a host is rejected because of an invalid user name/password, even if the user name/password is correct.</p>                                                                                                   | <p>The DKMS application converts the password to uppercase before sending it to the host.</p>                                                                                                                                                    | <p>Change the password on the host to one where all letters are in uppercase. The maximum length is 8.</p>                                                                                                                                                                                                                                                                                       |

| Error description                                                                                                       | Reason                                                                                                                                                                                                                                                                                                               | Solution                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>When starting the DKMS application, an error is displayed:<br/>Difference in CVI MAC and audit CVI MAC value</p>     | <p>The environment was changed outside the control of the DKMS application. There might be possible changes in the local database where the DKMS application stores part of the configuration. This message appears the first time that you start the DKMS application as well; in this case, you can ignore it.</p> | <ol style="list-style-type: none"> <li>1. Identify which changes were performed, and make sure that there is a valid reason for the changes.</li> <li>2. Start the DKMS application under a user profile name that starts with ADMIN and click <b>Yes</b> when asked to update the audit MAC values.</li> </ol>                                                                                                                     |
| <p>When starting CONF0041 Key Templates, an error is displayed:<br/>Error validating Preferences<br/>MAC Service...</p> | <p>A JDBC Connection is not established to the DB2 database on the host.</p>                                                                                                                                                                                                                                         | <ol style="list-style-type: none"> <li>1. Close the dialog box and click <b>Tools</b> → <b>Remote Database Settings</b>.</li> <li>2. Enter the correct settings to connect to DB2 on the server. The connection is tested when <b>Save</b> is clicked. It is not possible to save the settings if the test fails. It is possible to get information about the expected inputs by moving the mouse over the input fields.</li> </ol> |

| Error description                                                                                                                       | Reason                                                                                             | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>When trying to leave insecure mode in CONF0041 Key Template, an error is displayed:<br/>No active keys found for Key Template...</p> | <p>The MAC keys that are used to protect data in the database are not present in the database.</p> | <p>To generate the MAC keys by using the key templates that are specified in <b>Tools</b> → <b>Settings</b> → <b>MAC Service</b>, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Select the keys in <b>PROG0323 Symmetric Key Management</b>.</li> <li>2. Click <b>MAC</b> → <b>Generate MAC</b> → <b>Selected Key</b>.</li> <li>3. Click <b>Tools</b> → <b>Settings</b> → <b>MAC Service</b>.</li> <li>4. Clear the <b>Insecure Mode</b> check box.</li> </ol> |
| <p>Creating a key in PROG0323 Symmetric Key Management results in an error:<br/>CCA Verb=Unknown<br/>return=8 reason=23xxx</p>          | <p>The key template contains a forbidden combination of key instances.</p>                         | <p>Modify the key template in CONF0041 Key Templates.<br/>The key type and key encryption key for each instance are necessary to create the key without errors.<br/>Both “Key Instances” and “Export Key Instances” counts as instances for the key.</p>                                                                                                                                                                                                                                     |

| Error description                                                                                                     | Reason                                                                                                                                                                                                                                                                       | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>The Key Store Details tab for a key in PROG0323 Symmetric Key Management does not show the expected keystores.</p> | <p>To show the key status in a specific keystore, the key must have a key instance that matches the keystore that is defined in the device configuration.</p>                                                                                                                | <p>Use PROG0310 Device Configuration to review the device configuration. In PROG0323 Symmetric Key Management, click the <b>Key Instance</b> tab in Key Details for the selected key to verify that the key has a key instance that matches the keystore settings in the device configuration. To match, the Algorithm, Application, and Key Zone must be the same in the key instance and in the keystore configuration.</p>                                                                                                                                                   |
| <p>A key is expected to be present in a keystore, but the Key Store Status is Not Present.</p>                        | <p>One of the following reasons are possible:</p> <ol style="list-style-type: none"> <li>1. The key state of the key is not Active.</li> <li>2. The key is marked as not to be installed.</li> <li>3. It was not possible to install the key when it was created.</li> </ol> | <p>Each case has its own solution:</p> <ul style="list-style-type: none"> <li>▶ For the first one, select the key in PROG0323 Symmetric Key Management and click <b>Function</b> → <b>Activate Key</b>. This distributes the key to all matching keystores.</li> <li>▶ For the second one, modify the key template for the key creation and click <b>Install: Yes</b> for the relevant key instances. Create the key again.</li> <li>▶ For the third one, open the key click the <b>Key Store Details</b> tab, mark the relevant keystore, and click <b>install</b>.</li> </ul> |

## Tracing problems

To enable trace in the DKMS application, select the **Enable Trace** check box in the window that opens after the DKMS application starts, as shown in Figure A-1.

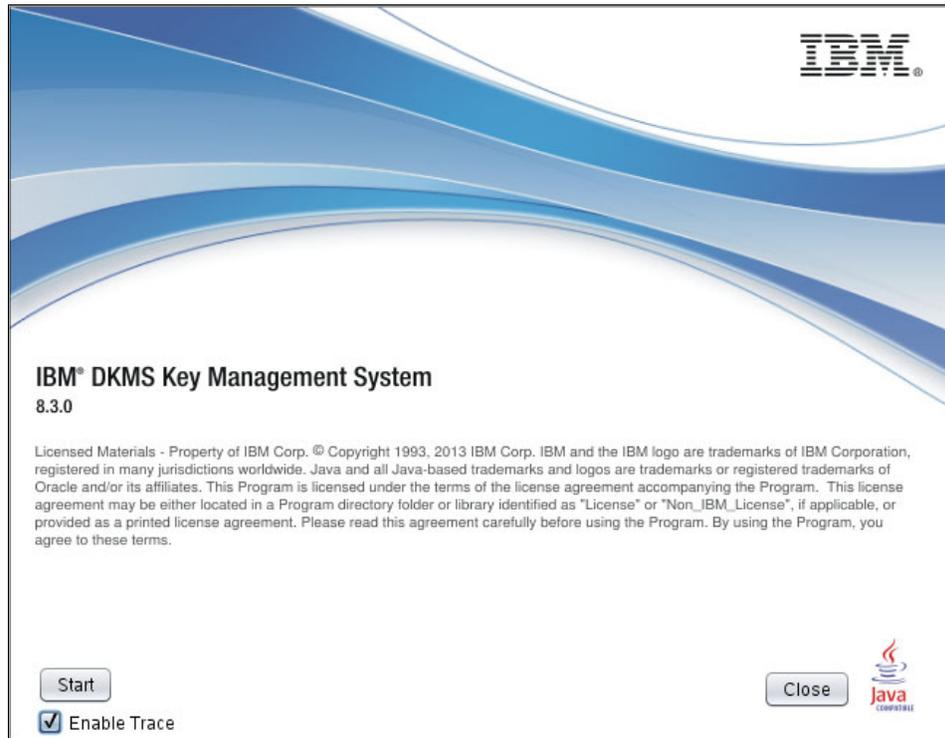


Figure A-1 Enable trace in the DKMS application

The trace files are placed in the trace directory for the started environment (in the `/var/opt/dkms/<environment>/trace` folder).

The trace files contain information from the different components in the DKMS application. When contacting IBM for help, it is important to include all files from this directory.

## EKMF agents

Two different implementations of agents exist in IBM Enterprise Key Management Foundation: One for z/OS on System z and one for other platforms. The setup and troubleshooting are different for the two implementations.

### Agents on z/OS

The EKMF Agent connects to DB2 through the DSNALI interface. The return and reason code is shown if there is an error, as shown in Table A-2.

Table A-2 Common problems and solutions with the EKMF agent on z/OS

| Error description                                                                                                                                                 | Reason                                                                                                                                                                                                                                      | Solution                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| Failed to connect to a DB2 system.<br><br>Error Code:<br><br>DKMS300E DSNALI FAILED TO CONNECT TO DB2 SYSTEM DSNB RETURN CODE: 0000000012 REASON CODE: 0015925254 | The reason code 0015925254 is X'00F30006' and can be found in the following publication:<br><i>DB2 10 for z/OS Codes, GC19-2971</i>                                                                                                         | See <i>DB2 10 for z/OS Codes, GC19-2971</i> .                                                    |
| Failed to access ICSF.<br><br>Error Code:<br><br>DKMS014E FAILED TO ACCESS ICSF - CSFKGN FUNCTION FAILED.<br>Return code: 0000000012<br>Reason code: 0000000000   | If you are using the parameter <b>&amp;CRYPTO-ENGINE (CCF)</b> in the option data set, the EKMF Agent checks whether ICSF is operational and issues a key generate verb. IF ICSF is not operational, the Agent task ends with this message. | Verify that ICSF is running. Start ICSF and the EKMF Agent on the mainframe again, if necessary. |

| Error description                                                                                                                                                                                                                               | Reason                                                                                                                                                                                                                                        | Solution                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>IP stack TCPIP.DATA is missing.</p> <p>Error Code:</p> <p>DKMS007E TCP/IP ERROR.<br/>SOCKET FUNCTION:<br/>INITAPI. SOCKET ERROR:<br/>0000001011<br/>TCP/IP HOSTNAME :<br/>UNKNOWN, ERRNO =<br/>0000001011.</p>                               | <p>You get this message if TCPIP.DATA is not available in general, or you might have several IP stacks and the one that you use for the connection is not defined.</p>                                                                        | <p>You must add a SYSTCPD DD-name to the started task procedure to connect to the specific IP stack and start the EKMF Agent for z/OS again.</p>                                                                                                                                                     |
| <p>APF authorization missing for a module.</p> <p>Error Codes:</p> <p>+DKMS050E KMGPRACF<br/>AUTH. PROGRAM NOT FOUND<br/>IN AN AUTH. LIBRARY.</p> <p>DKMS201E<br/>TSOLNK-KMGPRACF ERROR.<br/>CC1=FFFFFFFF<br/>CC2=00000038<br/>CC3=FFFFFFFF</p> | <p>The EKMF Agent for z/OS is delivered with several modules that must be APF-authorized (linked with AC(1). If you use a STEPLIB concatenation, where a library is not defined in the APF list, then the Agent ends with these messages.</p> | <p>You can choose to APF-authorize all the libraries in the STEPLIB concatenation, or move the APF module to a common link list library, so that they are not picked up from the STEPLIB concatenation. Also, the APF modules must be defined as AUTHTSF modules in the IKJTSoxx parmlib member.</p> |

## EKMF Agents on other platforms

The EKMF Agent is designed to run on various platforms and in various configurations. The most common problems are related to the configuration of the EKMF Agent in the specific environment.

When the EKMF Agent starts, it performs a self test of the configuration. When no errors are encountered, the EKMF Agent runs as a service on the system and waits for requests; otherwise, it exits and writes error messages to sysout. If the EKMF Agent is started from a command line, the information is shown in the terminal window, and if the agent was started as a service, the log can be found in `/var/log/dkmsagent.log`.

## CCA Node Management Utility

The CCA Node Management (CNM) utility is the management application for the IBM PCIe 4765 Cryptographic Coprocessor that is installed in the EKMF workstation.

The IBM Enterprise Key Management Foundation can manage several different crypto servers, which have one or more IBM 4765 coprocessors that are installed. If the CNM utility is used to configure the IBM 4765, some of the messages that are described in Table A-3 might display.

Table A-3 Common problems and solutions with the CNM utility

| Error description                                                                                                                         | Reason                                                                                                                                                                                                                                                                                     | Solution                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nothing happens when the <code>csu1c1u</code> program is run from <code>/opt/IBM/4765/cnm/</code> on Linux platforms.                     | The CNM utility requires Java to run.                                                                                                                                                                                                                                                      | Install Java on the machine, or point to the Java version that is delivered with IBM Enterprise Key Management Foundation. For example, include the following line in the <code>csu1c1u</code> file:<br><code>PATH=\$PATH:/opt/ibm/dkm/sagent/_jvm/jre/bin</code>                                               |
| When starting the CNM utility, an error is displayed: Not able to open <code>csuap.def</code> file. Utility function will be limited.     | <code>csu1cnm</code> must be run from the following folder:<br><code>/opt/IBM/4765/cnm</code>                                                                                                                                                                                              | Start the CNM utility from a terminal by running the following commands:<br><ul style="list-style-type: none"> <li>▶ <code>cd /opt/IBM/4765/cnm</code></li> <li>▶ <code>./csu1cnm</code></li> </ul>                                                                                                             |
| An error is displayed when using almost all the functions in the CNM utility: Device Not Installed<br>Return code: 12<br>Reason code: 338 | One of the following reasons is possible: <ul style="list-style-type: none"> <li>▶ The IBM 4765 is not physically installed in the machine.</li> <li>▶ The CCA firmware is not loaded in the IBM 4765.</li> <li>▶ The machine just rebooted, and the IBM 4765 is not yet ready.</li> </ul> | Each case has its own solution: <ul style="list-style-type: none"> <li>▶ Install the IBM 4765 in a free slot.</li> <li>▶ Load the CCA firmware into the IBM 4765 by using the <code>csu1c1u</code> utility.</li> <li>▶ Wait for 2 minutes and try the function after starting the CNM utility again.</li> </ul> |

| Error description                                                                              | Reason                                                                    | Solution                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>An error is displayed when using some functions:<br/>Return code: 8<br/>Reason code: 90</p> | <p>The active profile does not have access to the requested function.</p> | <p>A profile access is limited to the IBM 4765 functions by the role that is assigned to it.<br/>If a profile is not logged on, the DEFAULT role is active. Log on using a profile that has access to the specific function or change the access rights of your profile so that you can run the function.</p> |





# B

## Operational procedures

This appendix describes the individual tasks that are involved in a secure setup of the IBM 4765 access control system for an EKMF workstation that is configured with the smart card option.

Each task description includes details about its purpose and expected outcome, outlines the organizational requirements and eventual prerequisites, and lists easy-to-follow procedural steps.

This appendix includes the following sections:

- ▶ Smart card management using Smart Card Utility Program
- ▶ IBM PCIe 4765 Cryptographic Coprocessor management using CNM
- ▶ Managing the application

# Smart card management using Smart Card Utility Program

This section covers the following topics:

- ▶ Initializing and personalizing a certificate authority smart card
- ▶ Backup CA smart card
- ▶ Enrolling the IBM PCIe 4765 Cryptographic Coprocessor
- ▶ Initializing and enrolling a TKE smart card
- ▶ Personalizing a TKE smart card
- ▶ Unblocking a TKE smart card
- ▶ Changing the PIN of a CA smart card
- ▶ Changing the PIN of a TKE smart card
- ▶ SCUP logon by using split passphrase
- ▶ SCUP group logon by using smart cards

Smart cards must be inserted into the smart card readers with the chip of the smart card facing up and then inserted down into the reader, as shown in Figure B-1.



Figure B-1 Smart card readers

## Initializing and personalizing a certificate authority smart card

The aim of this procedure is to initialize and personalize a certificate authority (CA) smart card by using the Smart Card Utility Program (SCUP), which creates a unique crypto zone.

Each CA smart card requires two PINs, which enforce dual control access to the CA smart card:

- ▶ PIN-1: A 6-digit PIN
- ▶ PIN-2: A 6-digit PIN

If you cancel the CA smart card initialization process before the card is initialized, the smart card you are initializing is left in an unusable state and must be reinitialized to be made usable again. You might have to reinitialize the smart card as a different type of smart card to erase the partially initialized state. For example, if you cancel the initialization of a CA smart card in the middle of the initialization, you might need to initialize that card as a TKE smart card before you can initialize it as a CA smart card again.

## Participants

Table B-1 lists the participants with their roles and brief descriptions.

*Table B-1 Participants*

| Role indication | Role description                                         |
|-----------------|----------------------------------------------------------|
| ADM1n           | Holder of CA smart card, and PIN-1 for the CA smart card |
| ADM2n           | Holder of PIN-2 for the CA smart card                    |
| ADM1n / ADM2n   | Responsible for registering the CA smart card            |

## Special requirements

The participants that are listed in Table B-1 need the following components:

- ▶ ADM1n must have the following components:
  - A blank smart card (or an old one that is no longer needed)
  - A label for the blank smart card
  - A pre-labelled envelope for the storage of the CA smart card
  - A PIN-1 PIN Form for the CA smart card
  - A pre-labelled envelope for the PIN-1 PIN Form
- ▶ ADM2n must have the following components:
  - A PIN-2 PIN Form for the CA smart card
  - A pre-labelled envelope for the PIN-2 PIN Form

ADM1n and ADM2n are responsible for registering the CA smart card in the inventory of smart cards. This registration should include the following items:

- ▶ The creation date of the CA smart card
- ▶ The zone ID and description
- ▶ The card ID and description
- ▶ The ADM1n identification/name
- ▶ The ADM2n identification/name
- ▶ The storage location of CA smart card and PIN-1 for the CA smart card
- ▶ The Storage location of PIN-2 for the CA smart card

**Additional credentials that are needed:** When the SCUP starts, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

## Procedure: Initializing and personalizing a CA smart card

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**), as shown in Figure B-2.

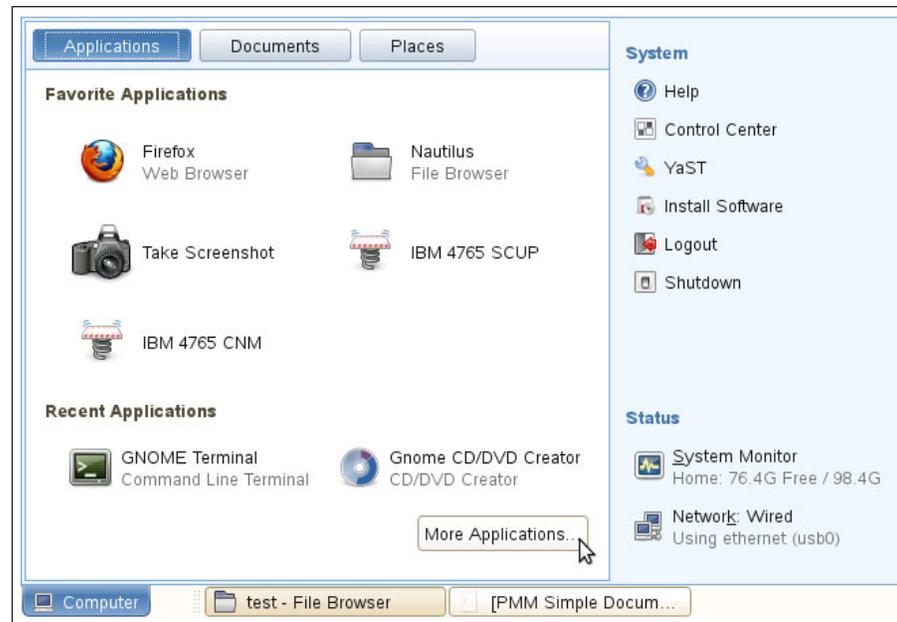


Figure B-2 Start the IBM 4765 SCUP application

2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290.
  - “SCUP group logon by using smart cards” on page 292.
3. Click **CA Smart Card** → **Initialize and Personalize CA smart card**, as shown in Figure B-3 on page 251.

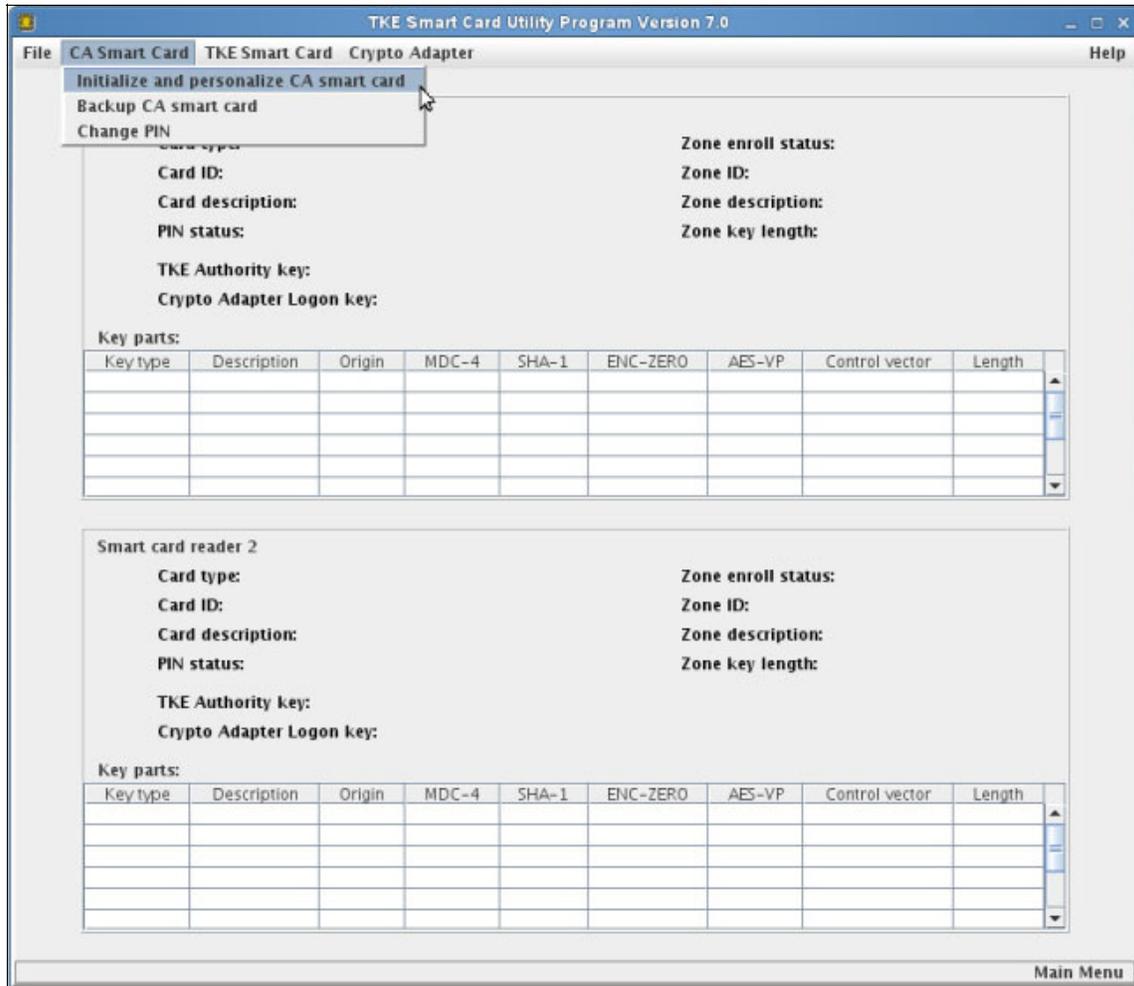


Figure B-3 Initialize and personalize a CA smart card

4. ADM1n is prompted to insert a blank smart card in reader 1. Click **OK**.

5. ADM1n is prompted to select a zone key length. Select **2048**. Click **OK**, as shown in Figure B-4.

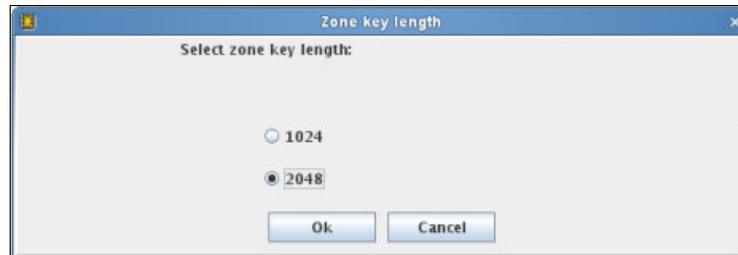


Figure B-4 Zone key length

6. A prompt opens and indicates that the initialization process is underway, as shown in Figure B-5.



Figure B-5 Initialization is processing

7. ADM1n is prompted to enter the 6-digit PIN-1 of the CA smart card *twice*.  
To enter the PIN twice, simply type it in and then type it in again, in one operation. Do not press any other keys in between. For example, if the PIN is 123456, type 123456123456.  
There is a timeout of around 30 seconds for entering information in to the card reader. If the reader times out before you complete the PIN entry, you must start over from step 2 on page 250, as shown in Figure B-6 on page 253.

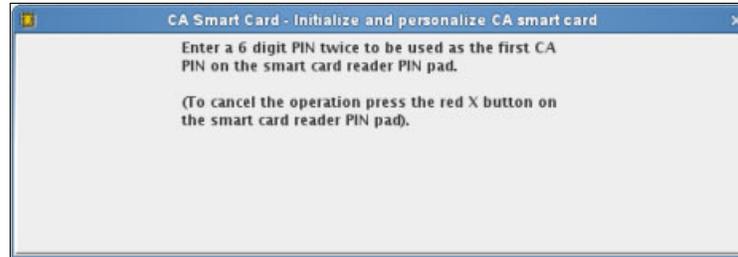


Figure B-6 Enter your PIN twice - ADM1n

8. ADM2n is prompted to enter the 6-digit PIN-2 of the CA smart card *twice*.  
If the reader times out before you complete the PIN entry, you must start over from step 2 on page 250, as shown in Figure B-7.

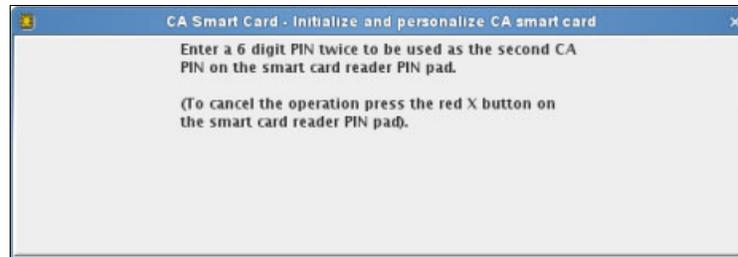


Figure B-7 Enter your PIN twice - ADM2n

9. ADM1n is prompted to enter an optional (yet recommended) zone description.  
Avoid the word *zone* in the description, as it is rather obvious in the context. Here are some example descriptions:  
CA **TEST** <company> <system>  
CA **PROD** <company> <system>

Select **OK**, as shown in Figure B-8.

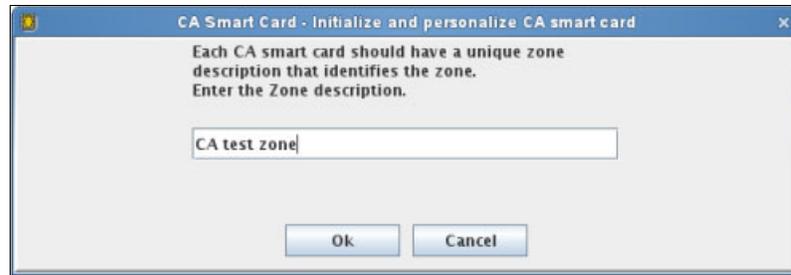


Figure B-8 Enter a zone

10. ADM1n is prompted to enter an optional description for the CA smart card. Avoid the word *card* in the description, as it is rather obvious in the context. A description that is similar to the one for the zone is recommended. Select **OK**, as shown in Figure B-9.

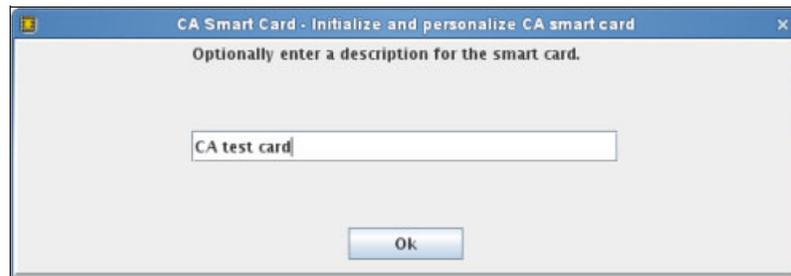


Figure B-9 Enter a description

11. A prompt displays and indicates that the CA smart card is being built, as shown in Figure B-10.

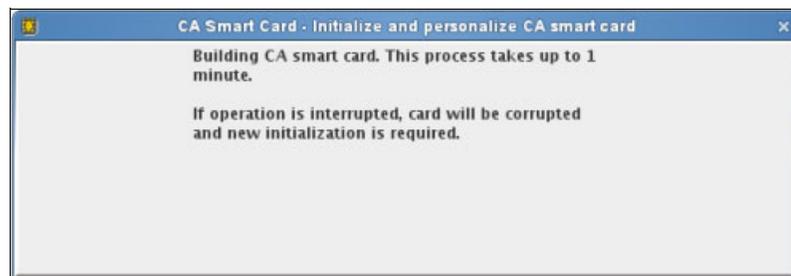


Figure B-10 Build the smart card

- 12.A prompt opens and confirms the success of the initialization and personalization of the new CA smart card. Select **OK**, as shown in Figure B-11.



*Figure B-11 Smart card created successfully*

After the process is complete, a message prompt opens showing the retrieval of the smart card information. After a few seconds, the main window opens with the information from the new CA smart card that is inserted in reader 1.

The procedure is complete, as shown in Figure B-12.

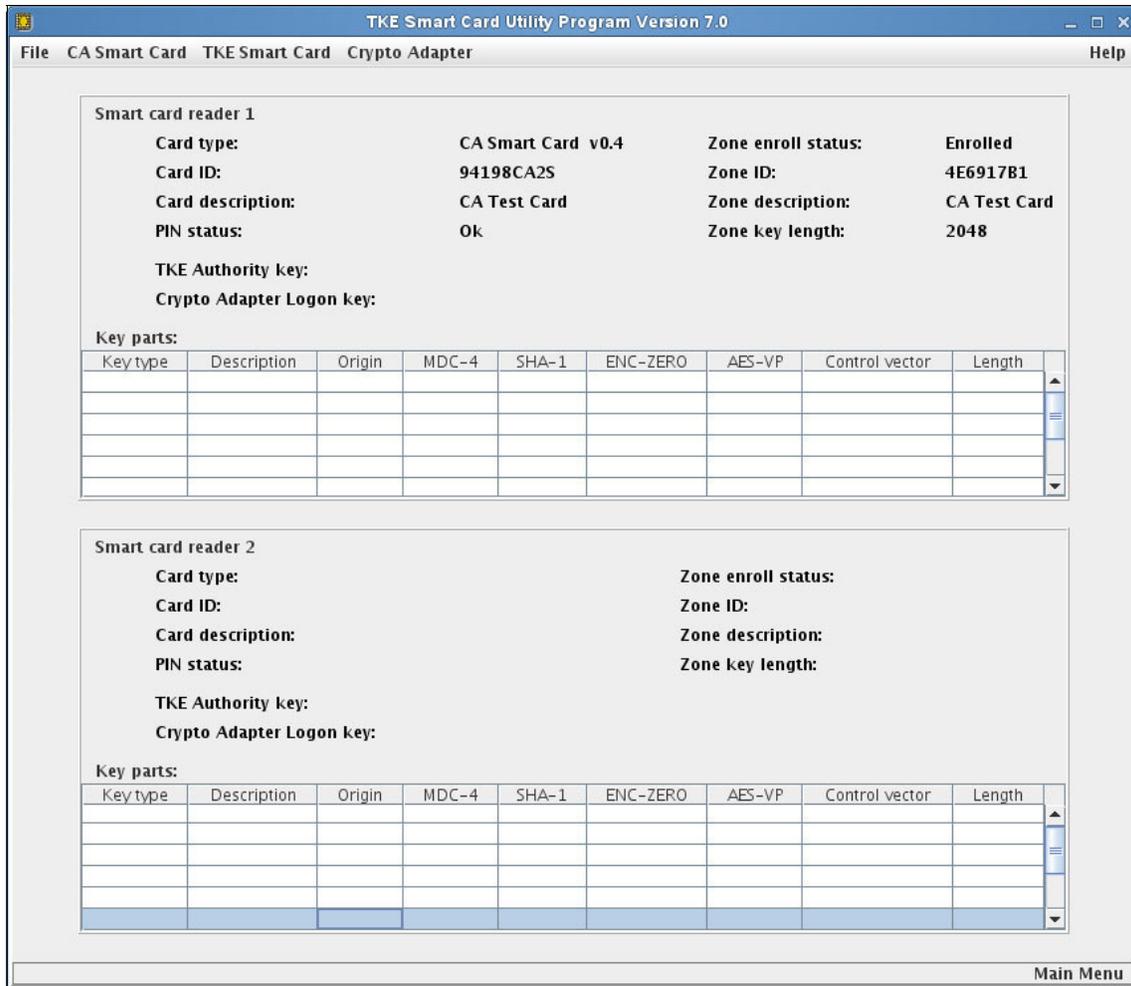


Figure B-12 Procedure complete

## Backup CA smart card

The aim of this procedure is to create a backup of the CA smart card by using the SCUP.

The IBM 4765 and all associated TKE smart cards are digitally signed by the CA smart card (that is, enrolled into the crypto zone of the CA smart card).

If the CA smart card is lost, the current operational set of TKE smart cards that is used for storing IBM 4765 master key parts can be used only with an IBM 4765 that is already enrolled into the same crypto zone. Also, the PIN reset function, which requires the presence of the CA smart card, is not available to any of the TKE smart cards that are created in the crypto zone.

**Blocking your smart card:** A CA smart card itself cannot be unblocked if the maximum number of PIN failures is reached. So, you have up to 10 PIN trials, on each of the two PINs, before the CA smart card is blocked permanently, compared to the maximum of three PIN trials on TKE smart cards.

Keep at least one backup of the CA smart card in a secure location.

You can use different PINs with the backup CA smart card, but it is preferable that you use the exact same PINs as for the primary CA smart card.

## Participants

Table B-2 lists the participants with their roles and a brief description.

*Table B-2 Participants*

| Role indication | Role description                                          |
|-----------------|-----------------------------------------------------------|
| ADM1n           | Holder of CA smart cards and PIN-1 for the CA smart cards |
| ADM2n           | Holder of PIN-2 for the CA smart cards                    |
| ADM1n / ADM2n   | Responsible for registering the backup CA smart card      |

## Special requirements

The participants that are listed in Table B-2 need the following components:

- ▶ ADM1n must have the following components:
  - An envelope containing the primary CA smart card
  - An envelope containing the PIN-1 PIN Form for the primary CA smart card
  - A blank smart card (or an old one that is no longer needed)
  - A label for the blank smart card
  - A pre-labelled envelope for storage of the backup CA smart card
  - A PIN-1 PIN Form for the backup CA smart card
  - A pre-labelled envelope for the PIN-1 PIN Form
- ▶ ADM2n must have the following components:
  - An envelope containing the PIN-2 PIN Form for the primary CA smart card
  - A PIN-2 PIN Form for the backup CA smart card
  - A pre-labelled envelope for the PIN-2 PIN Form

ADM1n and ADM2n are responsible for registering the backup CA smart card in the inventory of smart cards. This registration should include the following items:

- ▶ The creation date of the backup CA smart card
- ▶ The zone ID and description (inherited from the primary CA smart card)
- ▶ The card ID of the primary CA smart card
- ▶ The card ID of the backup CA smart card
- ▶ The card description of the backup CA smart card (inherited from the primary CA smart card)
- ▶ The ADM1n identification/name
- ▶ The ADM2n identification/name
- ▶ The storage location of the backup CA smart card, and PIN-1 for the backup CA smart card
- ▶ The storage location of PIN-2 for the backup CA smart card

**Additional credentials that are needed:** When the SCUP starts, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

### **Procedure: Backing up a CA smart card**

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290.
  - “SCUP group logon by using smart cards” on page 292.

3. Click **CA Smart Card** → **Backup CA smart card**, as shown in Figure B-13.

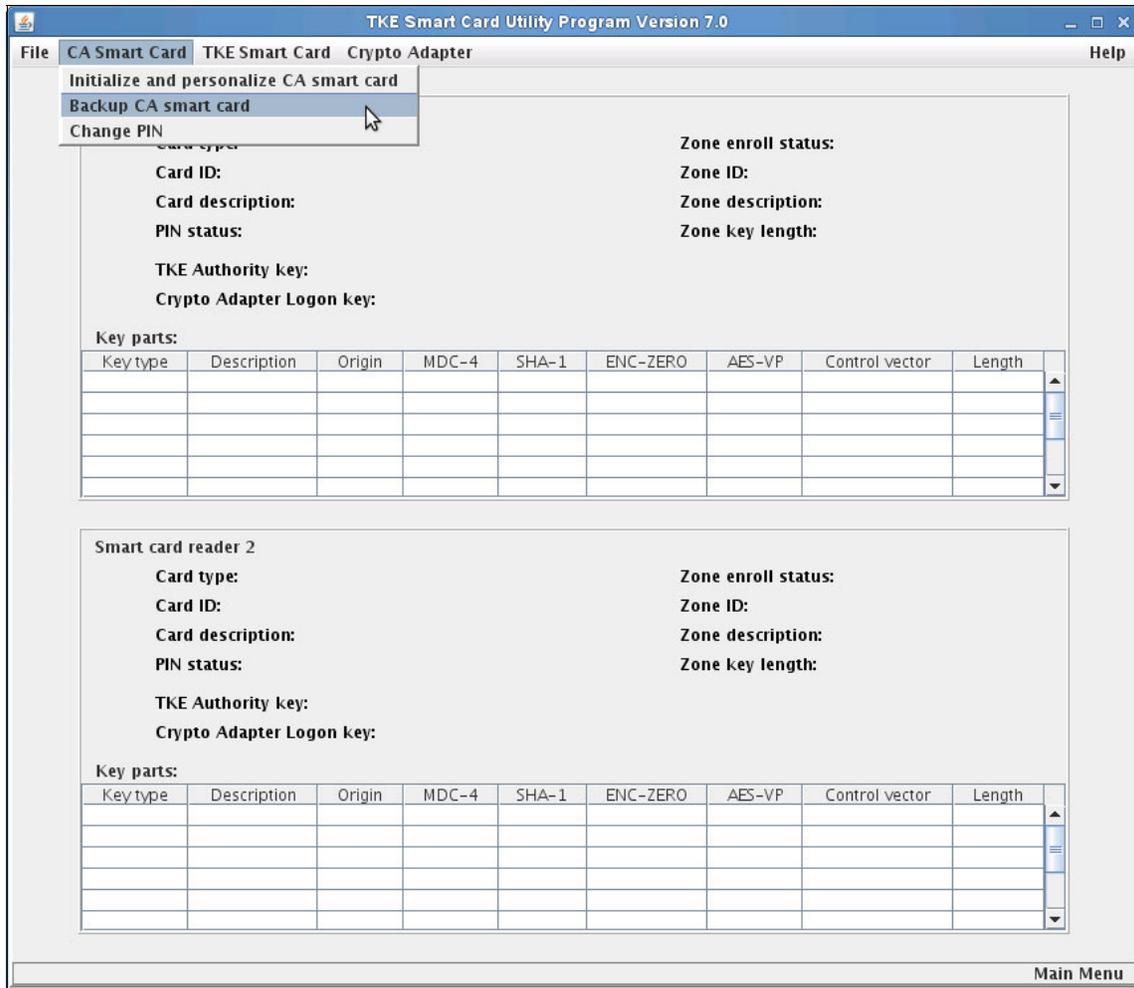


Figure B-13 Select Backup CA smart card

4. ADM1n is prompted to insert the primary CA smart card in to reader 1. Select **OK**, as shown in Figure B-14.

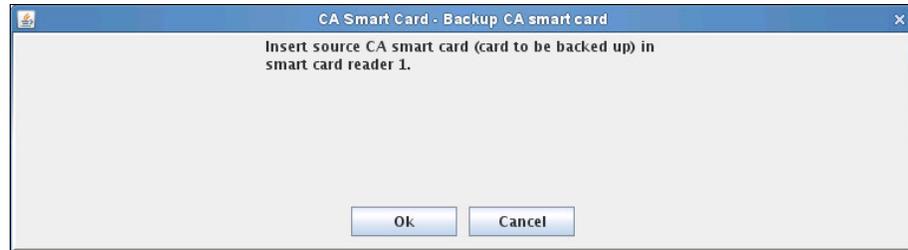


Figure B-14 Insert a smart card

5. ADM1n is prompted to enter the 6-digit PIN-1 of the primary CA Smart Card on reader 1.
6. ADM2n is prompted to enter the 6-digit PIN-2 of the primary CA smart card on reader 1.
7. ADM1n is prompted to insert the blank smart card in reader 2. Select **OK**, as shown in Figure B-15.

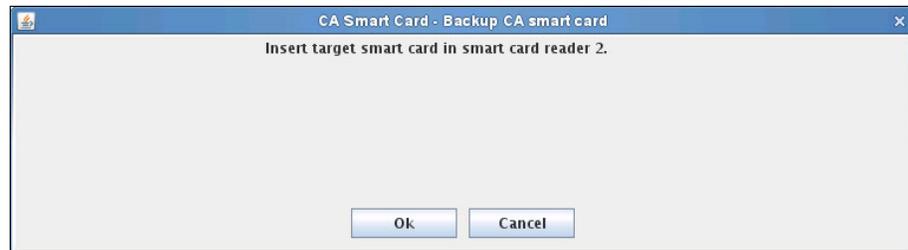


Figure B-15 Insert a blank smart card

8. A prompt opens and indicates that the initialization process is underway. After 10 - 30 seconds, a second prompt opens and indicates that the Prepare Smart Cards for Backup procedure is underway, as shown in Figure B-16 on page 261.

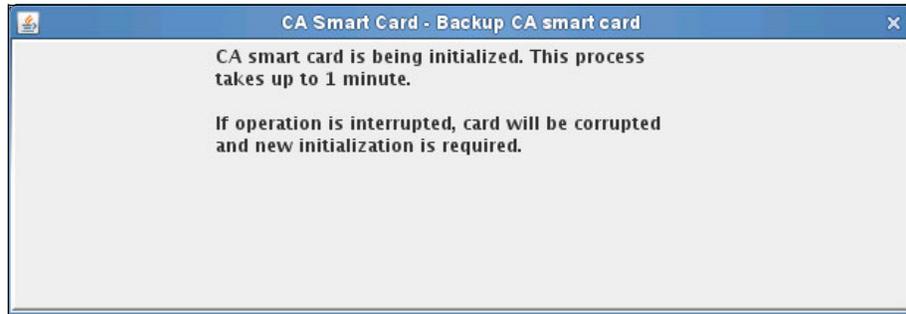


Figure B-16 The smart card is initialized

9. ADM1n is prompted to enter the 6-digit PIN-1 on reader 2.  
The PIN-1 *must* be the same as for the original CA smart card.
10. ADM2n is prompted to enter the 6-digit PIN-2 on reader 2.  
The PIN-2 *must* be the same as for the original CA smart card.
11. A prompt indicates that the card building backup process is underway, as shown in Figure B-17.

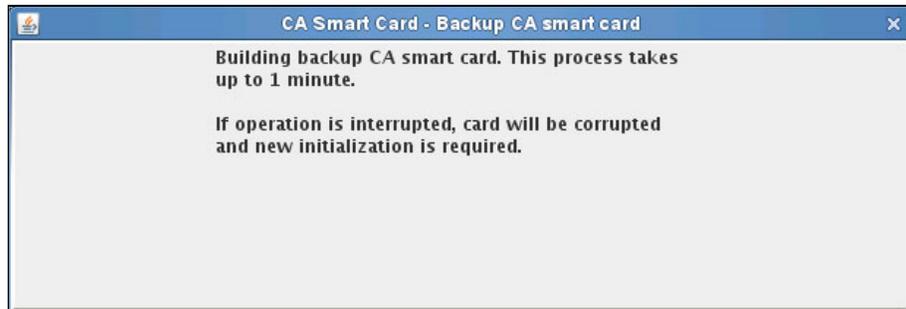


Figure B-17 Backup in progress

12. A prompt confirms the success of the backup, as shown in Figure B-18.



Figure B-18 Success

After the process is complete, a message prompt opens showing the retrieval of the smart card information. After a few seconds the main window opens with the information from the primary CA smart card that is inserted in reader 1 and the backup CA smart card that is inserted in reader 2, as shown in Figure B-19.

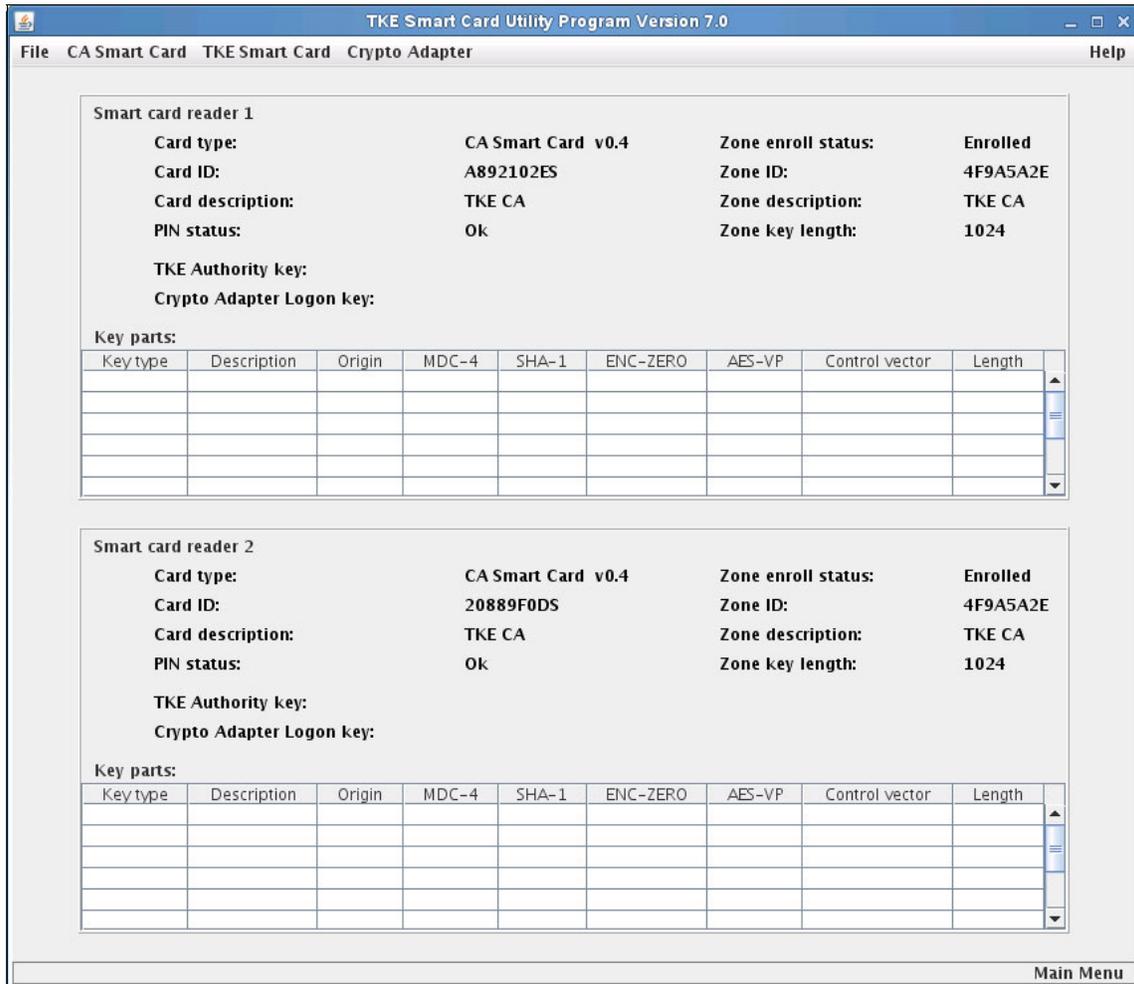


Figure B-19 Procedure complete

## Enrolling the IBM PCIe 4765 Cryptographic Coprocessor

The aim of this procedure is to enroll the IBM PCIe 4765 Cryptographic Coprocessor in to the cryptographic zone of the CA smart card by using the SCUP.

## Participants

Table B-3 lists the participants with their roles and a brief description.

Table B-3 Participants

| Role indication | Role description                                                                                                                            |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| ADM1n           | Holder of CA smart card, PIN-1 for the CA smart card, and either a first passphrase part of the CNMADMIN profile or ADM1n logon smart card. |
| ADM2n           | Holder of PIN-2 for the CA smart card, and either the last passphrase part of CNMADMIN profile or ADM2n logon smart card.                   |

## Special requirements

The participants that are listed in Table B-3 need the following components:

- ▶ ADM1n must have the following components:
  - An envelope containing the CA smart card
  - An envelope containing the PIN-1 PIN Form for the CA smart card
  - Either of these two logon credentials:
    - An envelope containing the *first* passphrase part of the (initial/temporary) CNMADMIN profile
    - An envelope containing the ADM1n logon smart card, and an envelope containing the PIN Form for the logon smart card.
- ▶ ADM2n must have the following components:
  - An envelope containing the PIN-2 PIN Form for the CA smart card,
  - Either of these two logon credentials:
    - An envelope containing the *last* passphrase part of the (initial/temporary) CNMADMIN profile,
    - An envelope containing the ADM2n logon smart card, and an envelope containing the PIN Form for the logon smart card.

## Access-control commands

Table B-4 lists the CCA access-control commands (access-control points) that are necessary to run the procedure.

Table B-4 CCA access-control commands

| Offset  | Command name                   |
|---------|--------------------------------|
| X'0103' | PKA96 PKA Key Generate         |
| X'0203' | Delete Retained Key            |
| X'0230' | List Retained Key              |
| X'02A5' | Import Card Device Certificate |
| X'02A6' | Import CA Public Certificate   |
| X'02A8' | Delete Device Retained Key     |
| X'02A9' | Export Card Device Certificate |
| X'02AA' | Export CA Public Certificate   |
| X'8002' | SCUP User Login                |

These access-control points must be enabled in the active role of the IBM PCIe 4765 Cryptographic Coprocessor when you perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor.

### Procedure: Enrolling the Crypto Adapter

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290.
  - “SCUP group logon by using smart cards” on page 292.

3. Click **Crypto Adapter** → **Enroll Crypto Adapter**, as shown in Figure B-20.

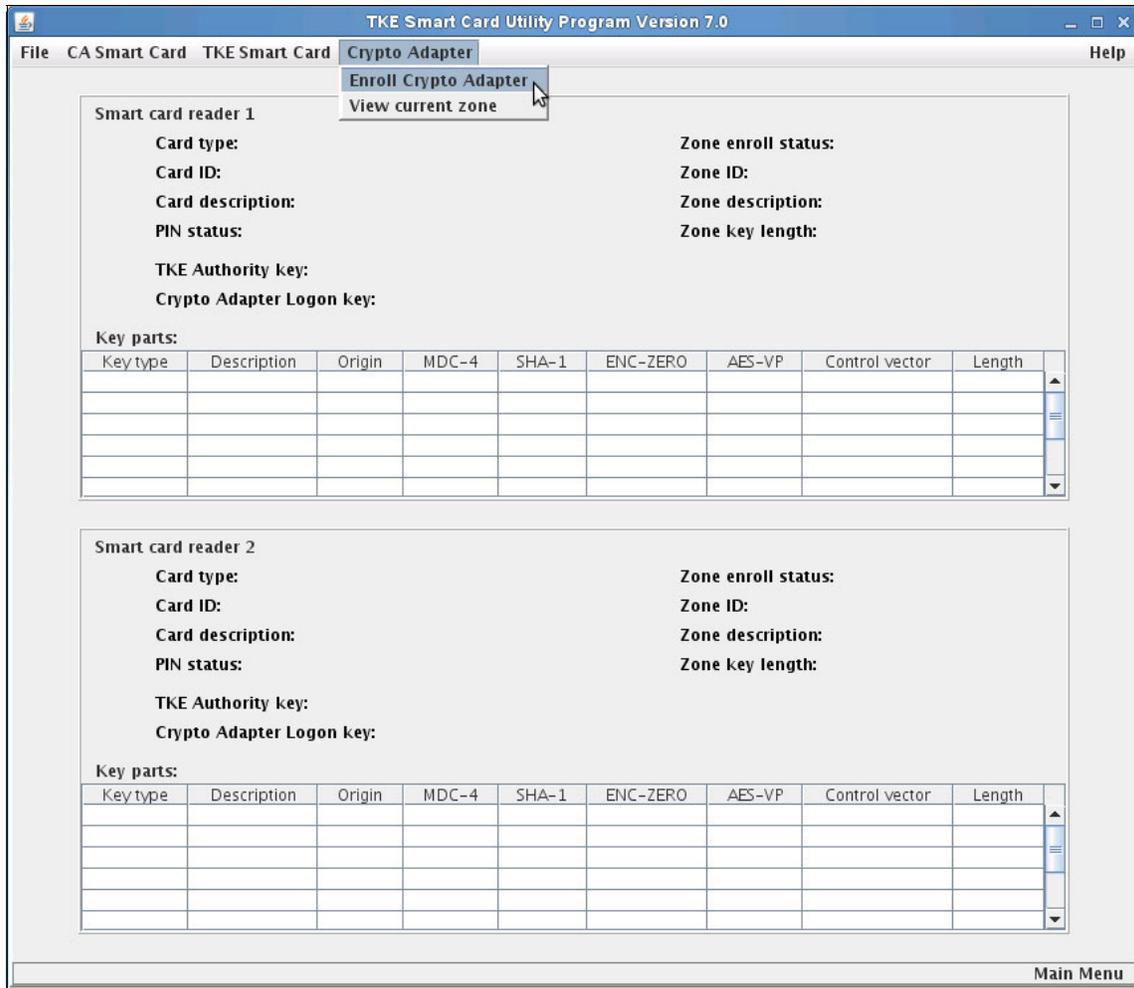


Figure B-20 Select Enroll Crypto Adapter

4. You are prompted to confirm the enrollment of the local IBM PCIe 4765 Cryptographic Coprocessor. Select **OK**, as shown in Figure B-21.



Figure B-21 Confirmation

5. ADM1n is prompted to insert the CA smart card in reader 1.
6. A prompt indicates the validation of smart card communication.

After a while, ADM1n is prompted to enter the 6-digit PIN-1 of the CA smart card on reader 1, as shown in Figure B-22.

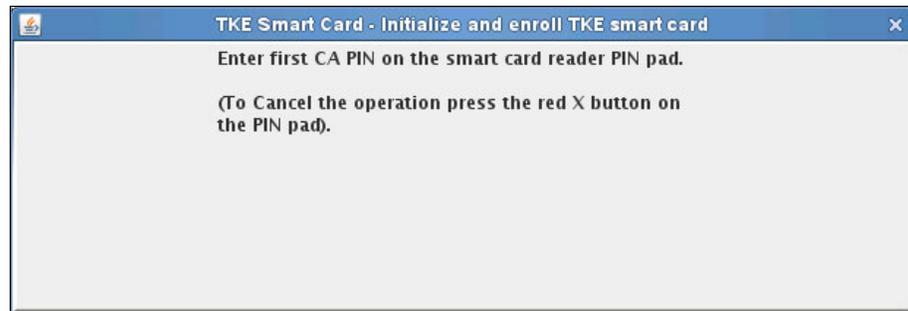


Figure B-22 Enter the first CA PIN - ADM1n

7. ADM2n is prompted to enter the 6-digit PIN-2 of the CA smart card on reader 1, as shown in Figure B-23 on page 267.

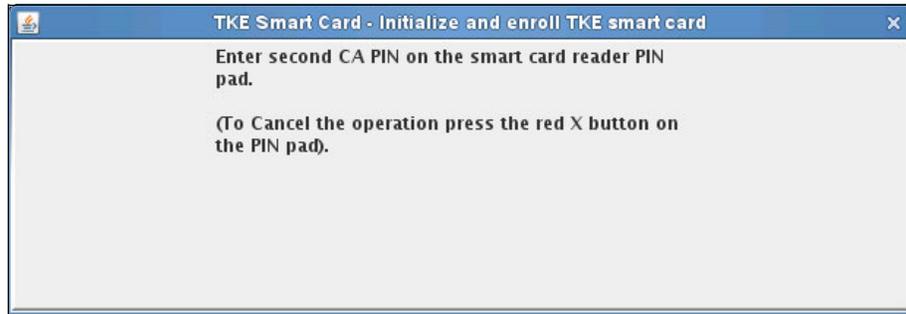


Figure B-23 Enter the second CA PIN - ADM2n

8. A prompt indicates that the certification of the enrollment request is underway. After a short while, another prompt opens and indicates the successful enrollment of the IBM PCIe 4765 Cryptographic Coprocessor, as shown in Figure B-24. Select **OK**.



Figure B-24 Success

## Initializing and enrolling a TKE smart card

The aim of this procedure is to initialize a TKE smart card and enroll it into the zone that is set by the CA smart card using the SCUP.

The TKE smart card must be personalized afterward (“Personalizing a TKE smart card” on page 272) before it can be used for one of the following purposes:

- ▶ Secure storage for new master key parts
- ▶ Crypto adapter logon
- ▶ Backup of an existing TKE smart card

**PIN re-entry not required:** If you have entered the two 6-digit PINs for the CA smart card, have not restarted SCUP, and have not removed the CA smart card, the two PINs (of the CA smart card) might not require re-entry when you are initializing TKE smart cards. This feature is used only when initializing TKE smart cards. All other functions that require the CA PINs require re-entry every time.

## Participants

Table B-5 lists the participants with their roles and a brief description.

*Table B-5 Participants*

| Role indication | Role description                                         |
|-----------------|----------------------------------------------------------|
| ADM1n           | Holder of CA smart card, and PIN-1 for the CA smart card |
| ADM2n           | Holder of PIN-2 for the CA smart card                    |
| ADM1n / ADM2n   | Responsible for registering the TKE smart card           |
| CARD HOLDER     | Holder of the TKE smart card to be personalized          |

## Special requirements

The participants that are listed in Table B-5 need the following components.

- ▶ ADM1n must have the following components:
  - An envelope containing the CA smart card
  - An envelope containing the PIN-1 PIN Form for the CA smart card
- ▶ ADM2n must have an envelope containing the PIN-2 PIN Form for the CA smart card.
- ▶ CARD HOLDER must have the following components:
  - A blank smart card (or an old one that is no longer needed)
  - A label for the blank smart card
  - A pre-labelled envelope for the storage of the initialized and enrolled TKE smart card

ADM1n and ADM2n are responsible for registering the TKE smart card in the inventory of smart cards. This registration should include the following items:

- ▶ The creation date of the TKE smart card.
- ▶ The zone ID/description (inherited from the CA smart card).
- ▶ The Card ID.
- ▶ The CARD HOLDER identification/name.
- ▶ The storage location.

- ▶ Intended use (master key parts and crypto adapter logon). If it is a backup of another TKE smart card, then it also needs the following components:
  - Original zone ID/description
  - Original card ID *and description*
  - Original CARD HOLDER identification/name
  - Original storage location

CARD HOLDER is responsible for the custody of the new initialized and enrolled TKE smart card.

**Additional credentials that are needed:** When the SCUP is started, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

### **Procedure: Initializing and enrolling a TKE smart card**

Complete the following steps:

1. Click **Computer** → **Applications** menu and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290.
  - “SCUP group logon by using smart cards” on page 292.

- Click **TKE Smart Card** → **Initialize and enroll TKE smart card**, as shown in Figure B-25.

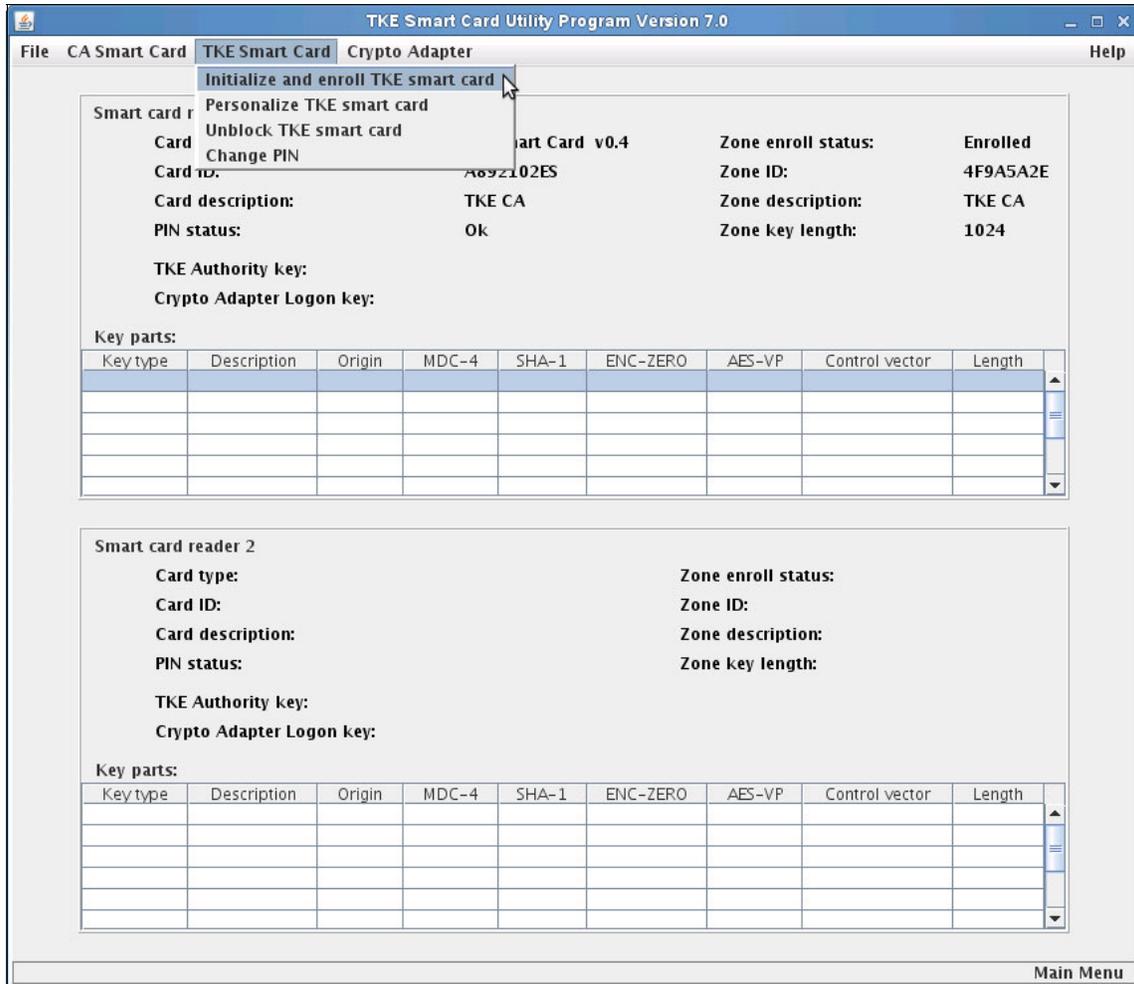


Figure B-25 Select Initialize and enroll the TKE smart card

- ADM1n is prompted to insert the CA smart card in to reader 1.
- ADM1n is prompted to enter the 6-digit PIN-1 of the CA smart card.
- ADM2n is prompted to enter the 6-digit PIN-2 of the CA smart card.
- CARD HOLDER is prompted to insert the target blank TKE smart card in to reader 2. Select **OK**, as shown in Figure B-26 on page 271.

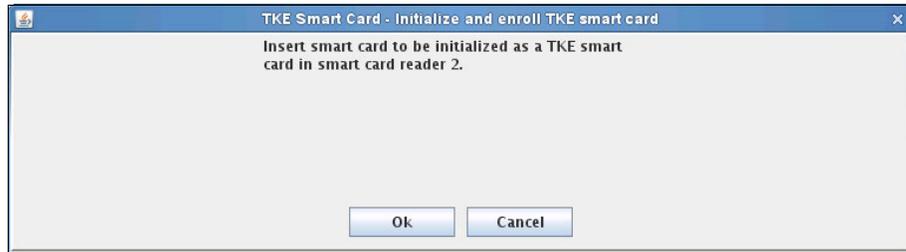


Figure B-26 Insert a blank smart card

8. A prompt opens and indicates that the initialization process is underway. After 10 - 30 seconds, a second prompt opens and indicates that the TKE smart card build process is underway, as shown in Figure B-27.

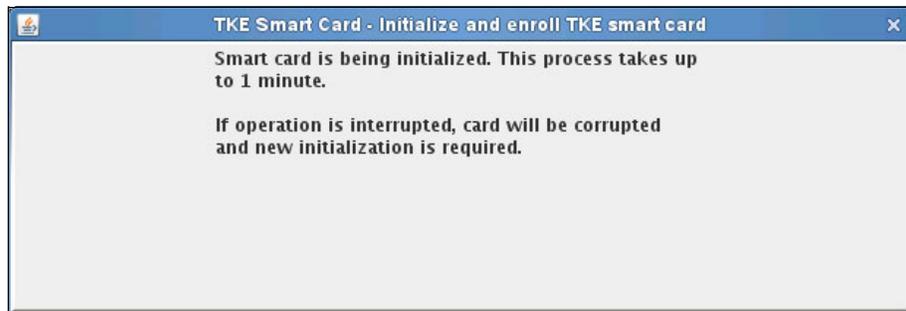


Figure B-27 Process underway

9. A prompt opens and confirms the success of the initialization and enrollment of the new TKE smart card. Select **OK**, as shown in Figure B-28.



Figure B-28 Success

10. After the process is complete, a message prompt opens showing the retrieval of the smart card information. After a few seconds, the main window opens with the information from the CA smart card that is inserted in reader 1 and the initialized and enrolled TKE smart card in reader 2, as shown in Figure B-29.

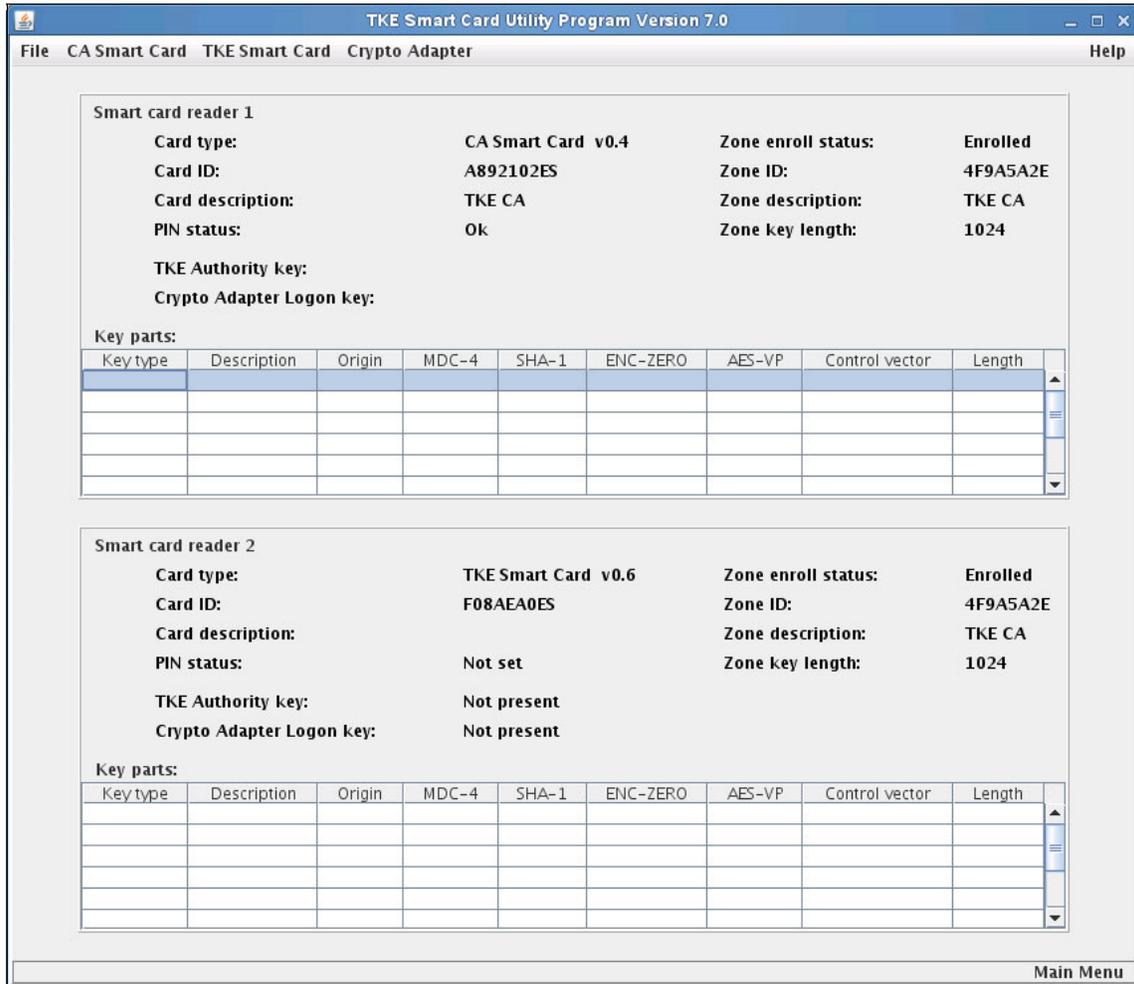


Figure B-29 Procedure complete

## Personalizing a TKE smart card

The aim of this procedure is to set the 6-digit PIN of an initialized and enrolled TKE smart card by using the SCUP.

After personalization, the intended use of the TKE smart card can be one of the following items:

- ▶ Secure storage for new master key parts
- ▶ Crypto adapter logon
- ▶ Backup of an existing TKE smart card

If the intended use is secure storage for (new) master key parts, then these parts must be created afterward (for more information, see “Procedure: Generating an IBM 4765 DES/PKA master key” on page 318).

If the intended use is crypto adapter logon, a logon key must be generated afterward (for more information, see “Generating an IBM 4765 logon key on TKE smart card” on page 306).

If the intended use is the creation of a backup of an existing TKE smart card, then continue with the procedure that is described in “Backing up a TKE smart card” on page 312.

## Participants

Table B-6 lists the participants with their roles and a brief description.

*Table B-6 Participants*

| <b>Role indication</b> | <b>Role description</b>                         |
|------------------------|-------------------------------------------------|
| ADM1n / ADM2n          | Responsible for registering the TKE smart card  |
| CARD HOLDER            | Holder of the TKE smart card to be personalized |

## Special requirements

The participants that are listed in Table B-6 need the following components:

- ▶ CARD HOLDER must have the following components:
  - An envelope with the initialized and enrolled smart card
  - A PIN Form for the smart card
  - A pre-labelled envelope for the PIN Form
- ▶ ADM1n and ADM2n are responsible for registering the TKE smart card in the inventory of smart cards. This registration should include the following components:
  - A zone ID/description (inherited from the CA smart card)
  - A card ID *and description*
  - The CARD HOLDER identification/name

- The storage location
- The intended use (master key parts and crypto adapter logon)
- If there is a backup of another TKE smart card, then the following components also are required:
  - Original zone ID/description
  - Original card ID and description
  - Original CARD HOLDER identification/name
  - Original storage location

CARD HOLDER is responsible for the custody of the TKE smart card.

**Additional credentials:** When the SCUP is started, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

### **Procedure: Personalizing a TKE smart card**

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290.
  - “SCUP group logon by using smart cards” on page 292.

- Click **TKE Smart Card** → **Personalize TKE smart card**, as shown in Figure B-30.

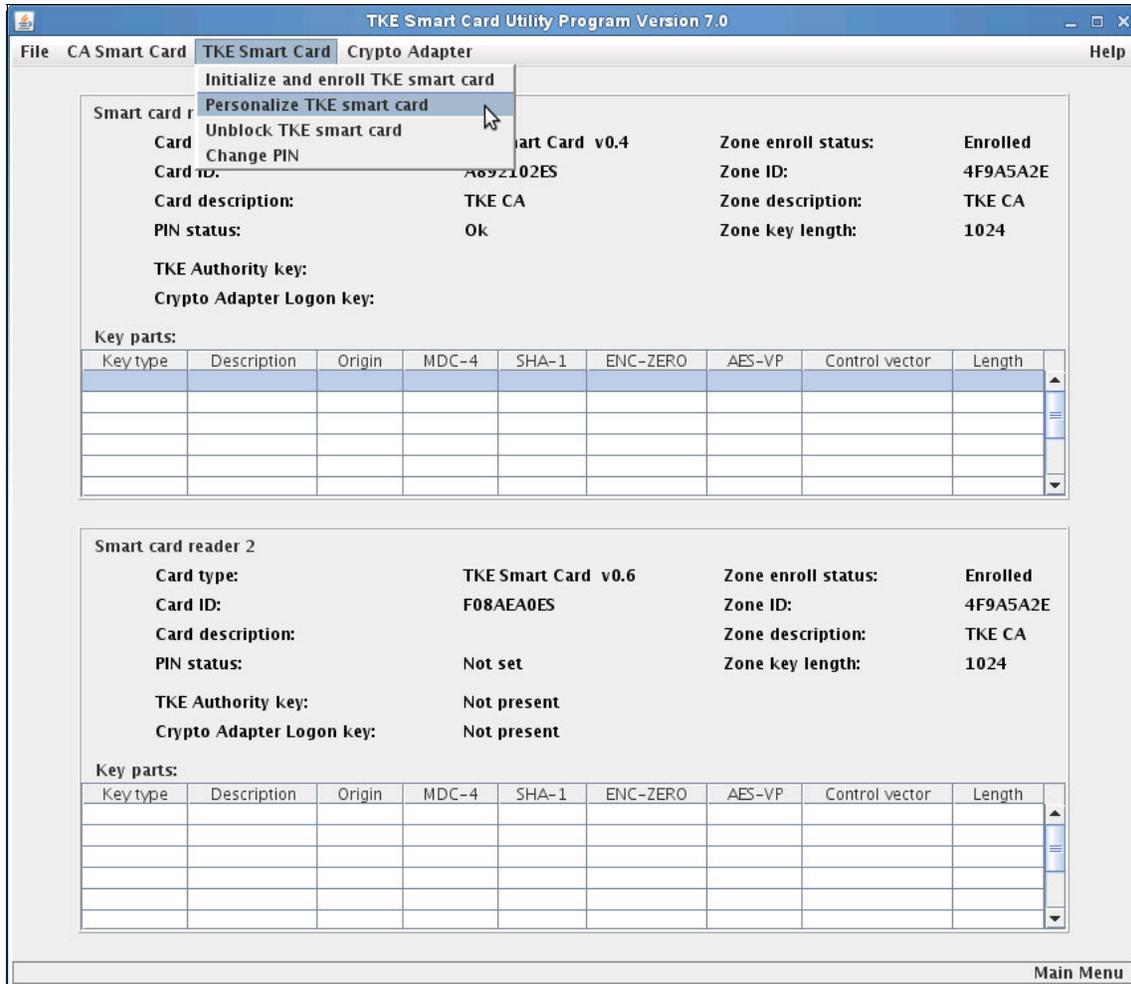


Figure B-30 Select Personalize TKE smart card

- CARD HOLDER is prompted to insert the target TKE smart card into reader 2. Select **OK**, as shown in Figure B-31.

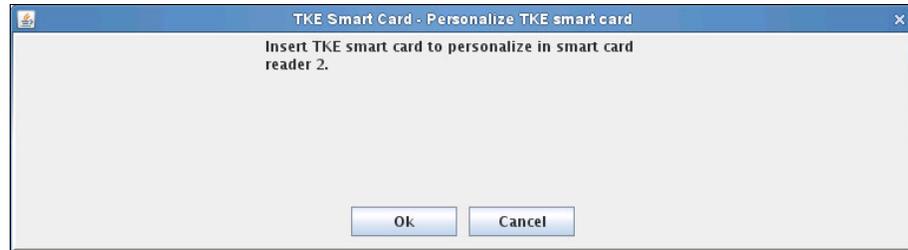


Figure B-31 Insert a smart card

- CARD HOLDER is prompted to enter the new 6-digit PIN *twice* on reader 2, as shown in Figure B-32.

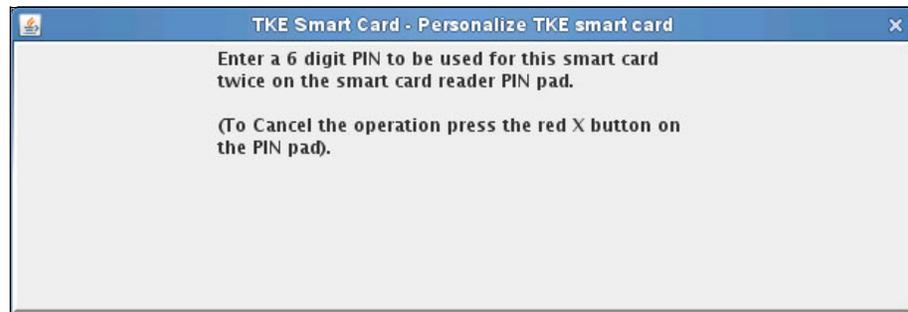


Figure B-32 Enter the PIN twice

- CARD HOLDER is prompted to enter the (optional) description for the TKE smart card. Click **OK**, as shown in Figure B-33.

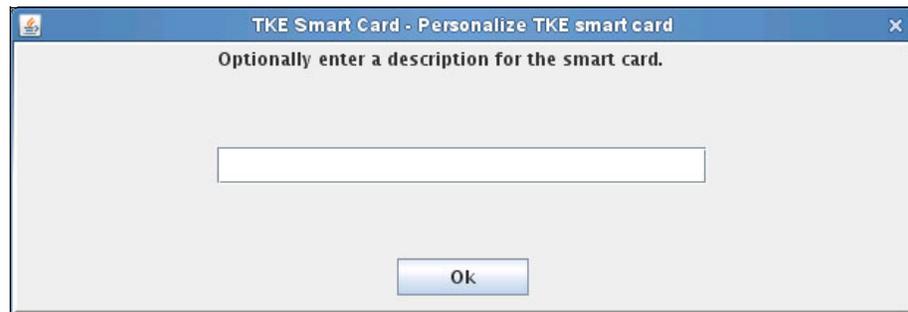
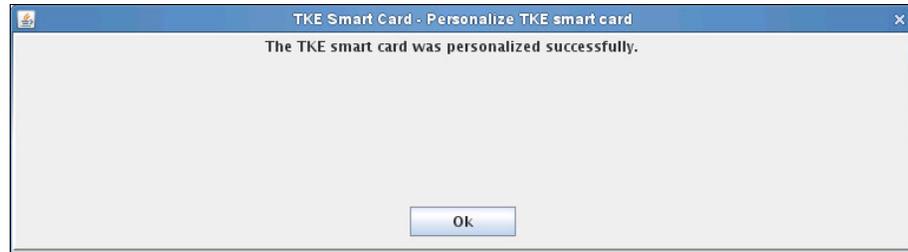


Figure B-33 Enter a description

7. A prompt confirms the success of the personalization of the TKE smart card. Select **OK**, as shown in Figure B-34.



*Figure B-34 Success*

8. After the process is complete, a message prompt opens showing the retrieval of the smart card information. After a few seconds, the main window opens with the information from the personalized TKE smart card in reader 2, as shown in Figure B-35.

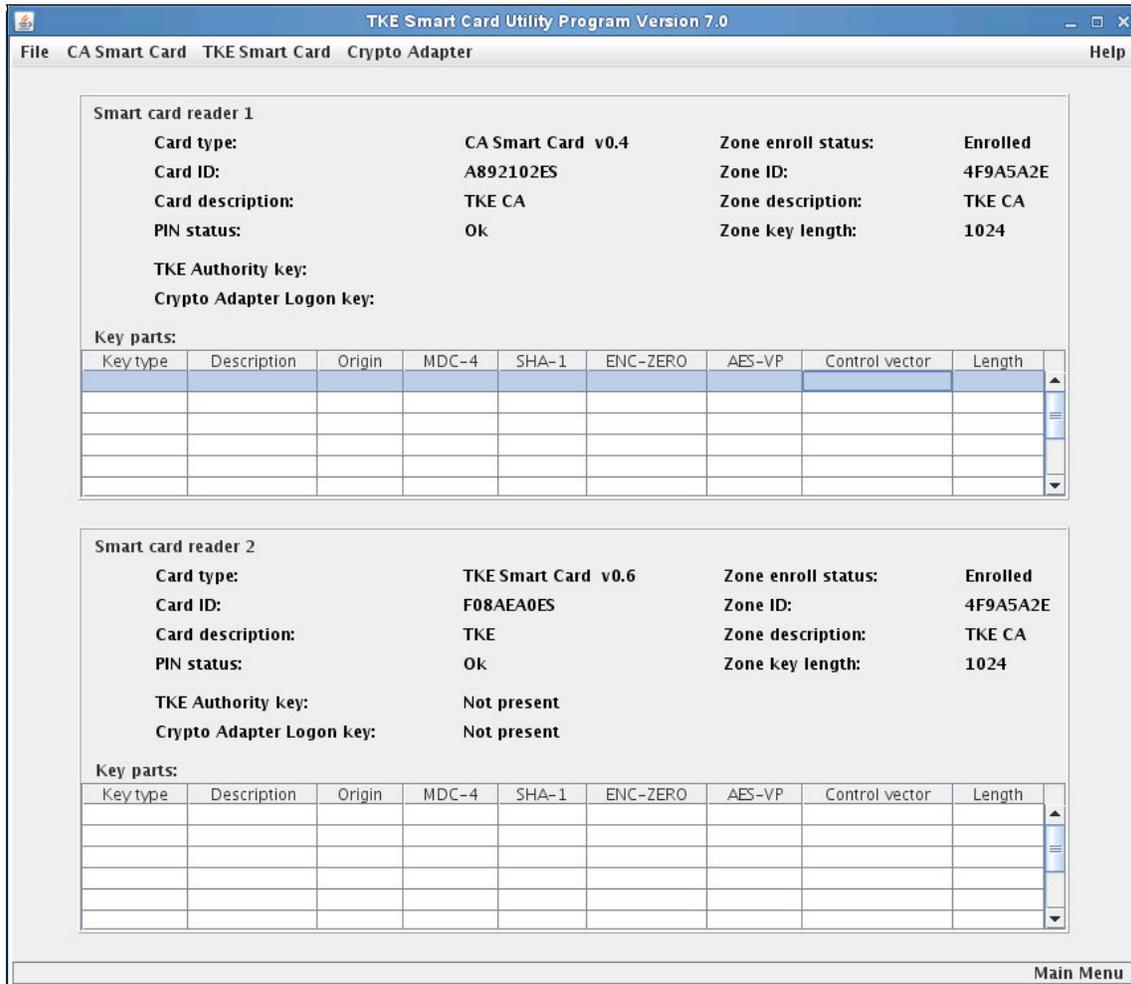


Figure B-35 Procedure complete

## Unblocking a TKE smart card

If a TKE smart card is blocked because of too many incorrect PIN entries, it can be unblocked by using the CA smart card that was used to initialize and enroll the TKE smart card by using the SCUP, as shown in Figure B-36 on page 279.

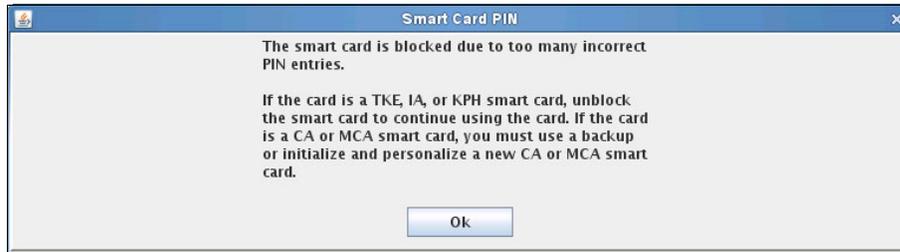


Figure B-36 Smart card blocked

Content that is stored on a blocked TKE smart card, such as IBM 4765 master key parts, or an IBM 4765 logon key pair, can be recovered only after a successful unblock *and* with knowledge of the correct PIN. You cannot set a new PIN during this process.

**PIN retries for a CA smart card:** It is not possible to unblock a CA smart card. For that reason, you have 10 attempts on both PINs for a CA smart card (compared to only three attempts for a TKE smart card).

## Participants

Table B-7 lists the participants with their roles and a brief description.

Table B-7 Participants

| Role indication | Role description                                           |
|-----------------|------------------------------------------------------------|
| ADM1n           | Holder of a CA smart card, and PIN-1 for the CA smart card |
| ADM2n           | Holder of PIN-2 for the CA smart card                      |
| CARD HOLDER     | Holder of the TKE smart card to be unblocked               |

## Special requirements

Participants that are listed in Table B-7 need the following components:

- ▶ ADM1n must have the following components:
  - An envelope containing the CA smart card
  - An envelope containing the PIN-1 PIN Form for the CA smart card
- ▶ ADM2n must have an envelope containing the PIN-2 PIN Form for the CA smart card.
- ▶ CARD HOLDER must have the following components:
  - An envelope with blocked TKE smart card
  - An envelope with a PIN Form for the blocked TKE smart card

**Additional credentials:** When the SCUP is started, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

### **Procedure: Unblocking a TKE smart card**

Complete the following steps:

1. Click **Computer** → **Applications and** locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290
  - “SCUP group logon by using smart cards” on page 292
3. Click **TKE Smart Card** → **Unblock TKE smart card**, as shown in Figure B-37 on page 281.

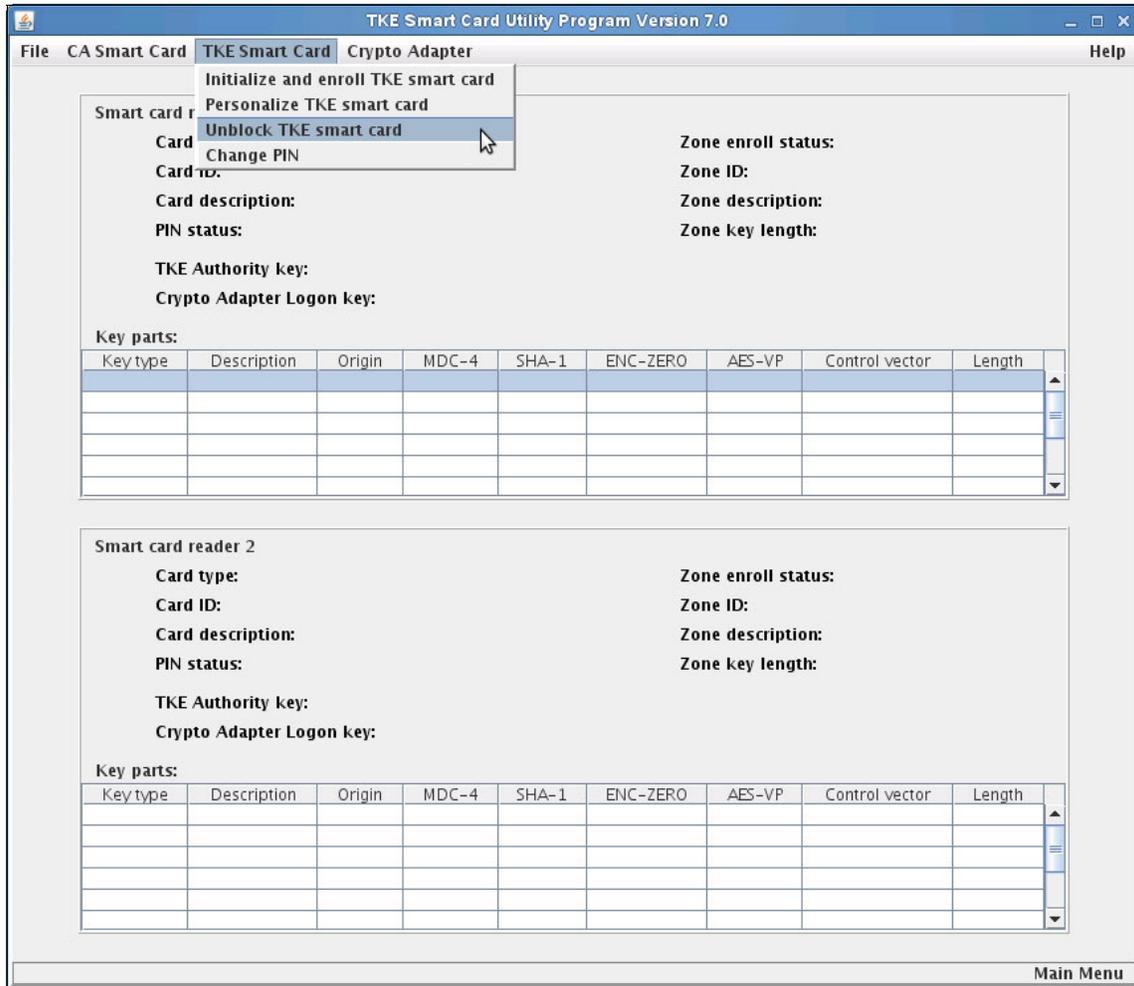
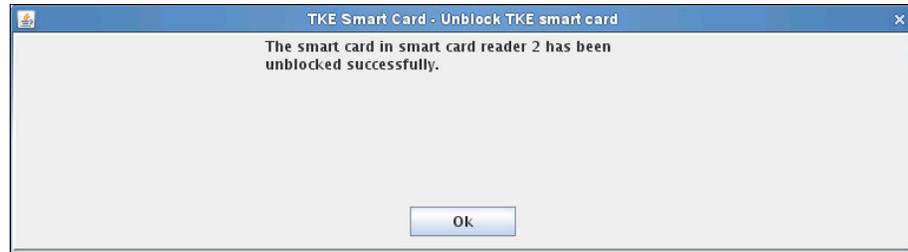


Figure B-37 Select Unblock TKE smart card

4. ADM1n is prompted to insert the CA smart card in to reader 1.
5. ADM1n is prompted to enter the 6-digit PIN-1 of the CA smart card in reader 1.
6. ADM2n is prompted to enter the 6-digit PIN-2 of the CA smart card in reader 1.
7. CARD HOLDER is prompted to insert the TKE smart card to be unblocked in to reader 2. Select **OK** to continue.
8. A prompt opens and indicates that the unblocking process is underway.

After a short while, a second prompt opens and confirms the success of the unblocking. Select **OK**, as shown in Figure B-38.



*Figure B-38 Smart card unblocked*

- After the process is complete, a message prompt opens showing the retrieval of the smart card information. After a few seconds the main window opens with the information from the CA smart card that is inserted in reader 1 and the unblocked TKE smart card that is inserted in reader 2, as shown in Figure B-39.

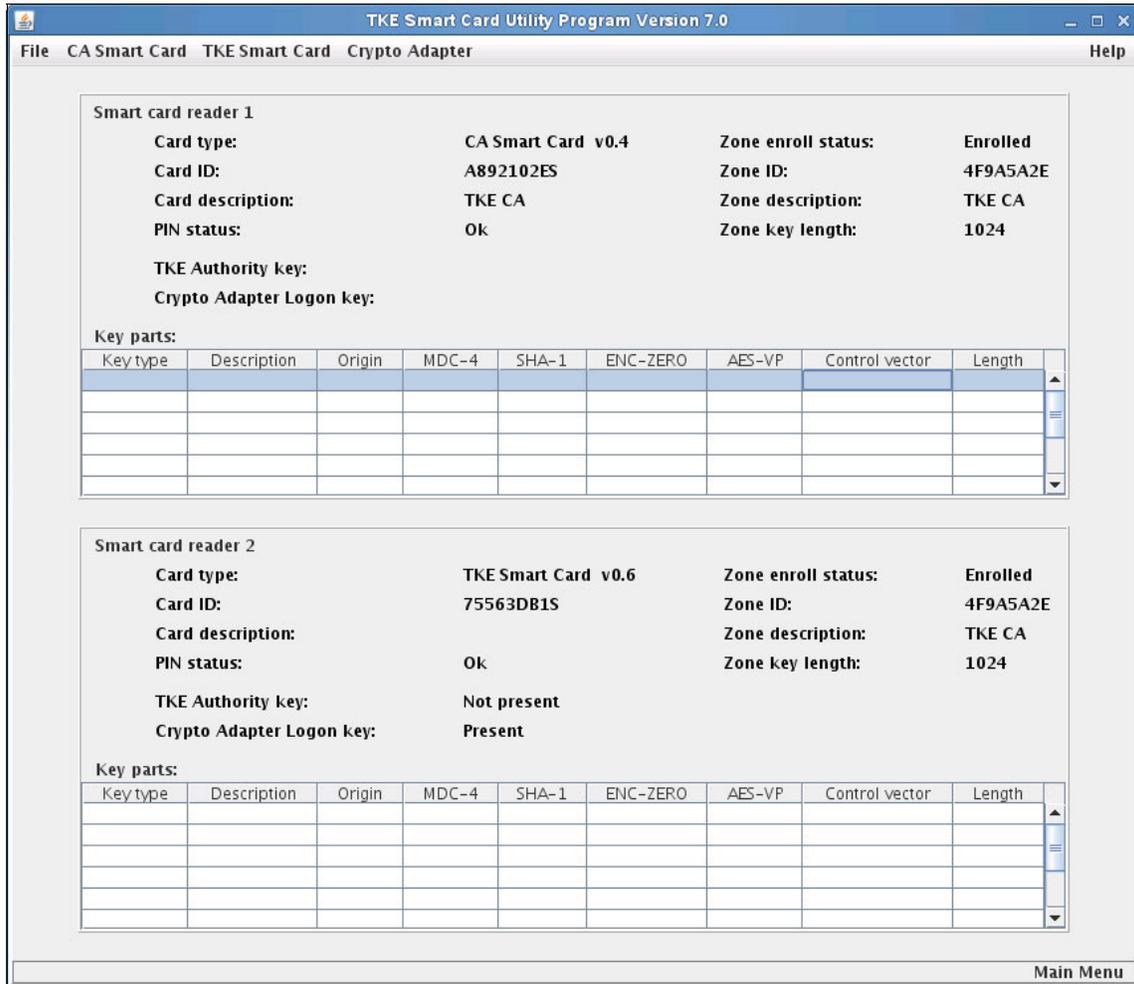


Figure B-39 Procedure complete

## Changing the PIN of a CA smart card

The aim of this procedure is to change the first PIN-1 or the second PIN-2 of a CA smart card by using the SCUP.

### Participants

Table B-8 lists the participants with their roles and a brief description.

*Table B-8 Participants*

| Role indication | Role description                                           |
|-----------------|------------------------------------------------------------|
| ADM1n           | Holder of CA smart cards, and PIN-1 for the CA smart cards |
| ADM2n           | Holder of PIN-2 for the CA smart cards                     |

### Special requirements

Participants that are listed in Table B-8 need the following components:

- ▶ ADM1n must have the following components:
  - An envelope containing the CA smart card
  - An envelope containing the old PIN-1 PIN Form for the CA smart card
  - A new PIN-1 PIN Form for the CA smart card (if PIN-1 will be changed)
- ▶ ADM2n must have the following components:
  - An envelope containing the PIN-2 PIN Form for the CA smart card
  - A new PIN-2 PIN Form for the CA smart card (if PIN-2 will be changed)

**Additional credentials:** When the SCUP is started, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

### Procedure: Changing the PIN of a CA smart card

Complete the following steps:

1. Click **Computer** → **Applications** menu and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290
  - “SCUP group logon by using smart cards” on page 292
3. Click **CA Smart Card** → **Change PIN**, as shown in Figure B-40 on page 285.

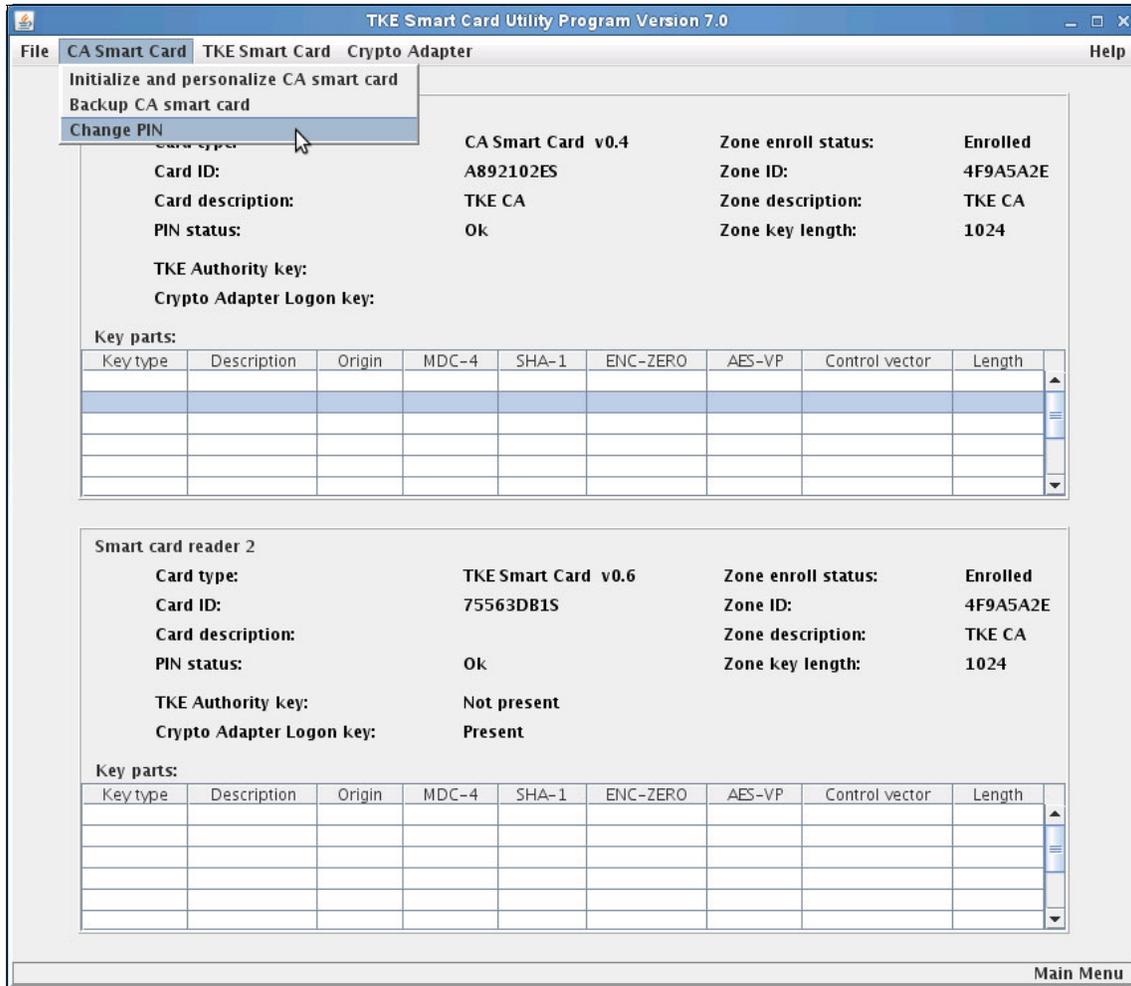


Figure B-40 Select Change PIN

4. ADM1n is prompted to insert the CA smart card in to reader 1. Select **OK** to continue.
5. Select the PIN to change. Select one of the following options:
  - **First CA PIN**
  - **Second CA PIN**

Select **OK**, as shown in Figure B-41.

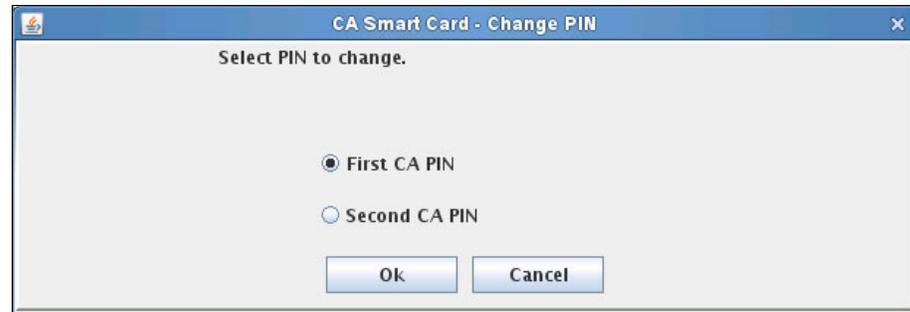


Figure B-41 Select PIN to change

6. Depending on the selection that you made in step 5 on page 285, either ADM1n or ADM2n is prompted to enter the *current* 6-digit PIN-1 (First CA PIN) or PIN-2 (Second CA PIN) of the CA smart card in reader 1, as shown in Figure B-42.

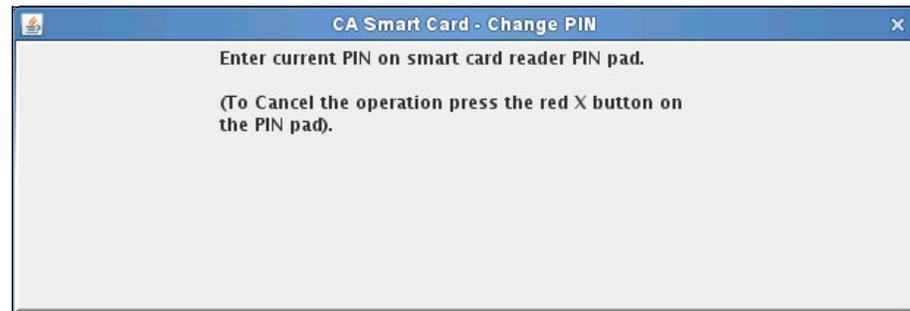


Figure B-42 Enter current PIN

7. Depending on the selection that you made in step 5 on page 285, either ADM1n or ADM2n is prompted to enter the *new* 6-digit PIN-1 or PIN-2 of the CA smart card in reader 1, as shown in Figure B-43 on page 287.

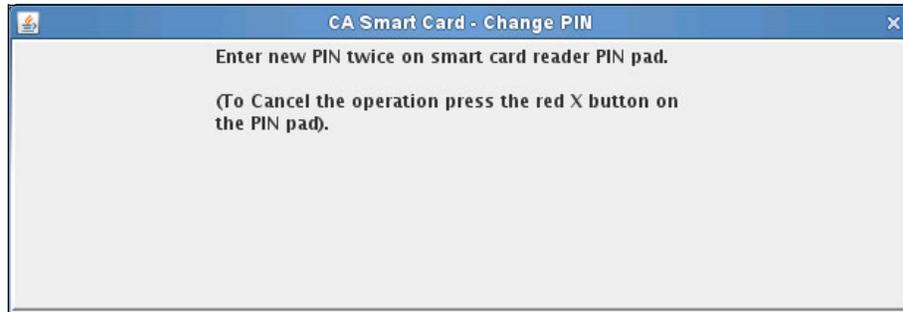


Figure B-43 Enter new PIN twice

8. A prompt opens and indicates that the PIN that was selected in step 5 on page 285 was changed successfully, as shown in Figure B-44. Click **OK**.



Figure B-44 Procedure complete

## Changing the PIN of a TKE smart card

The aim of this procedure is to change the PIN of a personalized TKE smart card by using the SCUP.

### Participant

Table B-9 lists the participant with its roles and a brief description.

Table B-9 Participant

| Role indication | Role description         |
|-----------------|--------------------------|
| CARD HOLDER     | Holder of TKE smart card |

## Special requirements

The CARD HOLDER participant that is listed in Table B-9 on page 287 needs the following components:

- ▶ An envelope containing the TKE smart card
- ▶ An envelope containing the old PIN Form for the TKE smart card
- ▶ A new PIN Form for the TKE smart card

**Additional credentials:** When the SCUP starts, you must log on to the IBM PCIe 4765 Cryptographic Coprocessor. Participants need additional credentials to perform this logon.

## Procedure: Changing the PIN of the TKE smart card

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. You are prompted to perform a SCUP logon to the IBM PCIe 4765 Cryptographic Coprocessor. You can perform this task by using either of the following procedures:
  - “SCUP logon by using split passphrase” on page 290
  - “SCUP group logon by using smart cards” on page 292
3. Click **TKE Smart Card** → **Change PIN** as shown in Figure B-45 on page 289.

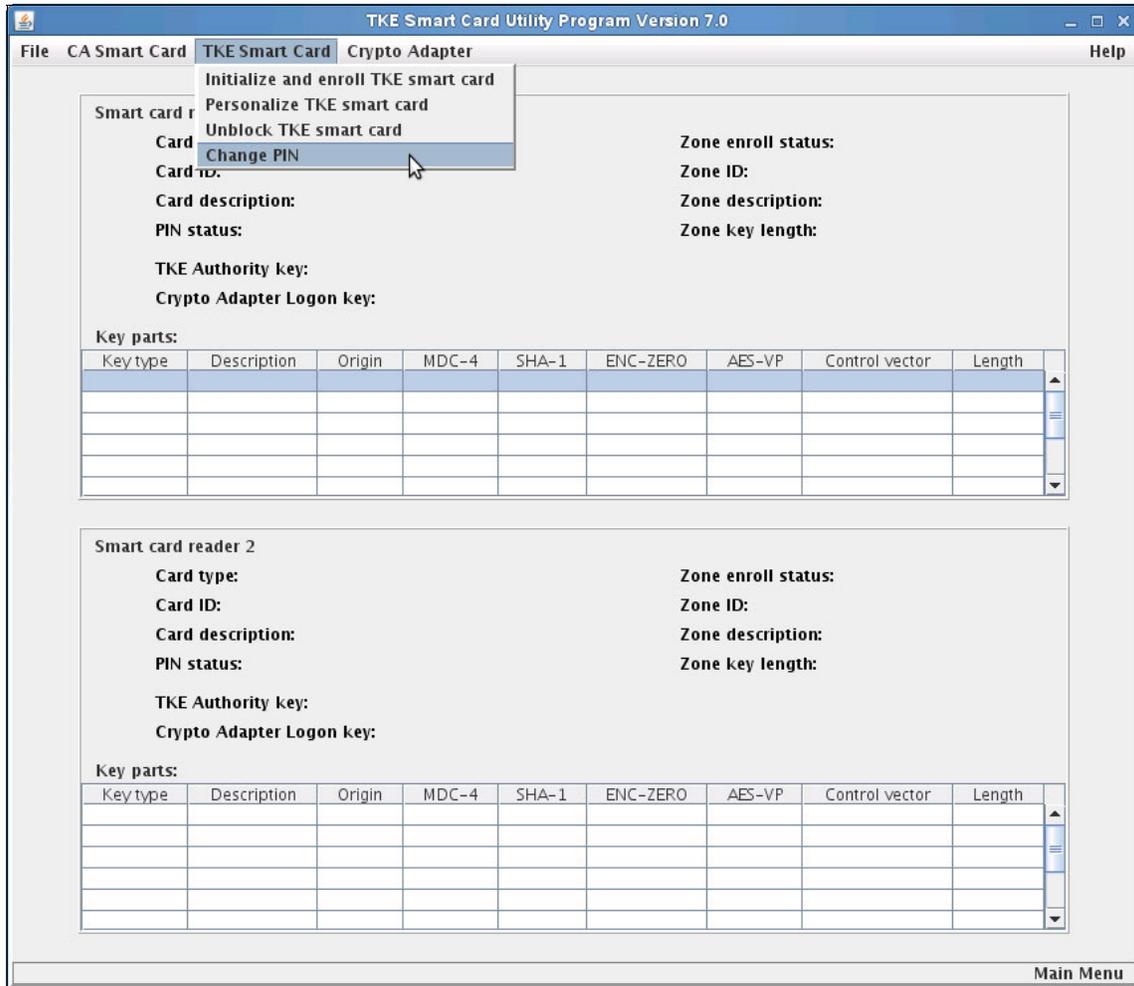


Figure B-45 Select Change PIN

4. CARD HOLDER is prompted to insert the TKE smart card in reader 2. Click **OK** to continue.
5. CARD HOLDER is prompted to enter the *current* 6-digit PIN of the TKE smart card on reader 2.
6. CARD HOLDER is prompted to enter the *new* 6-digit PIN of the TKE smart card, twice, on reader 2.
7. A prompt indicates that the PIN was changed successfully.  
Select **OK** to complete the procedure.

## SCUP logon by using split passphrase

The aim of this procedure is to log on to the IBM PCIe 4765 Cryptographic Coprocessor from the SCUP application by using a split knowledge passphrase.

In the initial phase of the secure setup of the EKMF workstation, the temporary CNMADMIN passphrase profile is created in the IBM PCIe 4765 Cryptographic Coprocessor. The profile passphrase is split so that one person knows the first half and another person knows the last half. The intended usage is to enforce dual-controlled access to necessary commands (access-control points) during the secure setup of the IBM PCIe 4765 Cryptographic Coprocessor.

### Participants

Table B-10 lists the participants with their roles and a brief description.

*Table B-10 Participants*

| Role indication | Role description                                                   |
|-----------------|--------------------------------------------------------------------|
| ADM1n           | Holder of the first half of the passphrase of the CNMADMIN profile |
| ADM2n           | Holder of the last half of the passphrase of the CNMADMIN profile  |

### Special requirements

The participants that are listed in Table B-10 need the following components:

- ▶ ADM1n must have an envelope with first half of the passphrase for the CNMADMIN profile.
- ▶ ADM2n must have an envelope with last half of the passphrase for the CNMADMIN profile.

### Procedure: SCUP logon by using a split passphrase

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. After you start the TKE SCUP (IBM 4765 SCUP), you automatically are prompted to log on to the IBM PCIe 4765 Cryptographic Coprocessor:
  - a. Select the **CNMADMIN** profile.
  - b. Select **OK**, as shown in Figure B-46 on page 291.



Figure B-46 Select the CNADMIN profile

3. ADM1n is prompted to enter *first half* of the split passphrase, as shown in Figure B-47. Click **OK**.



Figure B-47 First half of the passphrase

4. ADM2n is prompted to enter *last half* of the split passphrase, as shown in Figure B-48. Click **OK**.



Figure B-48 Second half of the passphrase

- The TKE SCUP (IBM 4765 SCUP) is now in an operative state and presents the main window, as shown in Figure B-49.

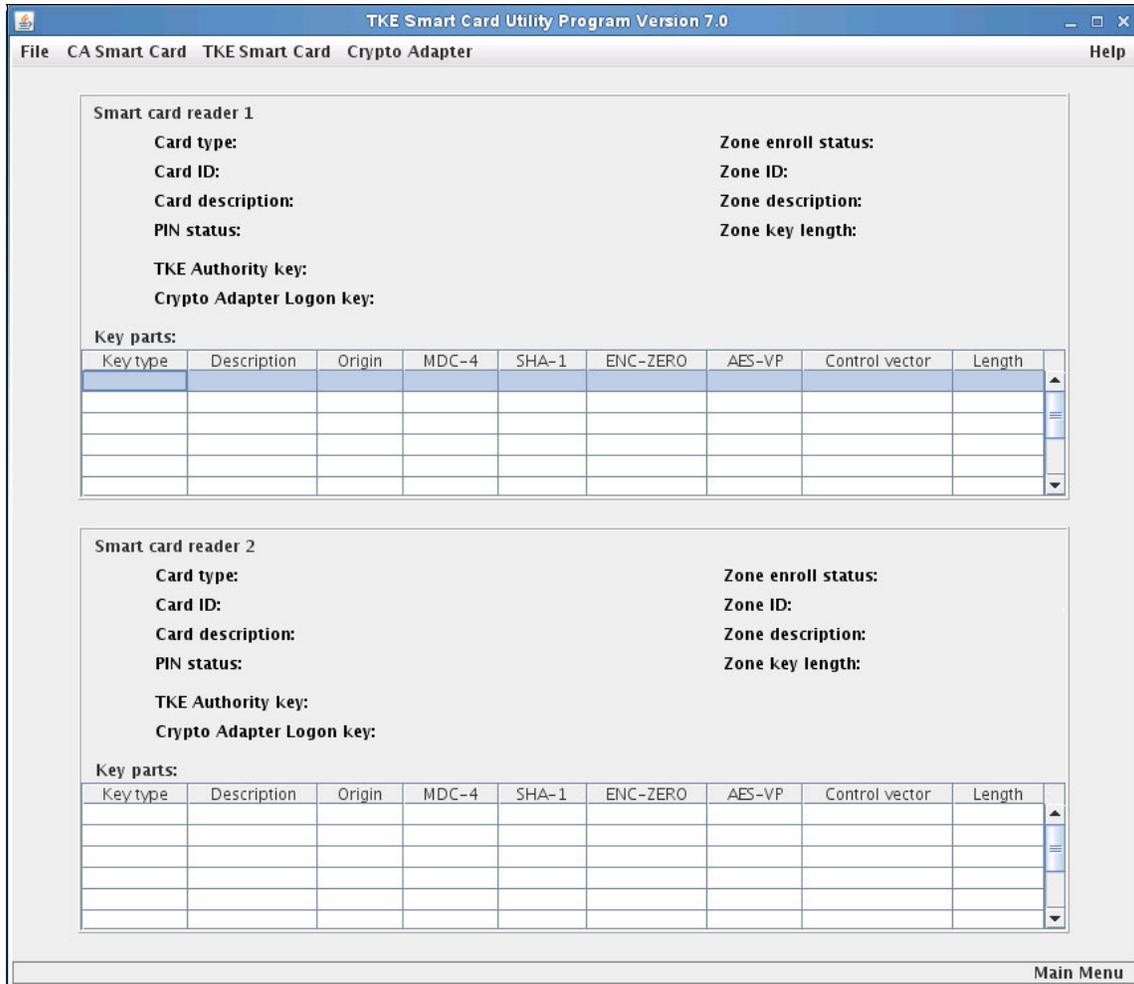


Figure B-49 Procedure complete

## SCUP group logon by using smart cards

The aim of this procedure is to perform a group logon to the IBM PCIe 4765 Cryptographic Coprocessor from the SCUP application by using logon smart cards.

## Participants

Table B-11 lists the participants with their roles and a brief description.

Table B-11 Participants

| Role indication | Role description                                              |
|-----------------|---------------------------------------------------------------|
| ADM1n           | The holder of a logon smart card belonging to the ADM1 group. |
| ADM2n           | The holder of a logon smart card belonging to the ADM2 group. |

## Special requirements

The participants that are shown in Table B-11 need the following components:

- ▶ ADM1n must have the following components:
  - An envelope with the ADM1n logon smart card
  - An envelope with a PIN Form for the ADM1n logon smart card
- ▶ ADM2n must have the following components:
  - An envelope with the ADM2n logon smart card
  - An envelope with a PIN Form for the ADM2n logon smart card

## Procedure: SCUP group logon by using smart cards

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the TKE SCUP (**IBM 4765 SCUP**).
2. After you start the TKE SCUP (IBM 4765 SCUP), you automatically are prompted to log on to the IBM PCIe 4765 Cryptographic Coprocessor:
  - a. Select the **ADMIN** profile.
  - b. Select **OK**, as shown in Figure B-50.



Figure B-50 Crypto Adapter Logon

3. ADM1n is prompted to select the ADM1n logon smart card, as shown in Figure B-51. Click **OK**.



Figure B-51 Crypto Adapter Group Logon

4. ADM1n is prompted to insert the ADM1n logon smart card in to reader 1, as shown in Figure B-52. Click **OK**.



Figure B-52 Smart card logon - insert smart card

5. ADM1n is prompted to enter the 6-digit PIN of the ADM1n logon smart card in reader 1, as shown in Figure B-53 on page 295.

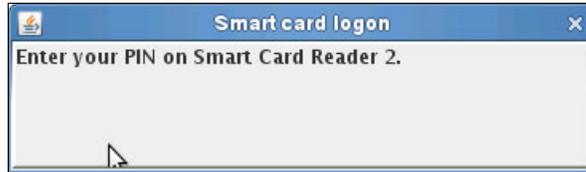


Figure B-53 Smart card logon - enter PIN

6. ADM2n is prompted to select the ADM2n logon smart card, as shown in Figure B-54. Click **OK**.



Figure B-54 Crypto Adapter Group Logon

7. ADM2n is prompted to insert the ADM2n logon smart card in to reader 1, as shown in Figure B-55. Click **OK**.



Figure B-55 Smart card logon - insert smart card 2

8. ADM2n is prompted to enter the 6-digit PIN of the ADM2n logon smart card, in reader 1, as shown in Figure B-56.



*Figure B-56 Smart card logon - enter PIN 2*

9. The TKE SCUP (IBM 4765 SCUP) is now in an operative state and presents the main window, as shown in Figure B-57 on page 297.

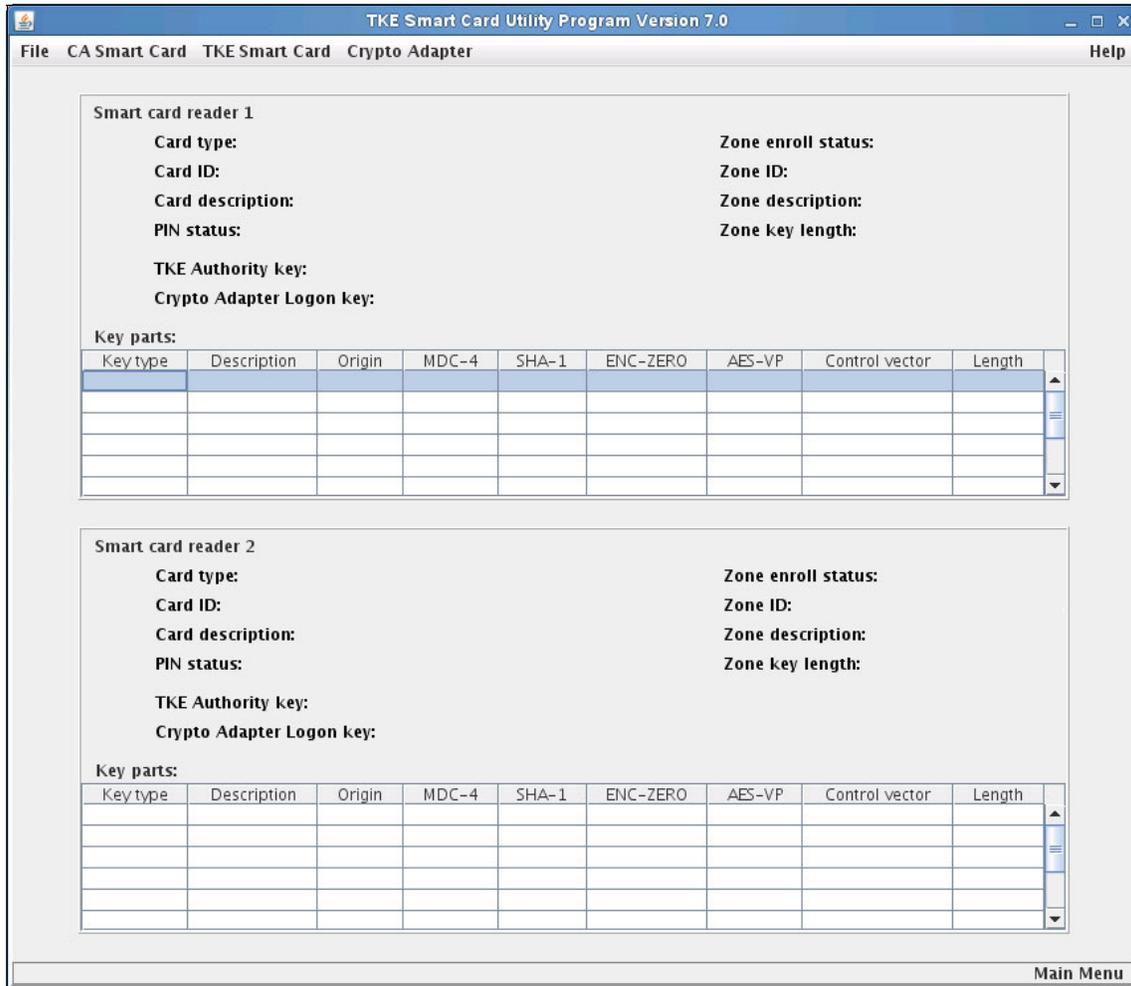


Figure B-57 Procedure complete

## IBM PCIe 4765 Cryptographic Coprocessor management using CNM

This section focuses on the following topics:

- ▶ IBM 4765 initialization
- ▶ Generating an IBM 4765 logon key on TKE smart card
- ▶ Backing up a TKE smart card
- ▶ Generating an IBM 4765 DES/PKA master key

- ▶ Generating an IBM 4765 DES/PKA master key
- ▶ Loading an IBM 4765 DES/PKA master key
- ▶ Setting the IBM 4765 DES/PKA master keys and re-enciphering the key storage
- ▶ Performing a CNM Utility logon by using a split passphrase
- ▶ Performing a CNM Utility logon by using a smart card
- ▶ Performing a CNM Utility group logon by using smart cards
- ▶ Using the CNM Utility to create, edit, or delete a role
- ▶ Using the CNM Utility to create a smart card profile
- ▶ Using the CNM Utility to create a smart card group profile
- ▶ Using the CNM Utility to create a group of groups profile
- ▶ Using the CNM Utility to restrict the DEFAULT role

## IBM 4765 initialization

The aim of this procedure is to set the IBM PCIe 4765 Cryptographic Coprocessor to a defined initial state that enables a secure configuration under dual control.

In the first steps of this procedure, the CCA test initialization utility (CCA Init) does a fast reset of the IBM 4765 that includes setting the function control vector (FCV), setting temporary random master keys (not recoverable), and initializing keystores.

The CCA Init is installed as a part of CCA 4.2. When installing CCA, click the **Custom install** option and then click **Smart Card Utility Programs**. If you already installed CCA without this option, run the installer again and click **Modify an Existing Instance**. Click **Add Features** and then click **Smart Card Utility Programs**. For instructions about installing CCA, see *4765 PCIe Cryptographic Coprocessor CCA Support Program Installation Manual*, found at:

<http://www.ibm.com/security/cryptocards/pciicc/library.shtml>

The steps that are performed by the CCA Init are a standard set of steps that you normally must perform manually to set up CCA for development and test activities. The CCA Init tool automates these steps for you. CCA Init removes all existing CCA setups and initializes CCA by performing these tasks:

- ▶ Initializes the adapter in the standard CCA manner.
- ▶ Expands the DEFAULT role to include all permissions.
- ▶ Creates a profile that is named tester, with a passphrase of tester, and attaches the DEFAULT role to that profile.
- ▶ Automatically sets master keys for data encryption standard (DES) / public key algorithm (PKA) and for advanced encryption standard (AES).

- ▶ Loads the function control vector (FCV).
- ▶ Initializes all three types of key storage: AES, DES, and PKA.

**Default role:** The DEFAULT role has all access points enabled, so there is an immediate need to define and load other predefined roles, passphrase, smart card profiles, and group profiles, and restrict the access control points of the DEFAULT role to an absolute minimum. The Key Management Workstation should *not* be left unattended until these tasks are accomplished.

In the last steps, create a CNMADMIN passphrase profile by using a two-part passphrase that is separated into a first part and a last part, which is associated with a new CNMADMIN role that has all access control points enabled. Then, remove the tester pass phrase profile and restrict the DEFAULT role to a minimum set of access control points.

After you perform this *initialize coprocessor* task, the ability to configure the access control system of the IBM 4765 requires the participation of two individuals, each holding a unique passphrase part of the CNMADMIN passphrase profile. Table B-12 lists these individual roles.

Table B-12 Participants

| Role indication | Role description           |
|-----------------|----------------------------|
| ADM1n           | A member of the ADM1 group |
| ADM2n           | A member of the ADM2 group |

## Special requirements

CCA Init must be run as root. If you run CCA Init by running **sudo**, the environment variables might not be set up correctly. CCA Init requires Java 6.

Participants need the following components:

- ▶ ADM1n must have an envelope with first part of the passphrase for the CNMADMIN passphrase profile.
- ▶ ADM2n must have an envelope with last part of the passphrase for the CNMADMIN passphrase profile.

## Procedure: IBM 4765 initialization

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start a command-line terminal window (**GNOME Terminal**), as shown in Figure B-58.

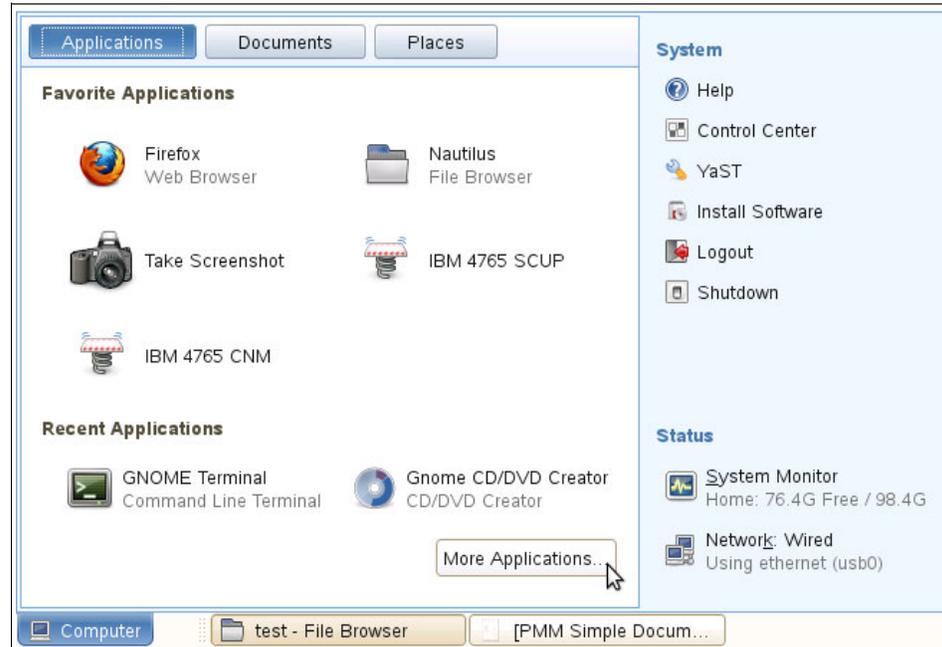


Figure B-58 Start GNOME Terminal

2. To start CCA Init, navigate to the directory containing CNM and then run the program that is shown in Example B-1.

*Example: B-1 Run CCA Init*

```
cd /opt/IBM/4765/cnm
./cca_test_init.e [> <somedirectory>/<outputfilename>]
[-adapter <adapternum>]
[-quiet]
[-overwrite_existing_data]
[-help]
```

3. A reset warning opens, as shown in Example B-2 on page 301.

*Example: B-2 CCA Init warning*

---

```
/opt/IBM/4765/cnm> ./cca_test_init.e
Warning! Proceeding with initialization will erase all existing CCA
data.
This includes all roles, profiles, retained keys, Master Keys, and
key storage files.
Do you want to continue with initialization (y/n)?
```

---

4. To proceed, enter y at the prompt and then press Enter. To see a list of the possible options, use the **-help** option.

A sample set of results is shown in Figure B-59.

```
-----CCA Initialization Started-----
Allocating Adapter: CRP01...
Adapter 1 allocated!
Working on adapter with serial number: 99002791
Adapter initialization TOKEN step complete...
Adapter Initialized using TOKEN from step 1...
Adapter Initialization Complete!
DEFAULT Role set to MAXROLE!
tester profile created with password "tester"
DES/PKA First Master Key Part Processed...
DES/PKA Last Master Key Part Processed...
DES/PKA Master Key Set!
AES First Master Key Part Processed...
AES Last Master Key Part Processed...
AES Master Key Set!
Loading FCV...
FCV Loaded!
Initializing AES, DES, & PKA Storage...
AES, DES, & PKA Storage Initialized!
Deallocating Adapter: CRP01
Adapter 1 deallocated.
-----CCA Initialization Complete-----
```

*Figure B-59 Sample result set*

5. Close the command-line (GNOME) terminal window.
6. Click **Computer** → **Applications** and locate and start the IBM 4765 CNM, as shown in Figure B-58 on page 300.

At this point, there is no need to log on to a profile to operate the CNM utility because the DEFAULT role still has all the available access control points set.

7. Create the CNMADMIN role with all the available access control points set. Click **Access Control** → **Roles** from the CNM main window (Figure B-60), and then click **New** from the Node Management window that opens.

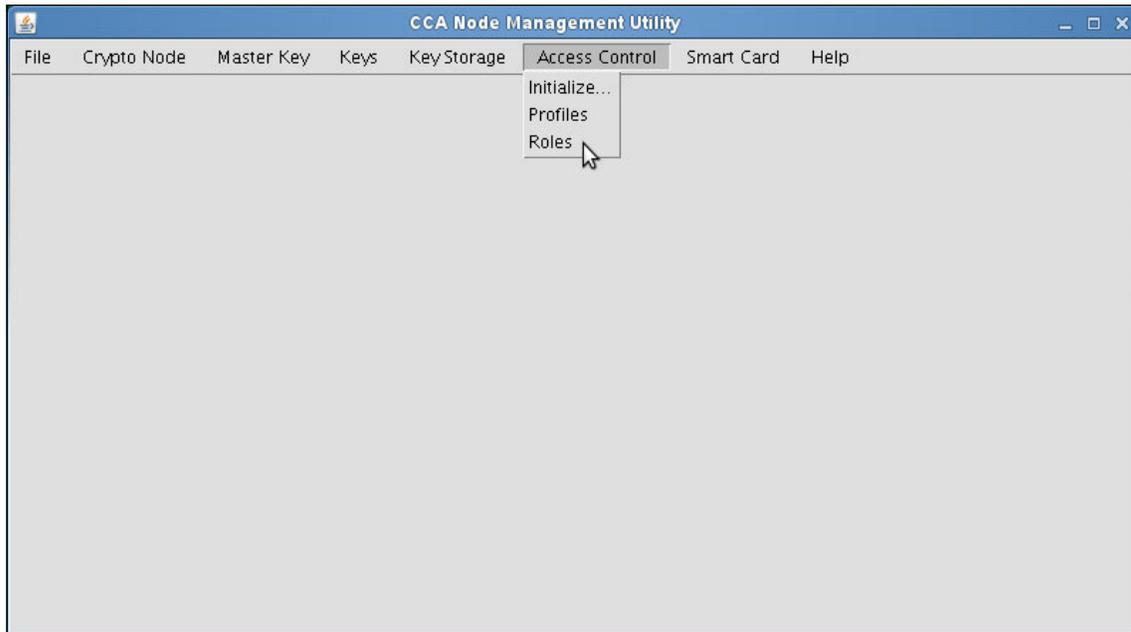


Figure B-60 CCA Node Management Utility - new roles

8. In the Role Management window, enter the Role ID (CNMADMIN), select all weekdays as valid days, click **Permit All**, and click **Load**, as shown in Figure B-61.

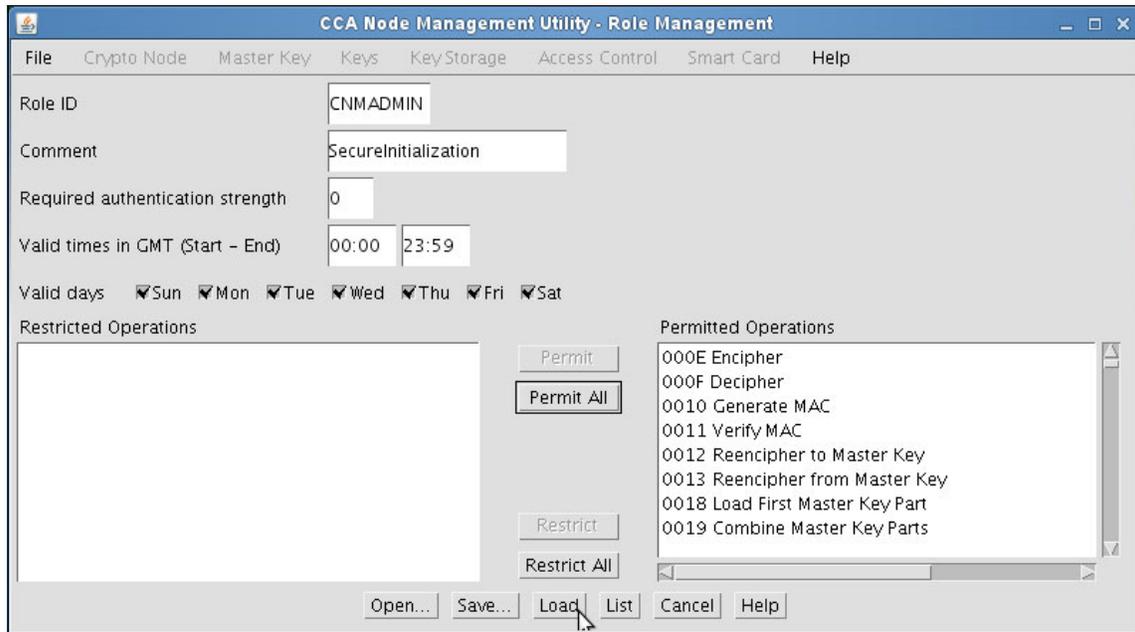


Figure B-61 Role Management

9. Create the CNMADMIN passphrase profile specifying a split passphrase. Click **Access Control** → **Profiles** from the CNM main window (Figure B-62), and then select **New** from the Profile Management window.

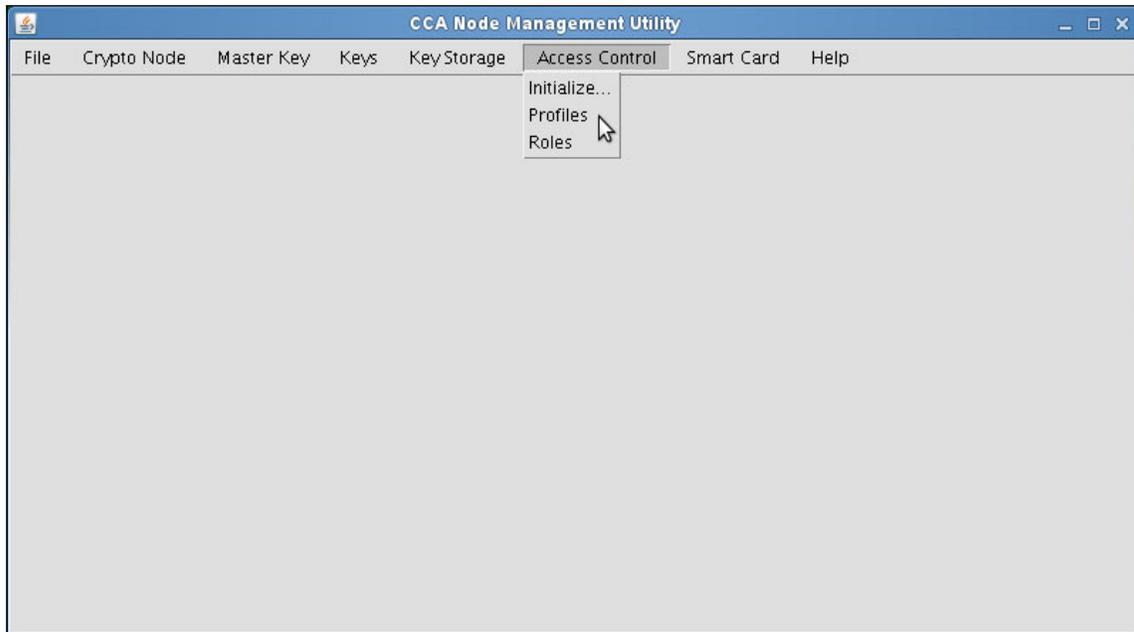


Figure B-62 CCA Node Management Utility - new profile

10. In the Profile Management window, enter the User ID (CNMADMIN) and select the Role (CNMADMIN). Enter an Expiration Date and a Passphrase Expiration Date that will not expire before you have finalized the secure setup of the IBM 4765 access control system. Enter the split passphrase twice and select **Load**, as shown in Figure B-63 on page 305. The following actions are required:
  - ADM1n must provide the first passphrase part (twice).
  - ADM2n must provide the last passphrase part (twice).

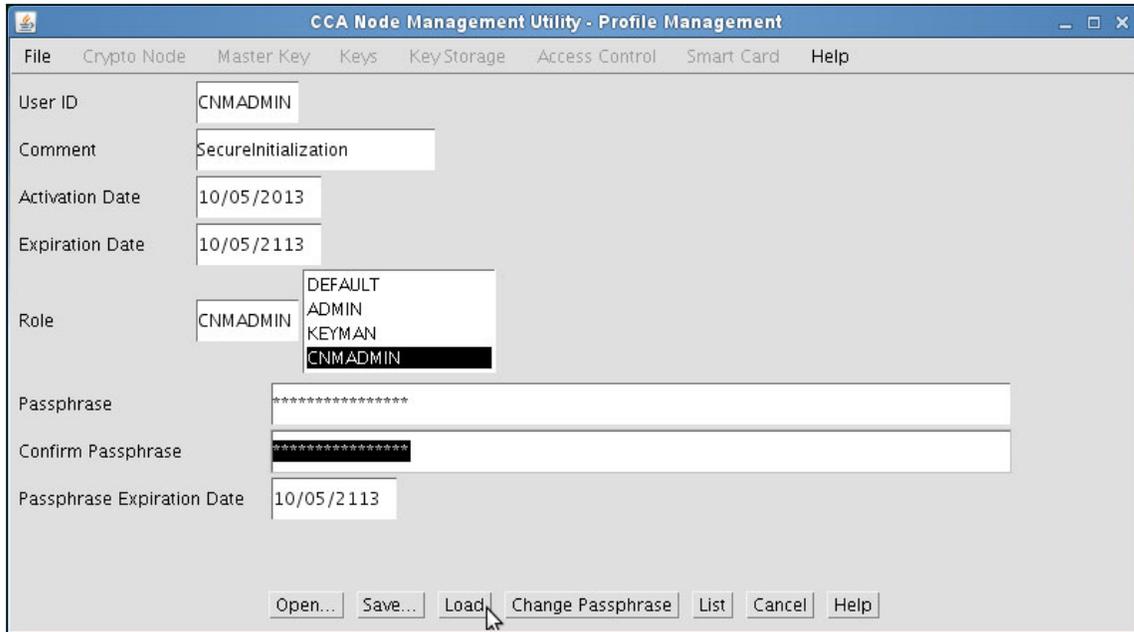


Figure B-63 Profile Management

11. Verify that you can log on by using the CNMADMIN passphrase profile, as described in “SCUP logon by using split passphrase” on page 290.  
If you fail to log on, then redo steps 7 on page 302 to 10 on page 304.
12. Delete the *tester* passphrase profile. Click **Access Control** → **Profiles** from the CNM main window, and then select the *tester* passphrase profile in the list that is shown in the Profile Management window that follows and select **Delete....**
13. Reduce the set of active access control points of the DEFAULT role to the following settings:
  - 001D Compute Verification Pattern
  - 0107 One-Way Hash, SHA-1
  - 0110 Set Clock
  - 0111 Reinitialize Device
  - 0116 Read Public Access-Control Information

Click **Access Control** → **Roles** from the CNM main window, and then select the DEFAULT Role from the Role Management window that opens. Next, click **Restrict All**, followed by the selection and permission of the five individual Access Control Points that are listed above, as shown in Figure B-64. Click **Load**.

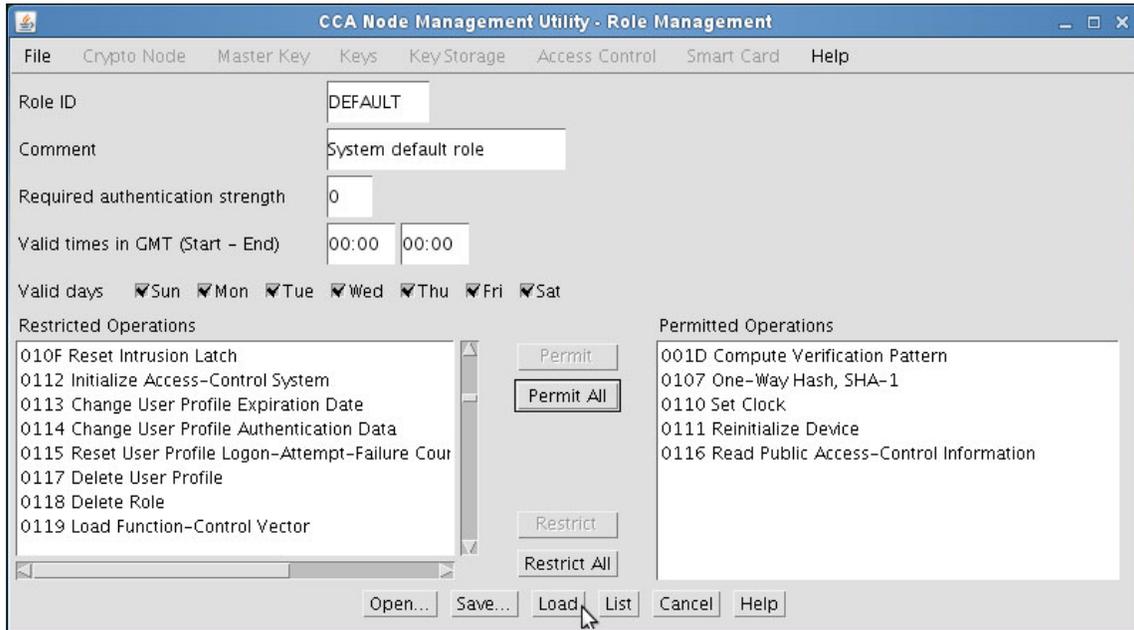


Figure B-64 Role Management

## Generating an IBM 4765 logon key on TKE smart card

The aim of this procedure is to instruct the CCA Node Management (CNM) utility program to generate a new unique logon RSA key pair.

The public part of the logon RSA key pair is afterward supposed to be *enrolled* in to the IBM PCIe 4765 Cryptographic Coprocessor, through the CNM utility, through the creation of a smart card profile.

Furthermore, the new smart card profile must be included in one or more Smart Card Group Profiles in the IBM 4765 through the usage of the CNM utility.

### Participants

Table B-13 on page 307 lists the participants with their roles and a brief description.

Table B-13 Participants

| Role indication | Role description                                                                                |
|-----------------|-------------------------------------------------------------------------------------------------|
| ADM1n           | The holder of either a first passphrase part of a CNMADMIN profile or an ADM1n logon smart card |
| ADM2n           | The holder of either a last passphrase part of a CNMADMIN profile or an ADM2n logon smart card  |
| ADM1n / ADM2n   | Responsible for registering the TKE smart cards                                                 |
| CARD HOLDER     | The holder of the TKE smart card to be personalized                                             |

### Special requirements

Participants that are listed in Table B-13 need the following components:

- ▶ ADM1n must have either of these two logon credentials:
  - An envelope containing the *first* passphrase part of the (initial/temporary) CNMADMIN profile
  - An envelope containing an ADM1n logon smart card, and an envelope containing the PIN Form for the logon smart card
- ▶ ADM2n must have either of these two logon credentials:
  - An envelope containing the *last* passphrase part of the (initial/temporary) CNMADMIN profile
  - An envelope containing a ADM2n logon smart card, and an envelope containing the PIN Form for the logon smart card
- ▶ CARD HOLDER must have the following components:
  - An envelope with a personalized TKE smart card
  - An envelope with a PIN Form for the personalized TKE smart card

ADM1n / ADM2n are responsible for registering the TKE smart card in the inventory of smart cards. This registration should include the following items:

- ▶ Zone ID/description (inherited from the CA smart card),
- ▶ Card ID *and description*,
- ▶ CARD HOLDER identification/name,
- ▶ Storage location,
- ▶ Intended use (crypto adapter logon),
- ▶ Profile ID/name.

CARD HOLDER is responsible for the custody of the TKE smart card.

## Procedure: Generating an IBM 4765 logon key on a TKE smart card

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**), as shown in Figure B-65.

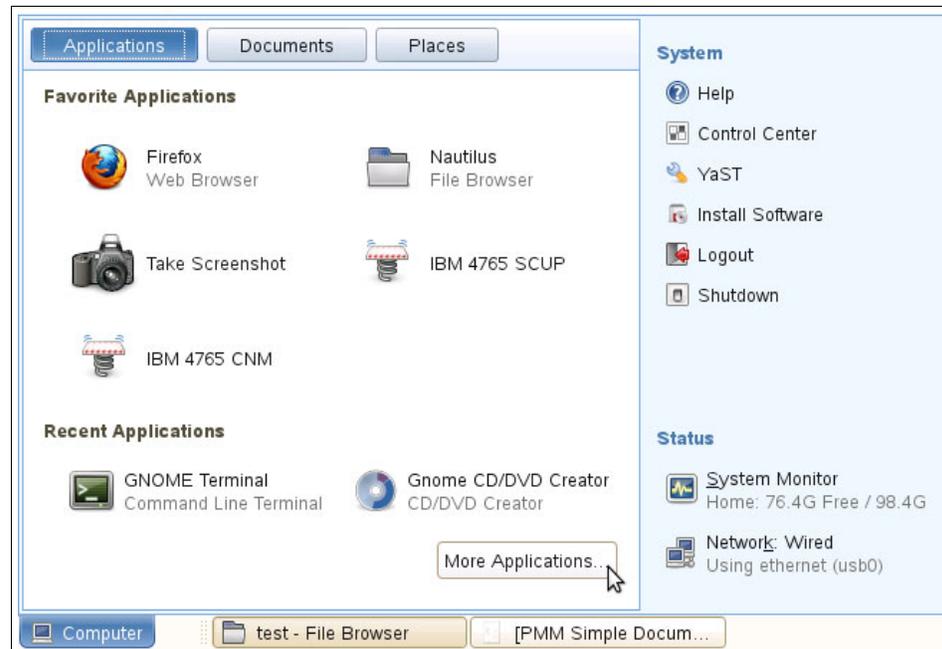


Figure B-65 Start the IBM 4765 CNM application

2. ADM1n and ADM2n must perform a CNM logon to the IBM PCIe 4765 Cryptographic Coprocessor to authorize the logon key generation. You can perform this task by using either of the following procedures:
  - “Performing a CNM Utility logon by using a split passphrase” on page 339
  - “Performing a CNM Utility group logon by using smart cards” on page 346
3. Click **Smart Card** → **Generate Crypto Adapter Logon Key**, as shown in Figure B-66 on page 309.

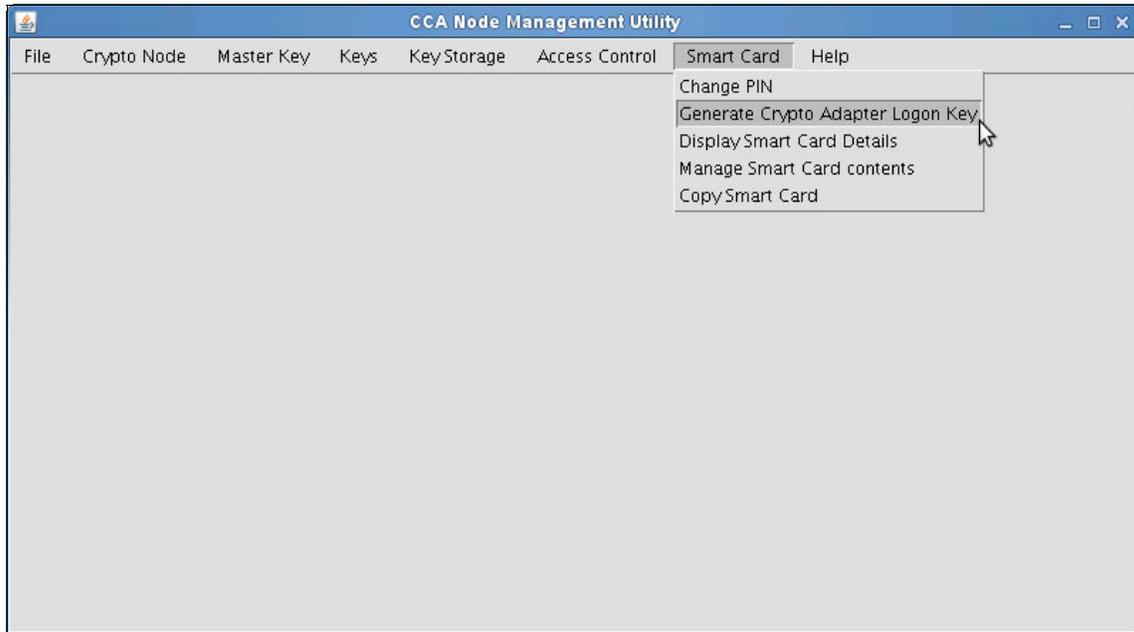


Figure B-66 Click Generate Crypto Adapter Logon Key

4. CARD HOLDER is prompted to insert the personalized TKE smart card in to reader 2, as shown in Figure B-67. Click **OK**.



Figure B-67 Insert a smart card

5. CARD HOLDER is prompted to enter the 6-digit PIN on reader 2 as shown in Figure B-68.



Figure B-68 Enter PIN

6. CARD HOLDER is prompted to enter a user ID for the personalized TKE smart card, as shown in Figure B-69. The user ID later is inherited as the name of a smart card profile. Click **OK**.



*Figure B-69 Enter a user ID*

7. A prompt confirms the success of the generation of the crypto adapter logon key on the TKE smart card, as shown in Figure B-70. Click **OK**.



*Figure B-70 Key generated confirmation*

8. The process can (optionally) be verified through the CNM smart card display function. Click **Smart Card** → **Display smart Card Details**, as shown in Figure B-71.

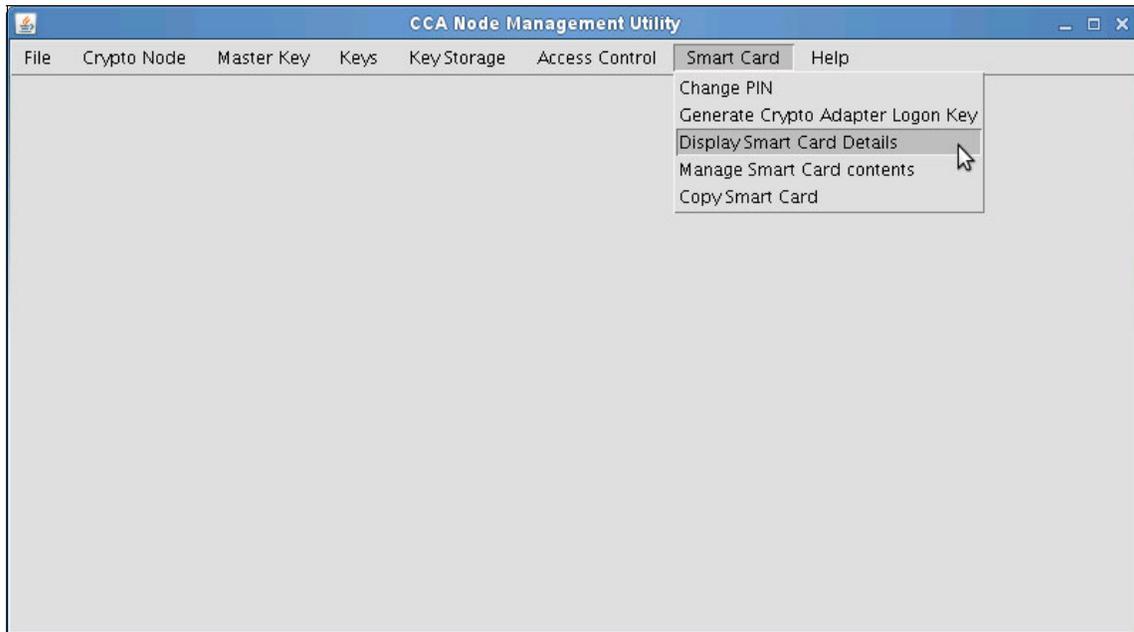


Figure B-71 Click Display Smart Card Details

After a few seconds, the smart card details window shows the Crypto adapter user ID, and that a Crypto adapter logon key is present, as shown in Figure B-72.

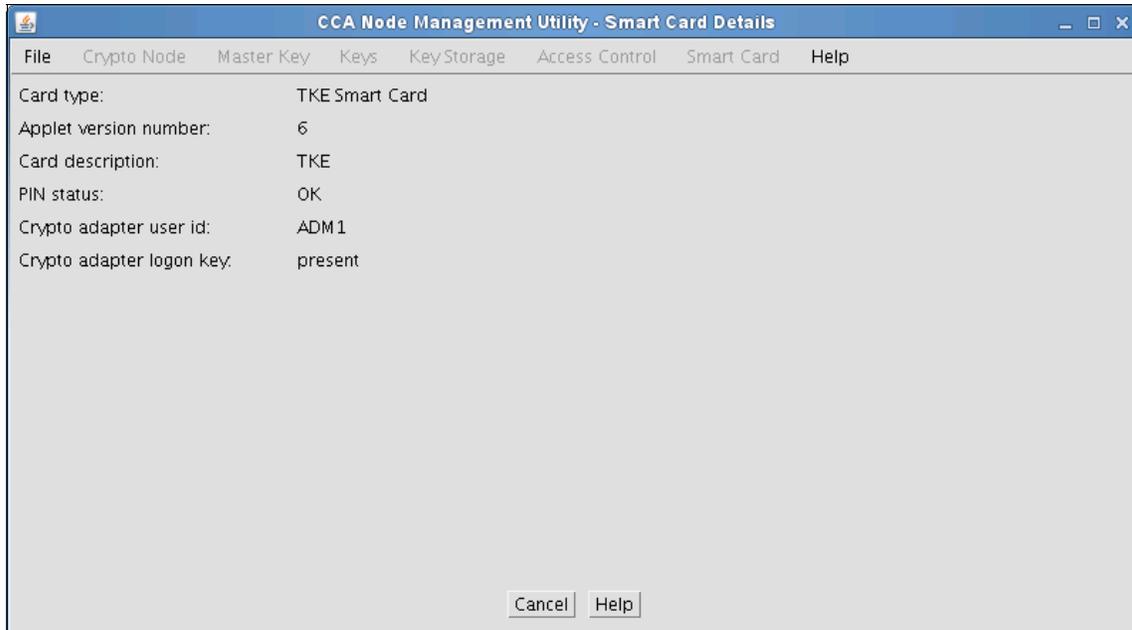


Figure B-72 Procedure complete

## Backing up a TKE smart card

The aim of this procedure is to create a backup of a TKE smart card, but it really is the process of copying contents from one TKE smart card to another card.

Both the source TKE smart card and the target TKE smart card must be initialized and enrolled in to the same crypto zone (using the same CA smart card). Both TKE smart cards must also be personalized (assigned a 6-digit PIN).

The contents that can be copied from the source TKE smart card are master key parts and a crypto adapter logon key.

### Participants

Table B-14 on page 313 lists the participants with their roles and a brief description.

Table B-14 Participants

| Role indication | Role description                                                                                  |
|-----------------|---------------------------------------------------------------------------------------------------|
| ADM1n           | The holder of either the first passphrase part of a CNMADMIN profile or an ADM1n logon smart card |
| ADM2n           | The holder of either the last passphrase part of a CNMADMIN profile or an ADM2n logon smart card  |
| ADM1n / ADM2n   | Responsible for registering the TKE smart cards                                                   |
| CARD HOLDER     | The holder of the source and target TKE smart cards                                               |

### Special requirements

The participants that are listed in Table B-14 need the following components:

- ▶ ADM1n must have either of these two logon credentials:
  - An envelope containing the *first* passphrase part of the (initial/temporary) CNMADMIN profile
  - An envelope containing an ADM1n logon smart card, and an envelope containing the PIN Form for the logon smart card
- ▶ ADM2n must have either of these two logon credentials:
  - An envelope containing the *last* passphrase part of the (initial/temporary) CNMADMIN profile
  - An envelope containing an ADM2n logon smart card, and an envelope containing the PIN Form for the logon smart card
- ▶ CARD HOLDER must have the following components:
  - An envelope with a source TKE smart card
  - An envelope with a PIN Form for the source TKE smart card
  - An envelope with a target TKE smart card
  - An envelope with a PIN Form for the target TKE smart card

ADM1n and ADM2n are responsible for registering TKE smart cards and their contents in the inventory of smart cards. This registration should include the following items:

- ▶ Zone ID/description (inherited from the CA smart card)
- ▶ Card ID *and description*
- ▶ CARD HOLDER identification/name
- ▶ Storage location
- ▶ Intended use (crypto adapter logon)

- ▶ Profile ID/name (if present)
- ▶ Master key parts (if present)

CARD HOLDER is responsible for the custody of the TKE smart card.

### Procedure: Backing up a TKE smart card

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).
2. ADM1n and ADM2n must perform a CNM logon to the IBM PCIe 4765 Cryptographic Coprocessor to authorize the copying of contents between two TKE smart cards. You can perform this task by using either of the following procedures:
  - “Performing a CNM Utility logon by using a split passphrase” on page 339
  - “Performing a CNM Utility group logon by using smart cards” on page 346
3. Click **Smart Card** → **Copy Smart Card**, as shown in Figure B-73.

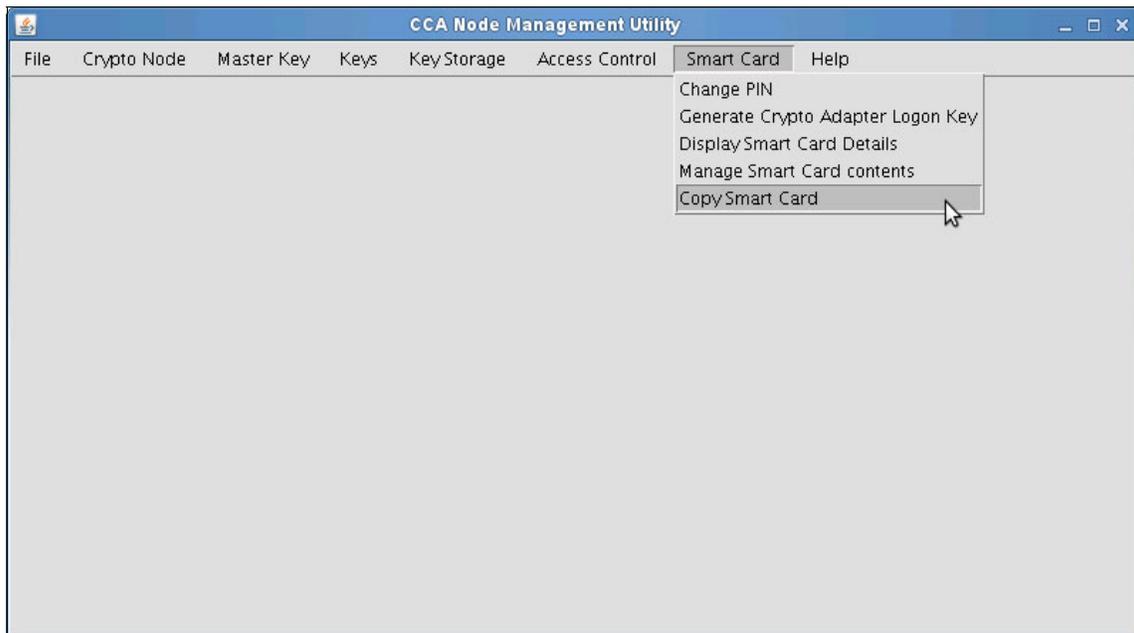


Figure B-73 Click Copy Smart Card

4. CARD HOLDER is prompted to insert the source TKE smart card in to reader 1. Select **OK** to continue.
5. CARD HOLDER is prompted to insert the target TKE smart card in to reader 2. Select **OK** to continue.

6. The Copy contents of Smart Card window shows the contents of the source and the target TKE smart cards, respectively.

A TKE smart card can contain the following items:

- One Crypto adapter Logon Key
- One or more Master Key parts, as shown in Figure B-74

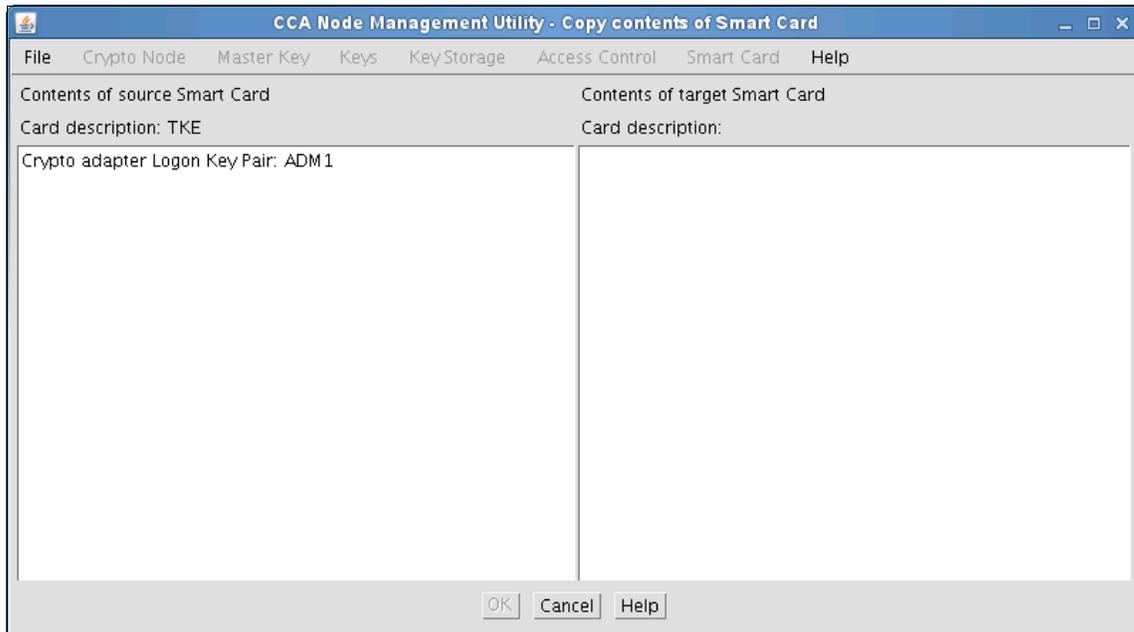


Figure B-74 Copy contents of Smart Card

7. CARD HOLDER is prompted to select the contents of the source TKE smart card that are to be copied to the target TKE smart card. Select **OK** to continue, as shown in Figure B-75.

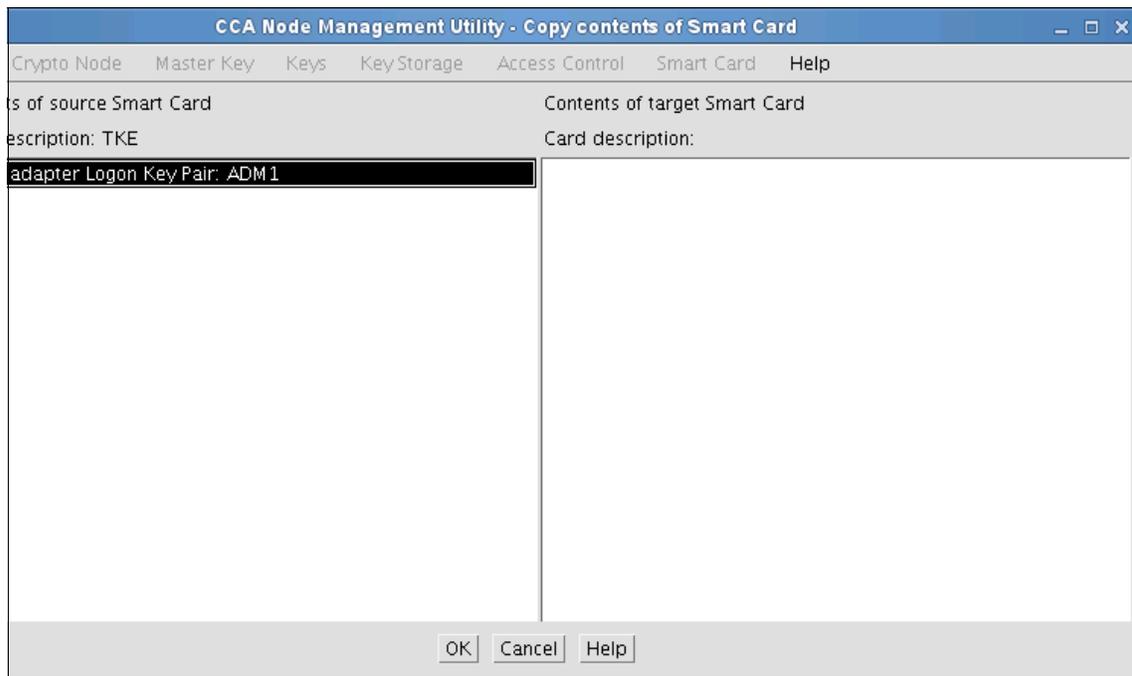


Figure B-75 Select the source content

8. CARD HOLDER is prompted to enter the 6-digit PIN of the source TKE Smart Card in reader 1, as shown in Figure B-76.



Figure B-76 Enter the PIN for the source card

9. CARD HOLDER is prompted to enter the 6-digit PIN of the target TKE Smart Card in reader 2, as shown in Figure B-77.

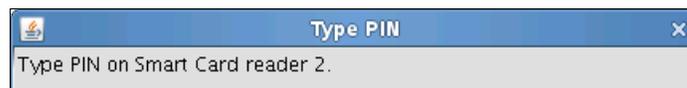


Figure B-77 Enter the PIN for the target card

A prompt indicates the establishment of a secure session between the two TKE smart cards. After a few seconds, a second prompt indicates that the copying is under way.

The Copy contents of Smart Cards window is updated to show the new contents of the target TKE smart card, as shown in Figure B-78.

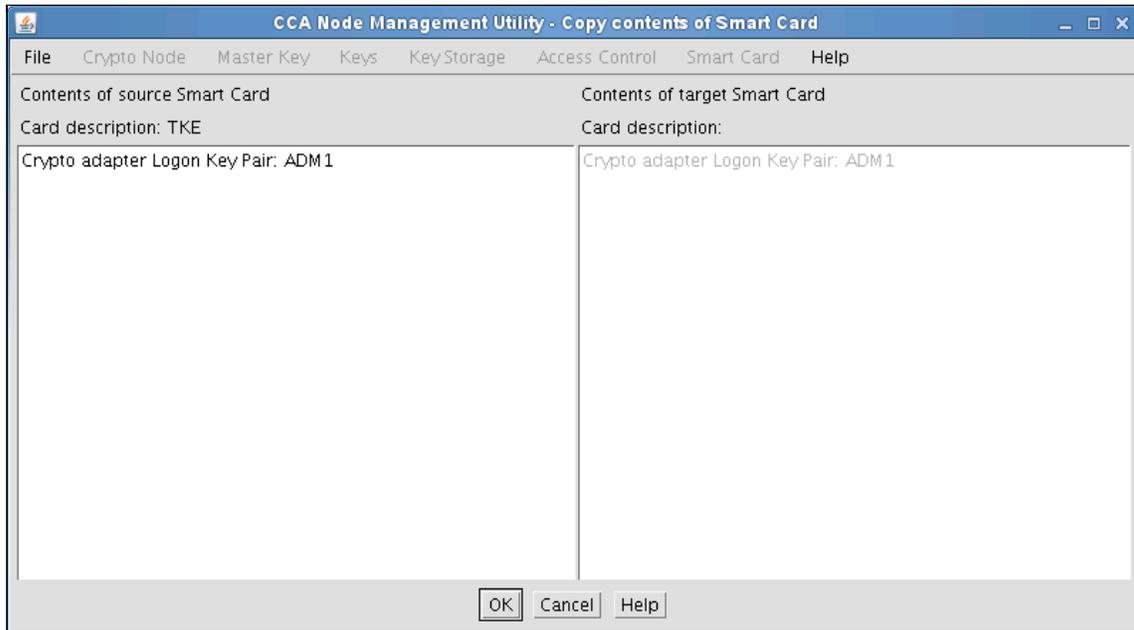


Figure B-78 Procedure complete

## Generating an IBM 4765 DES/PKA master key

The aim of this procedure is to randomly generate a DES/PKA master key in two parts, and store the parts on two separate TKE smart cards.

The two DES/PKA master key parts are generated by the IBM PCIe 4765 Cryptographic Coprocessor and transferred to the individual TKE smart cards in secure sessions.

The IBM PCIe 4765 Cryptographic Coprocessor and the two TKE smart cards that are used for the secure storage of the DES/PKA master key parts must belong to the same cryptographic zone.

**Note:** The generated DES/PKA master key parts are not installed within the IBM PCIe 4765 Cryptographic Coprocessor during this procedure.

## Participants

Table B-15 lists the participants with their roles and a brief description.

Table B-15 Participants

| Role indication | Role description                                                      |
|-----------------|-----------------------------------------------------------------------|
| SO1n            | The holder of an SO1n logon smart card and an MK1 key part smart card |
| SO2n            | The holder of an SO2n logon smart card and an MK2 key part smart card |

## Special requirements

Participants that are listed in Table B-15 need the following components:

- ▶ SO1n must have the following components:
  - An envelope containing the SO1n logon smart card
  - An Envelope containing the PIN Form for the logon smart card
  - An envelope containing the MK1 key part smart card
  - An envelope containing the PIN Form for the key part smart card
- ▶ SO2n must have the following components:
  - An envelope containing the SO2n logon smart card
  - An envelope containing the PIN Form for the logon smart card
  - An envelope containing the MK2 key part smart card
  - An envelope containing the PIN Form for the key part smart card

## Procedure: Generating an IBM 4765 DES/PKA master key

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).
2. Perform a CNM logon to the MANAGER group profile of the IBM PCIe 4765 Cryptographic Coprocessor to authorize generation of the IBM 4765 DES/PKA master key parts (see “Performing a CNM Utility group logon by using smart cards” on page 346).
3. SO1n clicks **Master Key** → **DES/PKA Master Keys** → **Smart Card Parts**, as shown in Figure B-79 on page 319.

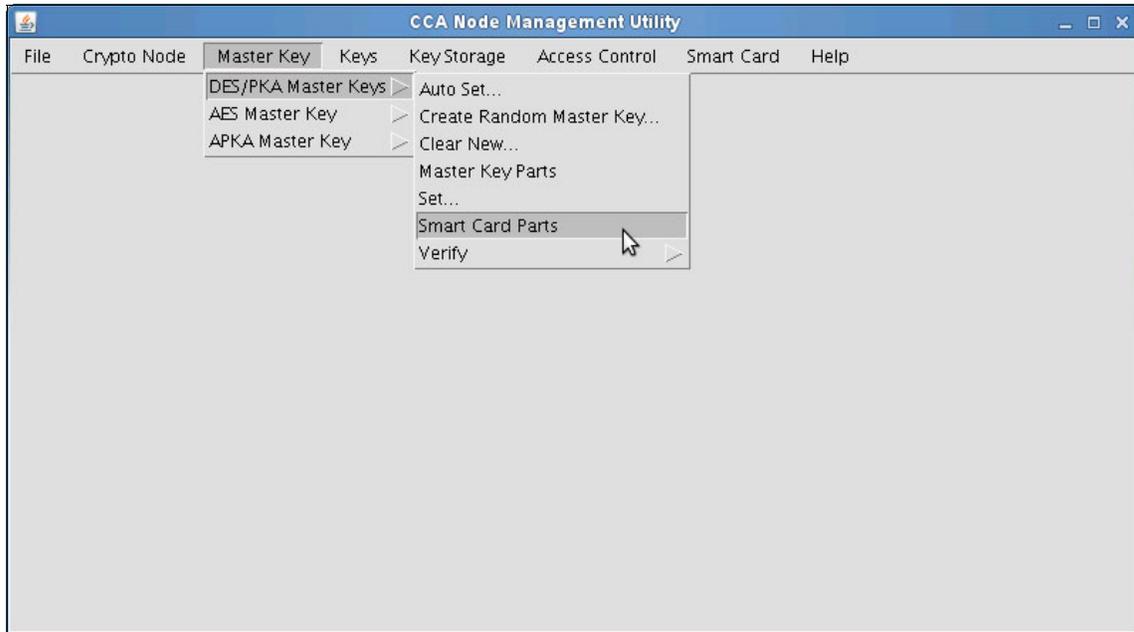


Figure B-79 Click Smart Card Parts

4. SO1n inserts the MK1 key part smart card that is intended to hold the first part of the new DES/PKA master key in to reader 2. Select **OK**, as shown in Figure B-80.

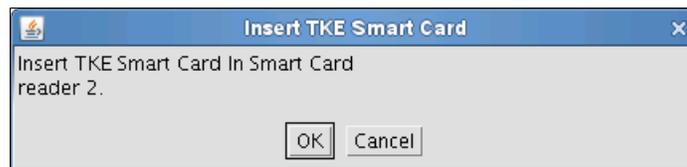


Figure B-80 Insert the smart card

5. SO1n is prompted to select the *first part* to be generated. Eventually, existing key parts that are stored on the smart card will be shown, but should *not* be selected.

Click **Generate & Save**, as shown in Figure B-81.

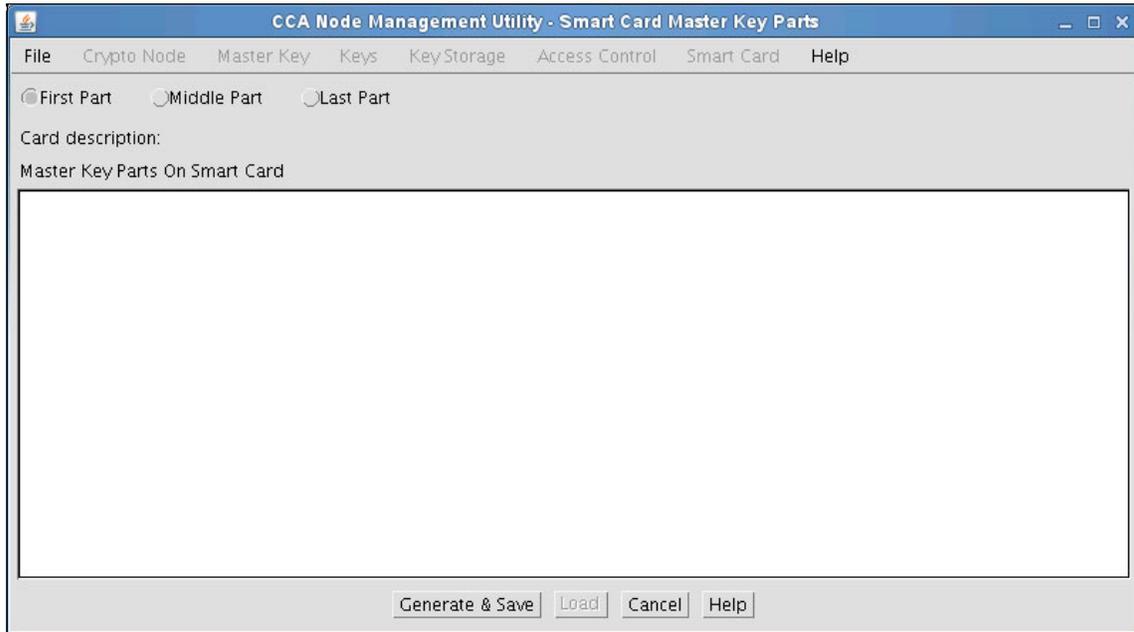


Figure B-81 Smart Card Master Key Parts

6. SO1n is prompted to enter a description for the *first part* of the new DES/PKA master key. Avoid First, Last, MK, and Part because they are already obvious in context. Here are some example descriptions:

- TEST <date-of-creation>
- PROD <date-of-creation>

Click **OK**, as shown in Figure B-82.



Figure B-82 Enter a description

7. SO1n is prompted to enter the 6-digit PIN of the MK1 key part smart card in reader 2, as shown in Figure B-83.

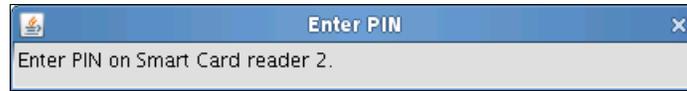


Figure B-83 Enter a PIN

A series of prompts that describe the generation and storage of the new DES/PKA master key part are shown in Figure B-84.



Figure B-84 Progression messages

After a few seconds, the first part of the generated DES/PKA master key part is shown in the Smart Card Master Key Parts window.

8. SO1n must now remove the MK1 key part smart card containing the new first part from reader 2. SO1n clicks **Cancel** to return to the Smart Card Master Key Parts window, as shown in Figure B-85.

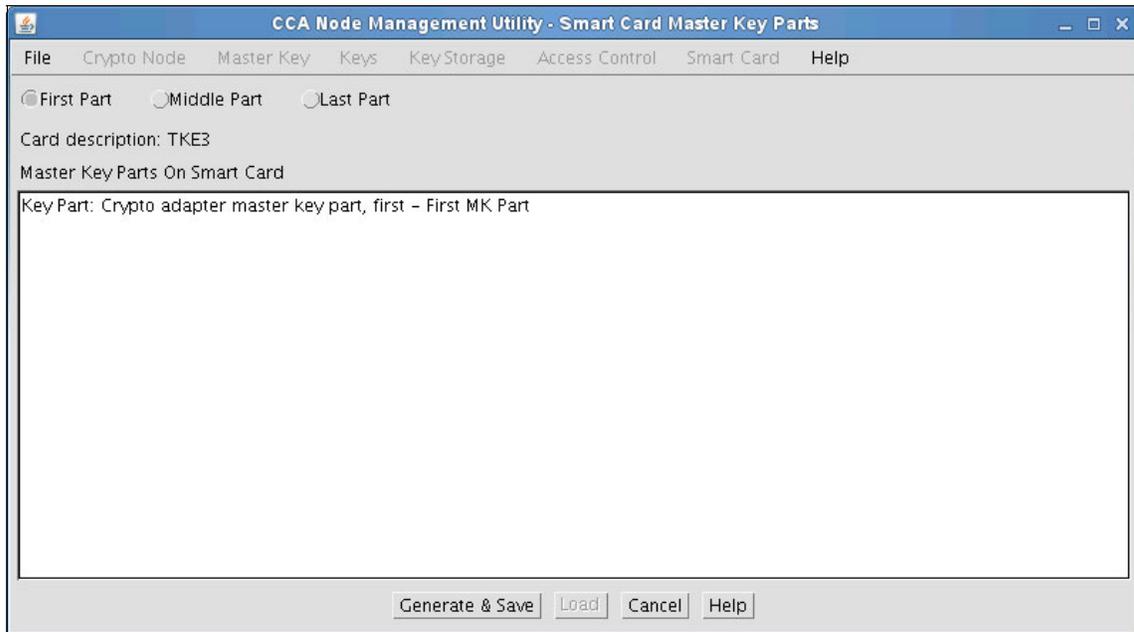


Figure B-85 Smart Card Master Key Parts

9. SO2n clicks **Master Key** → **DES/PKA Master Keys** → **Smart Card Parts**, as shown in Figure B-86 on page 323.

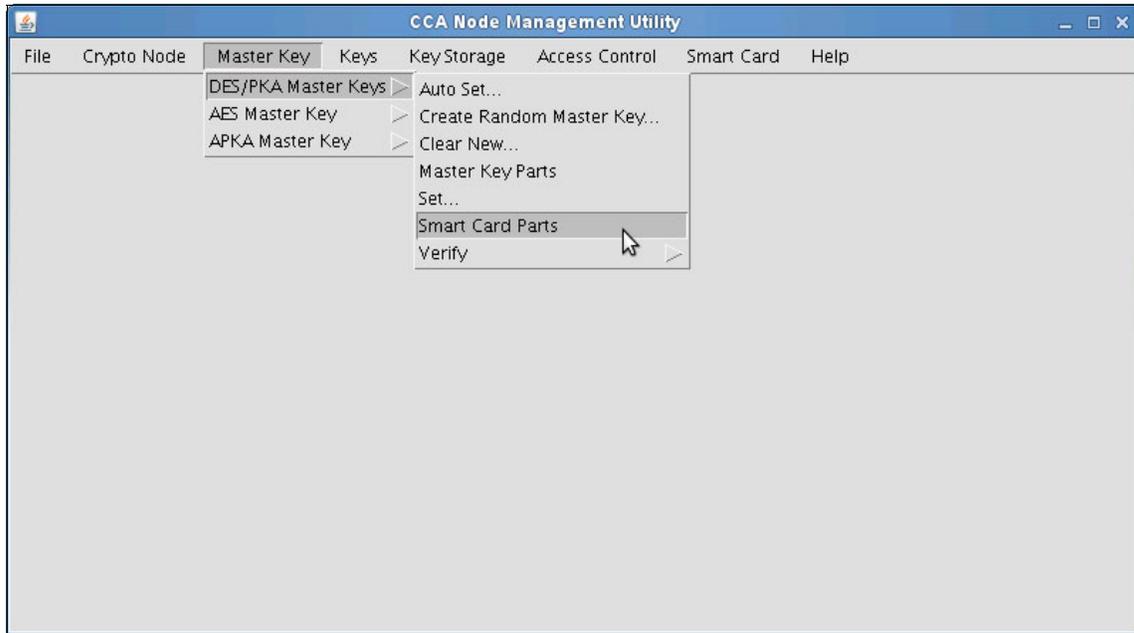


Figure B-86 Smart Card Parts

10. SO2n inserts the MK2 key part smart card that is intended to hold the last part of the new DES/PKA master key in reader 2. Select **OK** to continue.

11. SO2n selects the *last part* to be generated. Eventually, existing key parts that are stored on the smart card will be shown, but should *not* be selected. Click **Generate & Save**, as shown in Figure B-87.

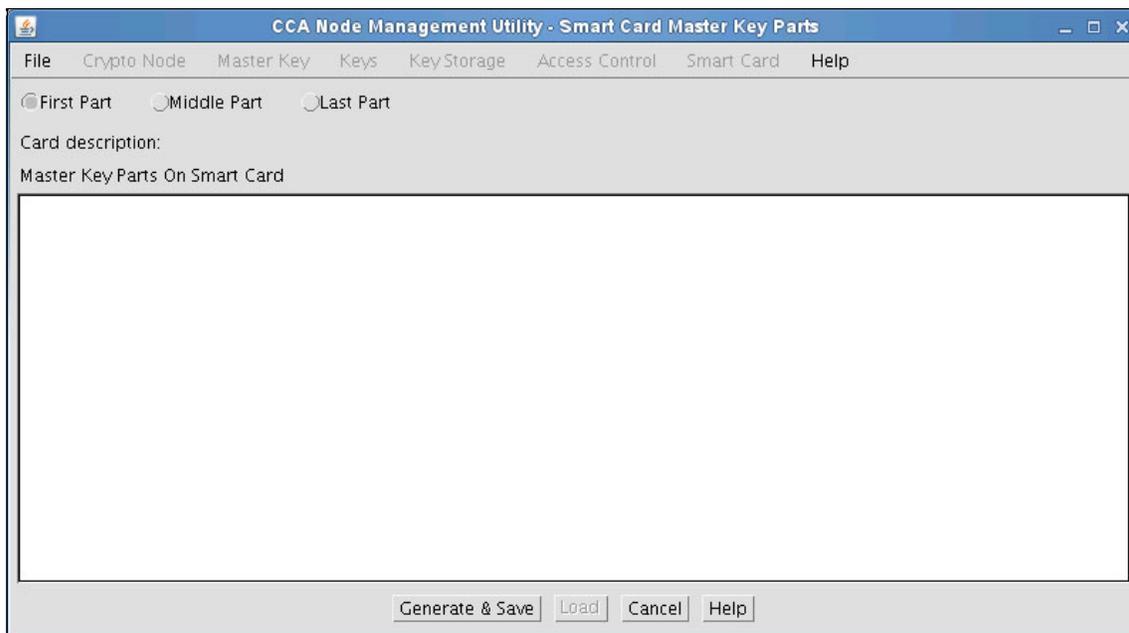


Figure B-87 Smart Card Master Key Parts

12. SO2n is prompted to enter a description for the *last part* of the new DES/PKA master key. Avoid First, Last, MK, and Part because they are already obvious in context. Here are some example descriptions:

- TEST <date-of-creation>
- PROD <date-of-creation>

Select **OK**, as shown in Figure B-88.

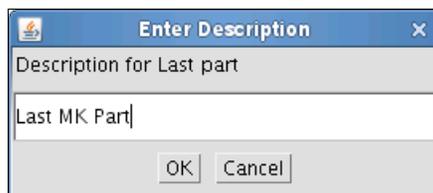


Figure B-88 Enter a description

13. SO2n is prompted to enter the 6-digit PIN of the MK1 key part smart card in reader 2, as shown in Figure B-89 on page 325.

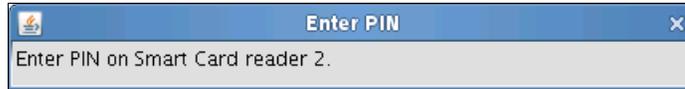


Figure B-89 Enter a PIN

14. A series of prompts that describe the generation and storage of the new DES/PKA master key part are shown in Figure B-90.



Figure B-90 Progression messages

After a few seconds, the last part of the generated DES/PKA master key part is shown in the Smart Card Master Key Parts window that is shown in Figure B-91.

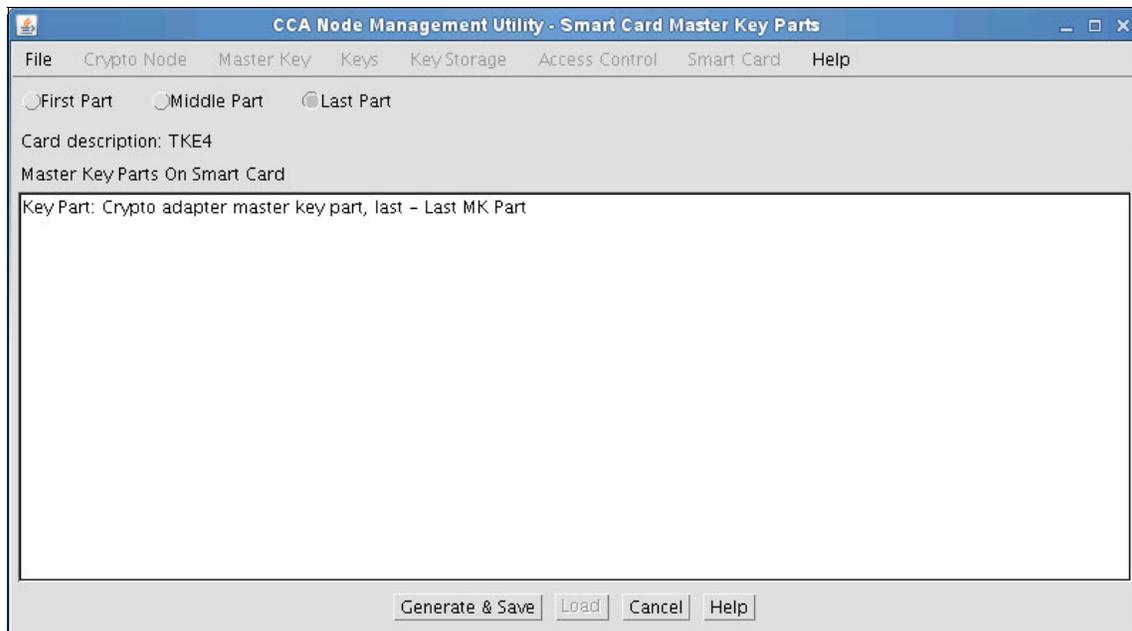


Figure B-91 Procedure complete

15. SO2n now must remove the MK2 key part smart card containing the new last part from reader 2. Select **Cancel** to return.

## Loading an IBM 4765 DES/PKA master key

The aim of this procedure is to load two IBM 4765 DES/PKA master key parts that are stored on individual TKE smart cards.

The IBM 4765 DES/PKA master key parts must be generated by an IBM PCIe 4765 Cryptographic Coprocessor that is enrolled into the same cryptographic zone as the coprocessor where the master key parts will be loaded.

### Participants

Table B-16 lists the participants with their roles and a brief description.

Table B-16 Participants

| Role indication | Role description                                                      |
|-----------------|-----------------------------------------------------------------------|
| SO1n            | The holder of an SO1n logon smart card and an MK1 key part smart card |
| SO2n            | The holder of an SO2n logon smart card and an MK2 key part smart card |

### Special requirements

The participants that are listed in Table B-16 need the following components:

- ▶ SO1n must have the following components:
  - An envelope containing the SO1n logon smart card
  - An envelope containing the PIN Form for the logon smart card
  - An envelope containing the MK1 key part smart card
  - An envelope containing the PIN Form for the key part smart card
- ▶ SO2n must have the following components:
  - An envelope containing the SO2n logon smart card
  - An envelope containing the PIN Form for the logon smart card
  - An envelope containing the MK2 key part smart card
  - An envelope containing the PIN Form for the key part smart card

### Procedure: Loading the IBM 4765 DES/PKA master key

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).

2. Perform a CNM logon to the MANAGER group profile of the IBM PCIe 4765 Cryptographic Coprocessor to authorize the loading of the IBM 4765 DES/PKA master key parts. To accomplish this task, see “Performing a CNM Utility group logon by using smart cards” on page 346.
3. Click **Crypto Node** → **Status**, as shown in Figure B-92.

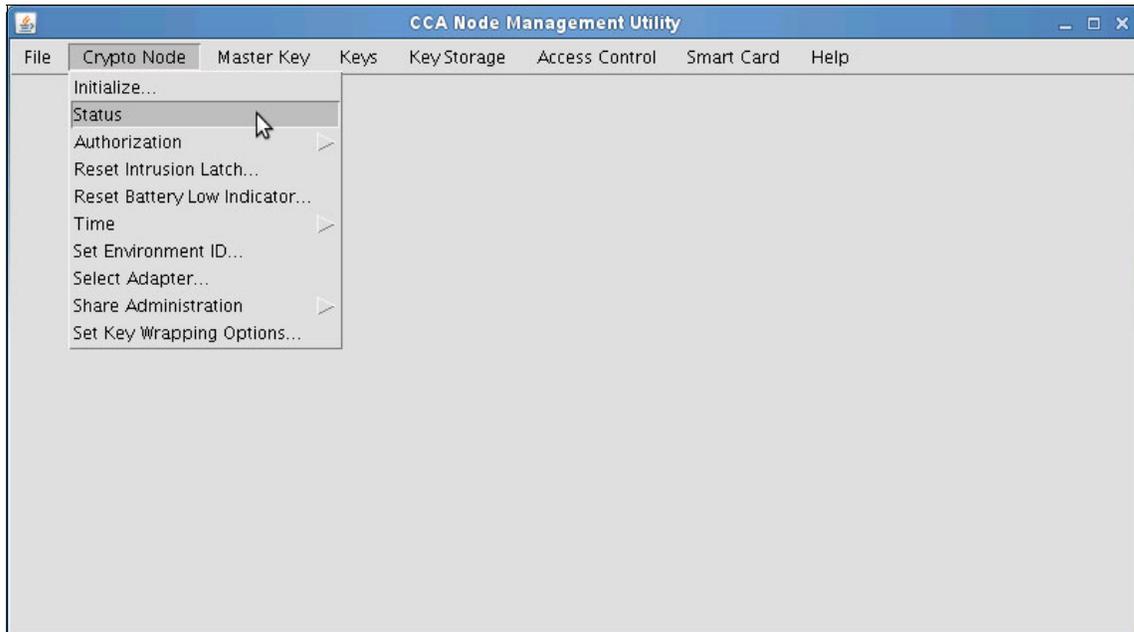


Figure B-92 Select Status

4. Verify that DES/PKA New master key register is set to Clear.  
If the DES/PKA New master key register is *not* set to Clear, then complete the following steps:
  - a. Click **Cancel**.
  - b. Click **Master Key** → **DES/PKA Master Keys** → **Clear New....**
  - c. At the prompt, click **Yes**.

Click **Cancel**, as shown in Figure B-93.

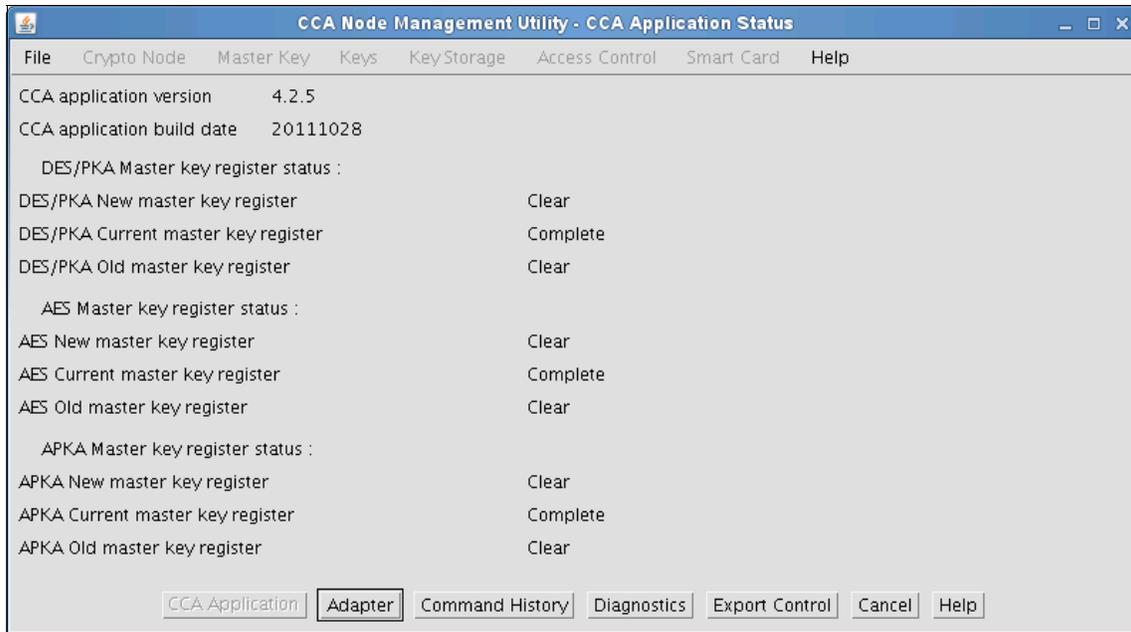


Figure B-93 CCA Application Status

5. Click **Master Key** → **DES/PKA Master Keys** → **Smart Card Parts.**, as shown in Figure B-94 on page 329.

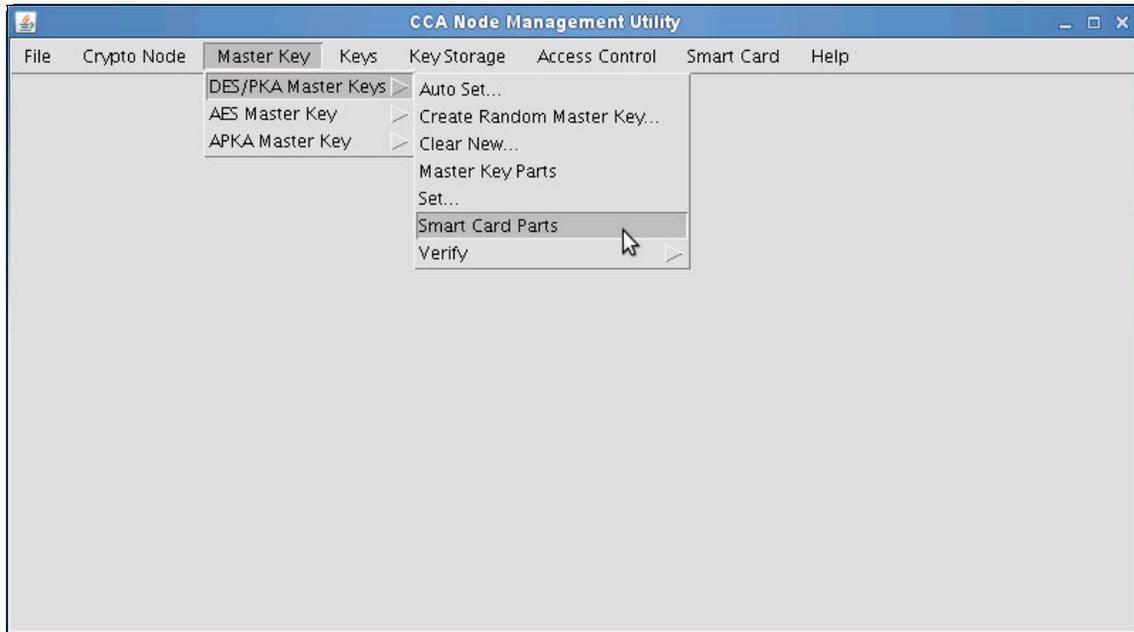


Figure B-94 Select Smart Card Parts

6. SO1n is prompted to insert the MK1 key part smart card that contains the first part of the DES/PKA master key in to reader 2. Click **OK** to continue.

- SO1n selects **First Part**, then selects the master key first part to be loaded from the Master Key Parts On Smart Card list. Click Load, as shown in Figure B-95.

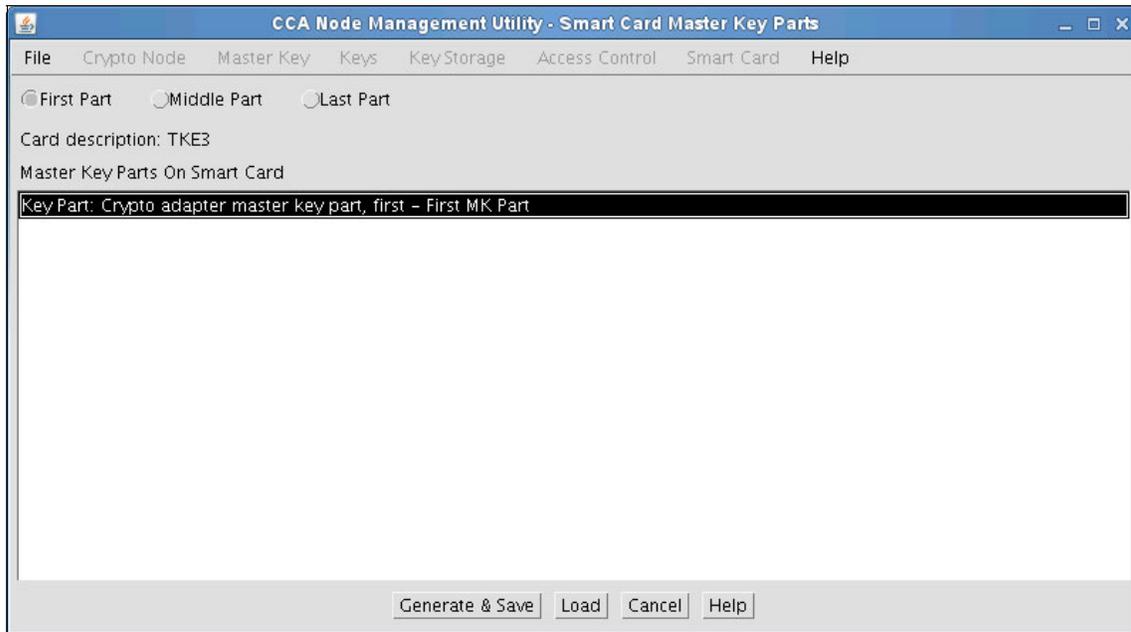


Figure B-95 Load Master Key Part

- SO1n is prompted to enter the 6-digit PIN of the MK1 key part smart card in reader 2, as shown in Figure B-96.



Figure B-96 Enter PIN

- A series of prompts indicate that a secure session is established and the key part is being loaded, as shown in Figure B-97.



Figure B-97 Progression messages

10. After a few seconds, a new prompt indicates a successful load of the first DES/PKA master key part. Select **OK**.  
SO1n must now remove the MK1 key part smart card from reader 2. SO1n selects **Cancel**.
11. Click **Master Key** → **DES/PKA Master Keys** → **Smart Card Parts**, as shown in Figure B-98.

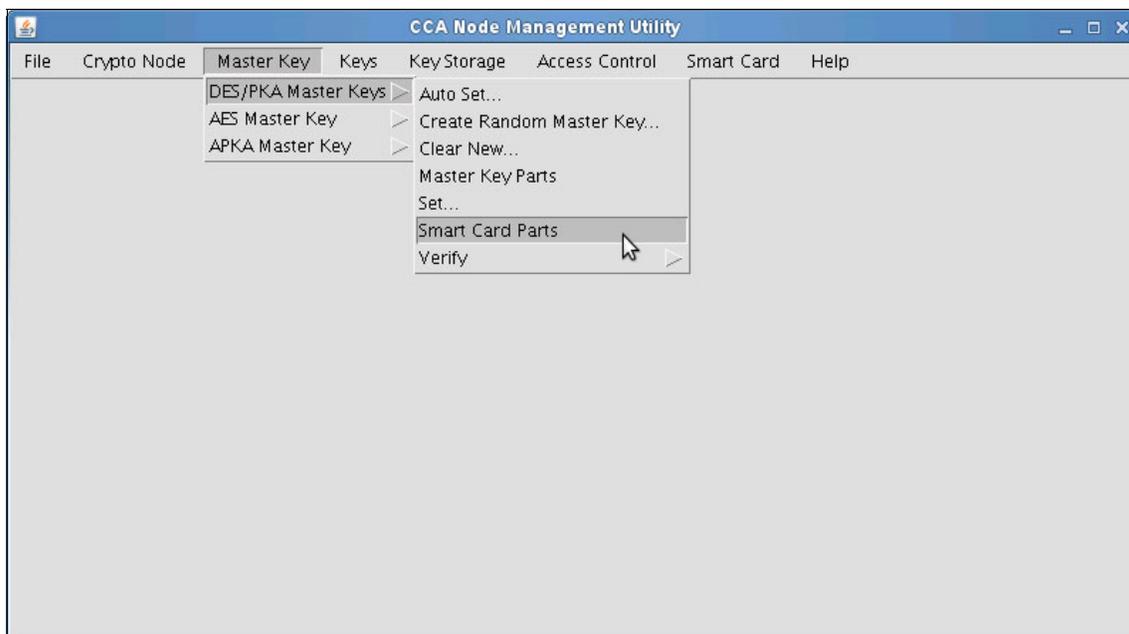


Figure B-98 Select Smart Card Parts

12. SO2n is prompted to insert the MK2 key part smart card that contains the last part of the DES/PKA master key in reader 2. Click **OK**, as shown in Figure B-99.



Figure B-99 Insert smart card

13. SO2n clicks **Last Part**. SO2n then selects the master key last part to be loaded from the Master Key Parts On Smart Card list. Click **Load**, as shown in Figure B-100.

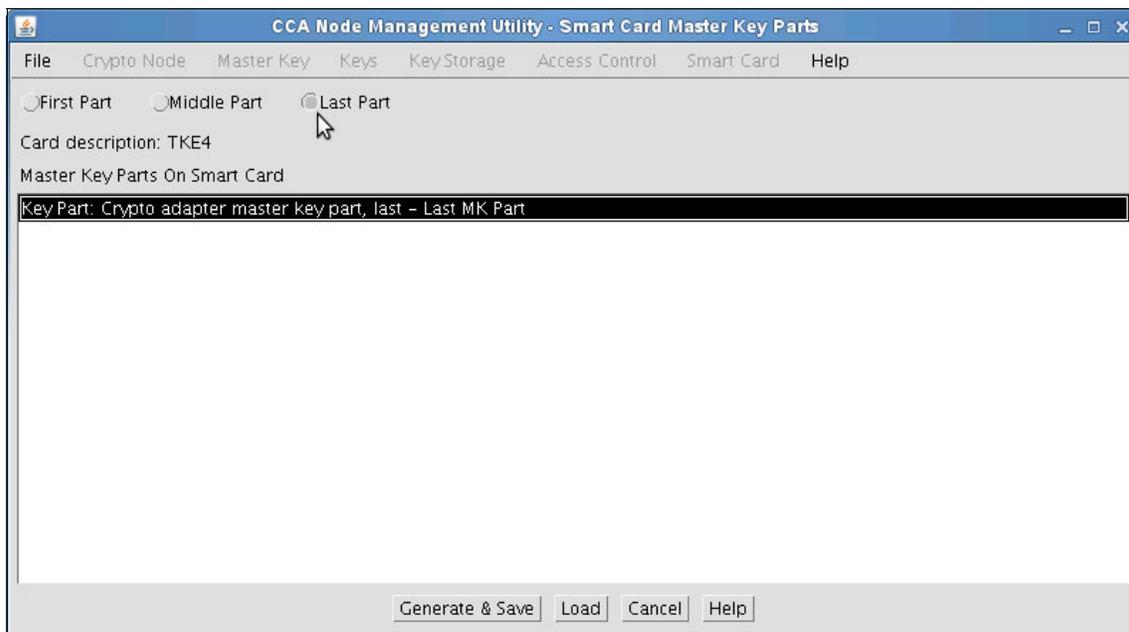


Figure B-100 Load Master Key Part

14. SO2n is prompted to enter the 6-digits PIN of the MK2 key part smart card in reader 2, as shown in Figure B-101.



Figure B-101 Enter PIN

15. A series of prompts indicate that a secure session is established and the key part is being loaded, as shown in Figure B-102.

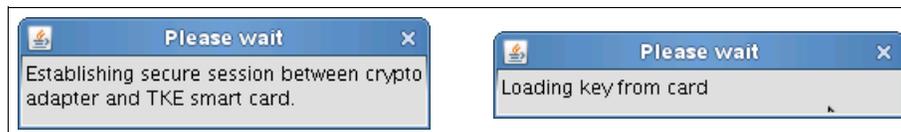


Figure B-102 Progression messages

16. After a few seconds, a new prompt indicates a successful load of the last master key part. Select **OK**. SO2n must now remove the MK2 key part smart card from to reader 2. SO2n clicks **Cancel**.
17. Click **Crypto Node** → **Status**, as shown in Figure B-103.

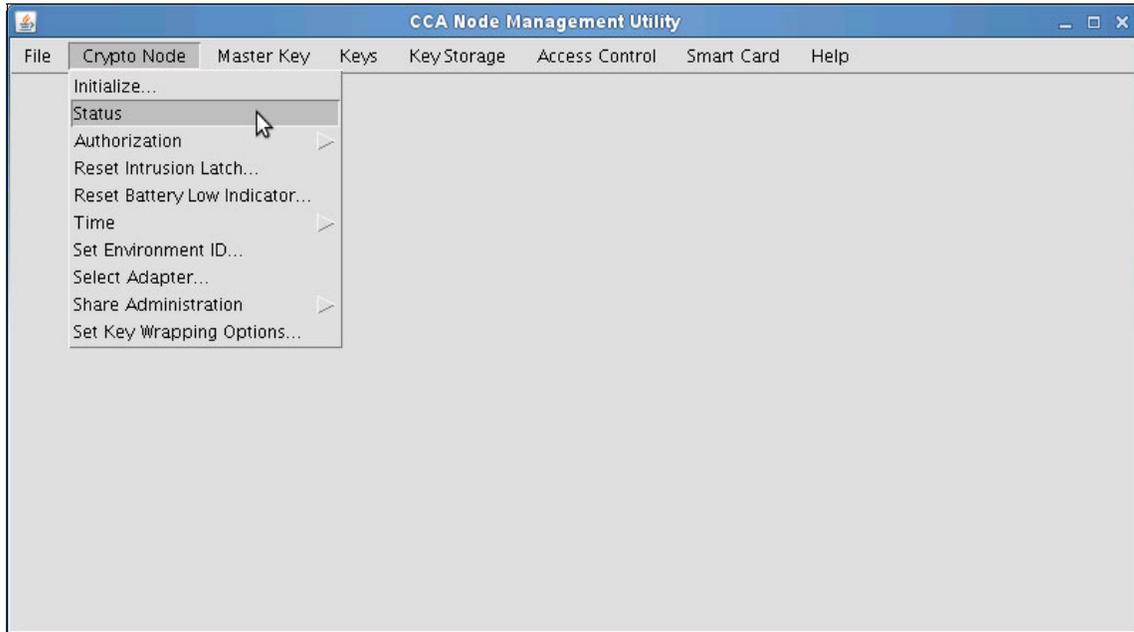


Figure B-103 Select Status

18. Verify that the DES/PKA New master key register is set to Complete, as shown in Figure B-104. If so, the procedure is complete.

If the DES/PKA New master key register is not set to Complete, then redo steps 3 on page 327 to 17 on page 333.

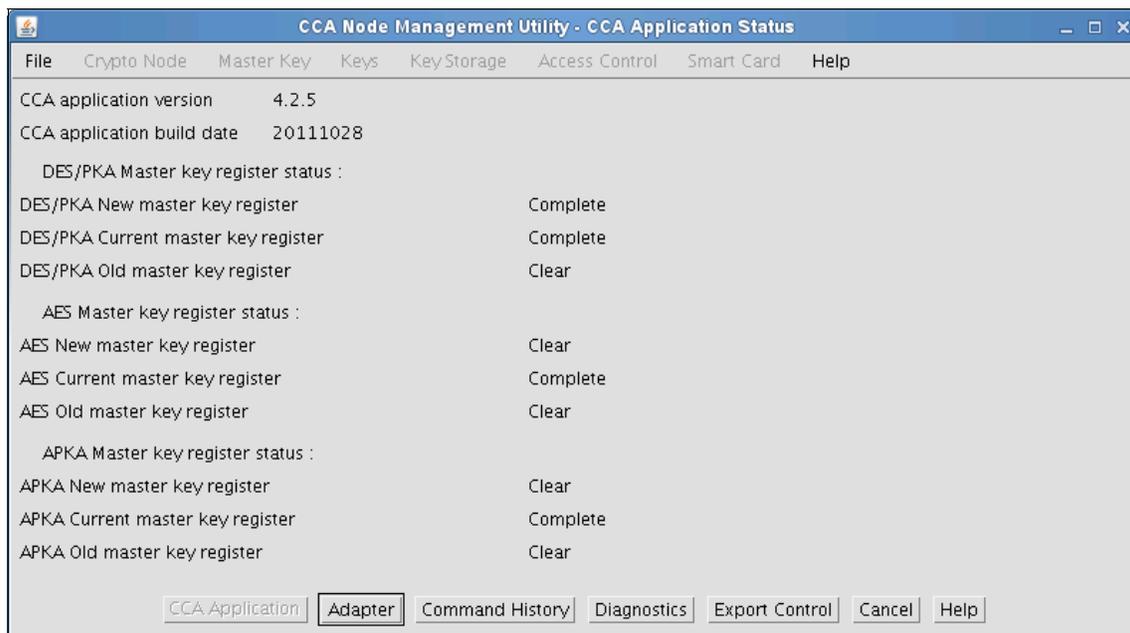


Figure B-104 Procedure complete

## Setting the IBM 4765 DES/PKA master keys and re-enciphering the key storage

The aim of this procedure is to set the DES/PKA master keys in the new master key registers to become the new current master keys of the IBM PCIe 4765 Cryptographic Coprocessor.

Then, the DES and PKA key storages are re-enciphered from the old master keys to the new current master keys.

### Participants

Table B-17 on page 335 lists the participants with their roles and a brief description.

Table B-17 Participants

| Role indication | Role description                       |
|-----------------|----------------------------------------|
| SO1n            | The holder of an SO1n logon smart card |
| SO2n            | The holder of an SO2n logon smart card |

### Special requirements

Participants that are listed in Table B-17 need the following components:

- ▶ SO1n must have the following components:
  - An envelope containing the SO1n logon smart card
  - An envelope containing the PIN Form for the logon smart card
- ▶ SO2n must have the following components:
  - An envelope containing the SO2n logon smart card
  - AN envelope containing the PIN Form for the logon smart card

### Procedure: Setting the IBM 4765 DES/PKA master keys and re-enciphering the key storage

Complete the following steps:

1. Perform a CNM logon to the MANAGER group profile of the IBM PCIe 4765 Cryptographic Coprocessor to authorize activation of the IBM 4765 DES/PKA master key. To accomplish this task, see “Performing a CNM Utility group logon by using smart cards” on page 346.

2. Click **Master Key** → **DES/PKA Master Keys** → **Set...**, as shown in Figure B-105.

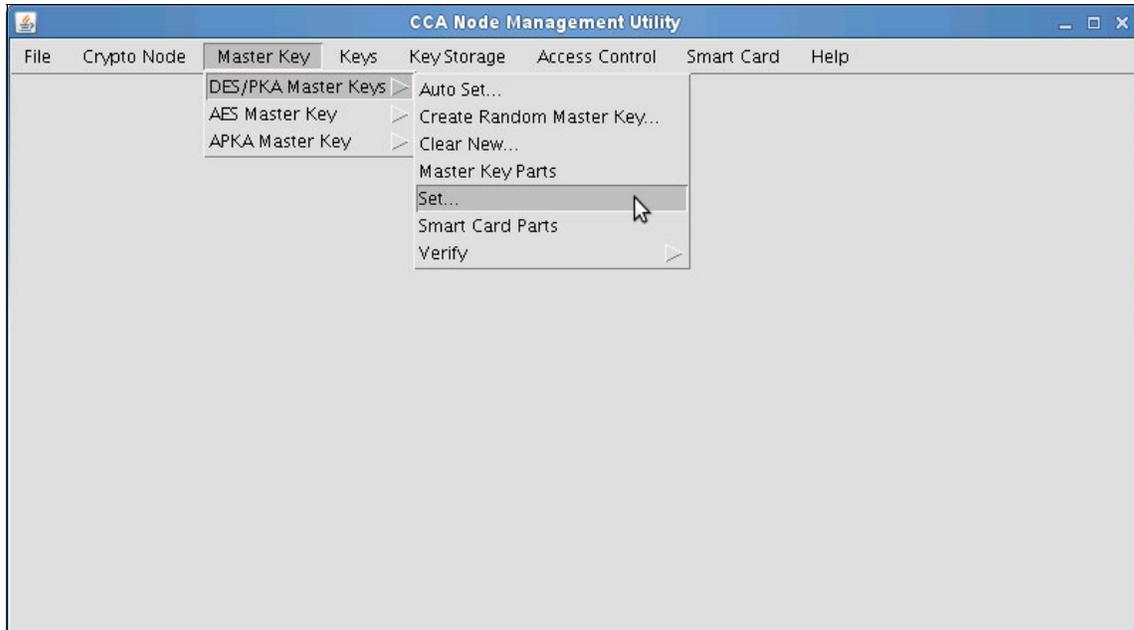


Figure B-105 Set the DES/PKA Master Keys

3. You are prompted to confirm the move from the new master key register to the current master key. Click **Yes**, as shown in Figure B-106.



Figure B-106 Set New Master Key

4. After a few seconds, a new prompt indicates that the current master key registers are successfully set. Click **OK**, as shown in Figure B-107.



Figure B-107 Success message

5. Click **Key Storage** → **DES Key Storage** → **Manage**, as shown in Figure B-108.

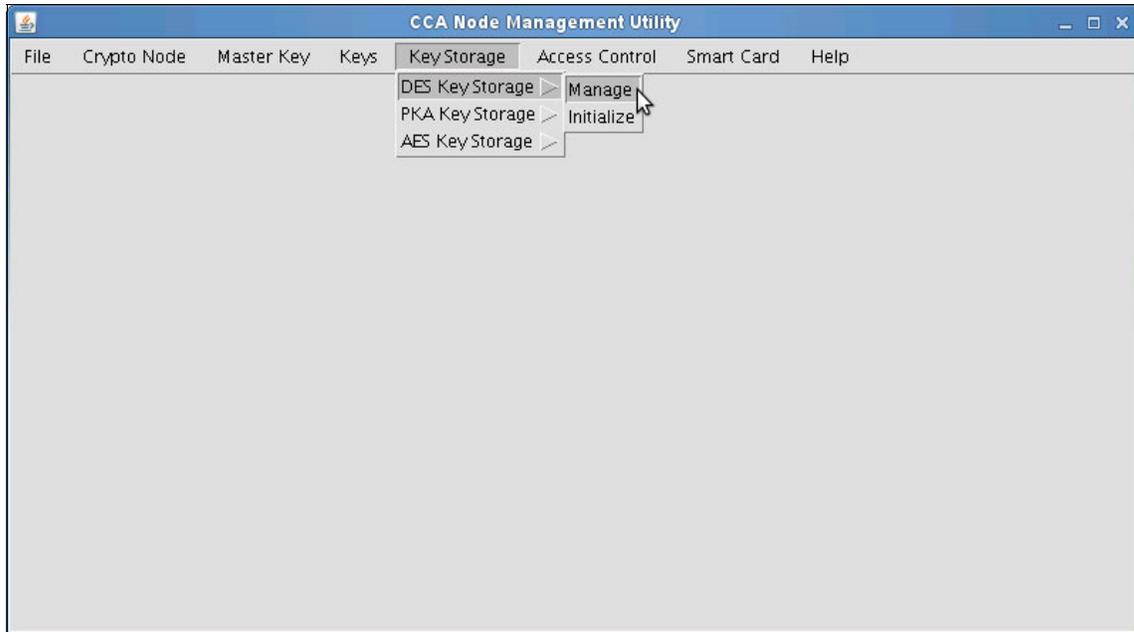


Figure B-108 Manage the DES Key Storage

6. The content of the DES key storage (file) is listed.

Even though there are no keys in the DES key storage, it must be re-enciphered to adapt the new master key. Click **Reencipher**, as shown in Figure B-109.

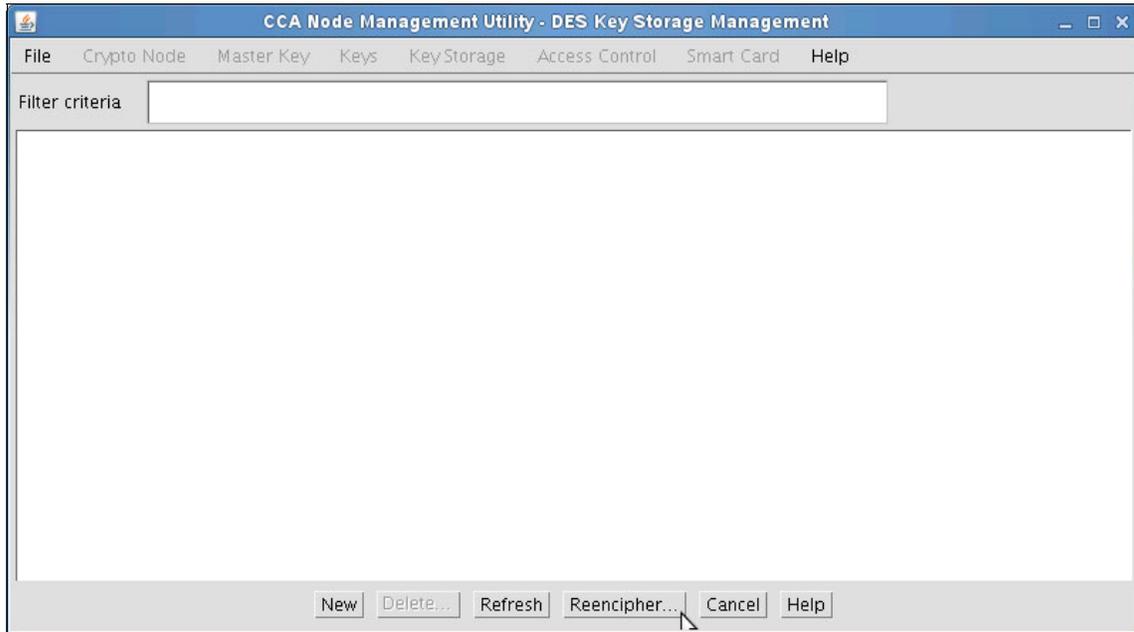


Figure B-109 Re-encipher key

7. You are prompted to confirm the re-encipherment of the DES key storage from the old master key to the new current master key. Click **Yes**, as shown in Figure B-110.



Figure B-110 Re-encipher key storage

8. Click **Key Storage** → **PKA Key Storage** → **Manage**.

9. The content of the PKA key storage (file) is listed.

Even though there are no keys in the PKA key storage, it must be re-enciphered to adapt the new master key. Click **Reencipher**.

10. You are prompted to confirm the re-encipherment of the PKA key storage from the old master key to the new current master key. Click **Yes**.
11. Click **Key Storage** → **AES Key Storage** → **Manage**.
12. The content of the AES key storage (file) is listed.  
Although the list is (supposed to be) empty, the AES key storage (file) must adapt to an eventual new master key. Click **Reencipher**.
13. You are prompted to confirm the re-encipherment of the PKA key storage from the old master key to the new current master key. Click **Yes**.

## Performing a CNM Utility logon by using a split passphrase

The aim of this procedure is to log on to the IBM PCIe 4765 Cryptographic Coprocessor from the CNM utility by using a split knowledge passphrase.

In the initial phase of the secure setup of the EKMF workstation, the temporary CNMADMIN passphrase profile is created in the IBM PCIe 4765 Cryptographic Coprocessor. The profile passphrase is split so that one person knows the first half and another person knows the last half. The intended use is to enforce dual-controlled access to necessary commands (access-control points) during the secure setup of the IBM PCIe 4765 Cryptographic Coprocessor.

### Participants

Table B-18 lists the participants with their roles and a brief description.

*Table B-18 Participants*

| Role indication | Role description                                                       |
|-----------------|------------------------------------------------------------------------|
| ADM1n           | The holder of the first half of the passphrase of the CNMADMIN profile |
| ADM2n           | The holder of the last half of the passphrase of the CNMADMIN profile  |

### Special requirements

Participants that are listed in Table B-18 need the following components.

- ▶ ADM1n must have the envelope with the first half of the passphrase for the CNMADMIN profile.
- ▶ ADM2n must have the envelope with the last half of the passphrase for the CNMADMIN profile.

## Procedure: Performing a CNM Utility logon by using a split passphrase

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).
2. Click **File** → **Passphrase Logon...**, as shown in Figure B-111.

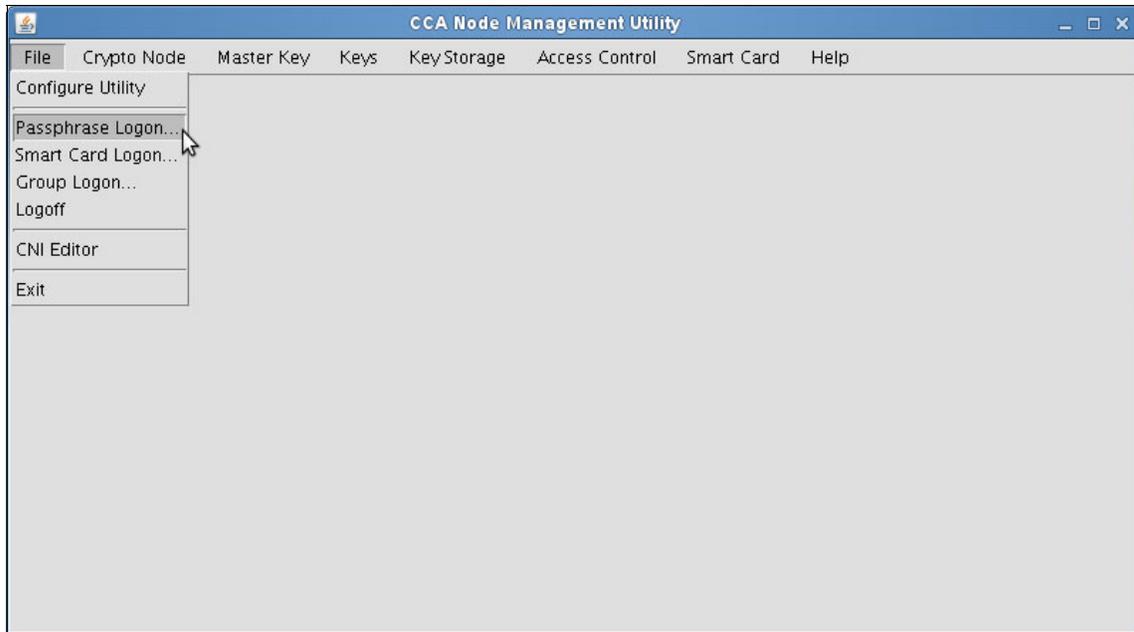


Figure B-111 Select Passphrase Logon

3. ADM1n is prompted to select the passphrase profile to log on to. Enter CNMADMIN as the user ID.
4. ADM1n then must enter the *first half* of the split passphrase, as shown in Figure B-112.

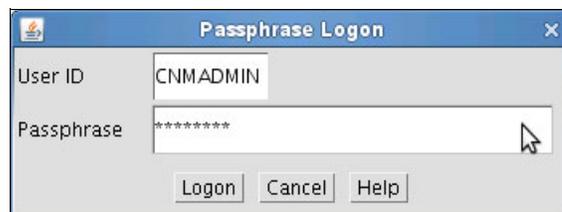


Figure B-112 Provide the first half of the passphrase

5. ADM2n is prompted to enter the *last half* of the split passphrase. Click **Logon**, as shown in Figure B-113.



Figure B-113 Log on with the second half of the passphrase

6. The CCA Node Management utility program (IBM 4765 CNM) indicates that the logon using the split passphrase is successful by opening the main window, as shown in Figure B-114.

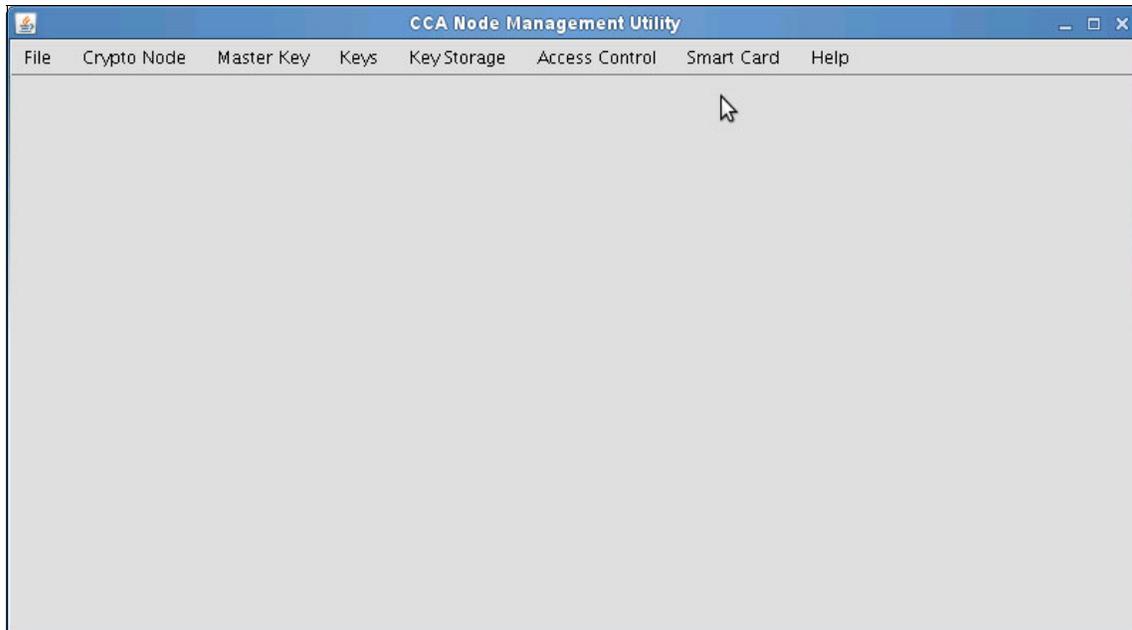


Figure B-114 CNM main window

7. After you use the CNM Utility, log off from the group profile (or stop the CNM program) by clicking **File** → **Logoff**, as shown in Figure B-115.

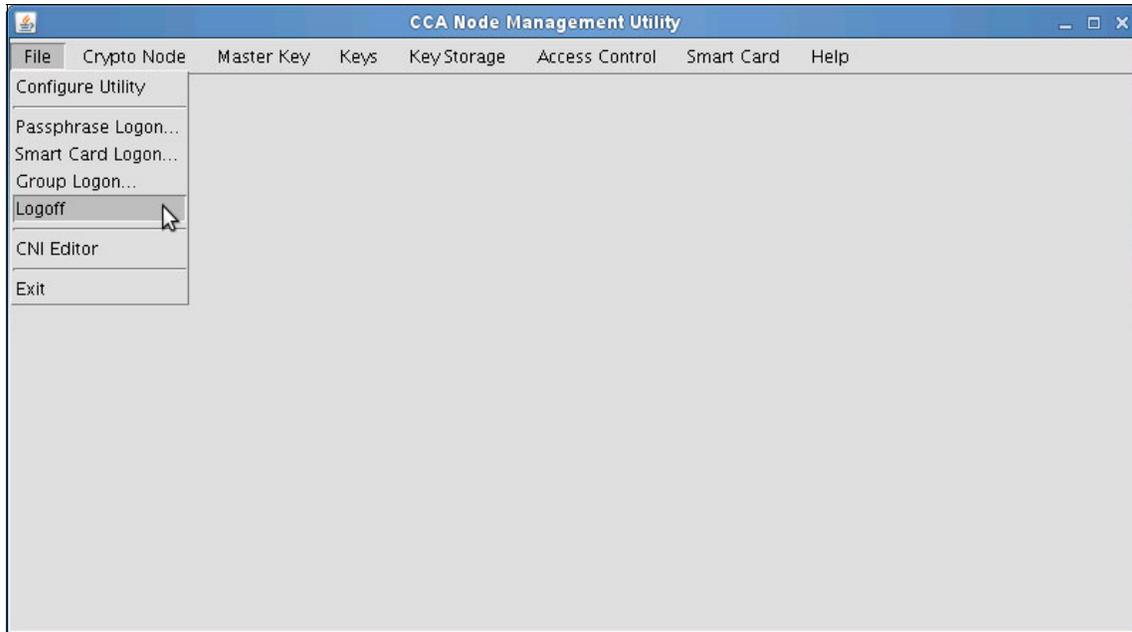


Figure B-115 CNM logoff

8. Click **Yes** when you are prompted to confirm the logoff. Click **OK**.

## Performing a CNM Utility logon by using a smart card

The aim of this procedure is to log on to the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using a logon smart card.

### Participant

Table B-19 lists the participant with its role and a brief description.

Table B-19 Participant

| Role indication | Role description                                                    |
|-----------------|---------------------------------------------------------------------|
| SO1n (or SO2n)  | The holder of logon smart card belonging to the SO1 (or SO2) group. |

## Special requirements

The participant that is listed in Table B-19 on page 342 need the following components. SO1n (or SO2n) must have the following components:

- ▶ An envelope with the SO1n logon smart card (SO2n: The SO2n logon smart card)
- ▶ An envelope with the PIN Form for the logon smart card

## Procedure: Performing a CNM Utility logon by using a smart card

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).
2. Click **File** → **Group Logon...**, as shown in Figure B-116.

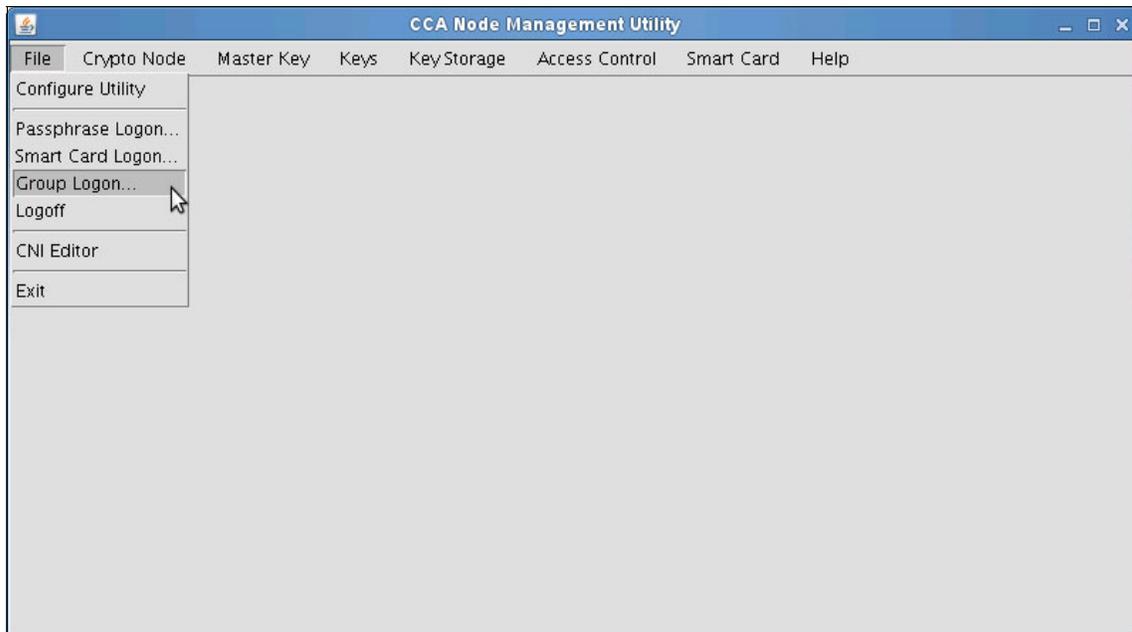


Figure B-116 Group Logon

3. SO1n is prompted to select the group of smart card profiles to log on to. Enter SO1 as the Group ID. Click **OK**, as shown in Figure B-117.



Figure B-117 Log on with Group ID

4. The Smart Card Group Logon window shows the logon smart cards that can participate in this logon.

SO1n is prompted to select the SO1n logon smart card. SO1n then clicks **Read Smart Card**, as shown in Figure B-118.

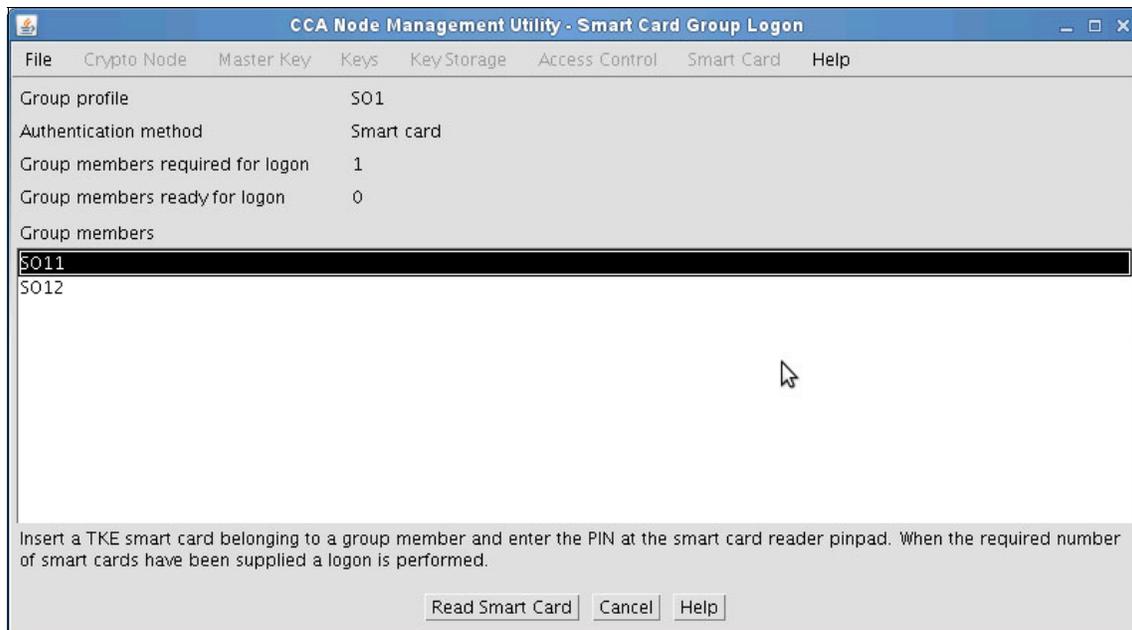


Figure B-118 Perform a Group Logon with a smart card

5. SO1n is prompted to insert the SO1n logon smart card in to reader 2. Click **OK**, as shown in Figure B-119 on page 345.



Figure B-119 Insert a smart card

6. SO1n is prompted to enter the 6-digit PIN of the SO1n smart card in reader 2, as shown in Figure B-120.



Figure B-120 Enter PIN

7. A prompt indicates that the group logon was successful. Click **OK**, as shown in Figure B-121.

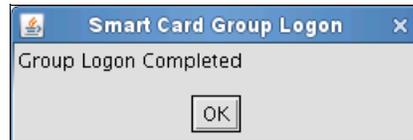


Figure B-121 Logon successful

8. After you use the CNM Utility, log off from the group profile (or stop the CNM program) by clicking **File** → **Logoff**, as shown in Figure B-122.

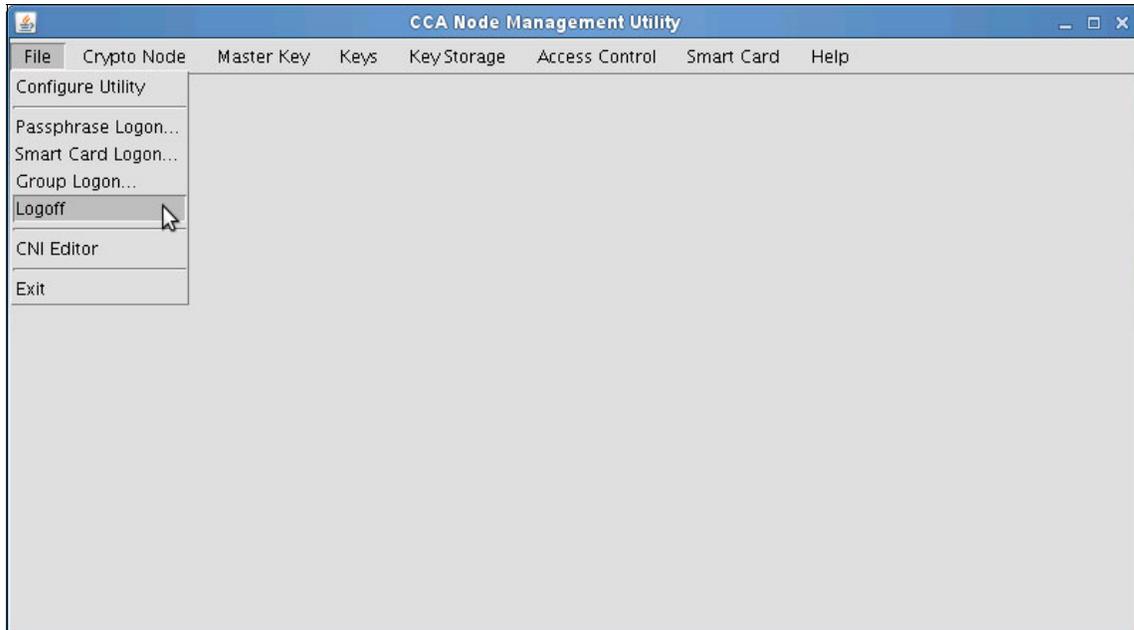


Figure B-122 Log off when finished

9. The user is prompted to confirm the logoff by clicking **Yes**.

## Performing a CNM Utility group logon by using smart cards

The aim of this procedure is to perform a group logon to the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

### Participants

Table B-20 lists the participants with their roles and a brief description.

Table B-20 Participants

| Role indication | Role description                                                      |
|-----------------|-----------------------------------------------------------------------|
| ADM1n (or SO1n) | The holder of a logon smart card belonging to the ADM1 (or SO1) group |
| ADM2n (or SO2n) | The holder of a logon smart card belonging to the ADM2 (or SO2) group |

## Special requirements

The participants that are listed in Table B-20 on page 346 need the following components:

- ▶ ADM1 $n$  must have the following components:
  - An envelope with an ADM1 $n$  logon smart card (SO1 $n$ : The SO1 $n$  logon smart card)
  - An envelope with a PIN Form for the logon smart card
- ▶ ADM2 $n$  must have the following components:
  - An envelope with an ADM2 $n$  logon smart card (SO2 $n$ : The SO2 $n$  logon smart card)
  - An envelope with a PIN Form for the logon smart card

## Procedure: Performing a CNM Utility group logon by using smart cards

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start the CCA Node Management Utility program (**IBM 4765 CNM**).
2. Click **File** → **Group Logon...**, as shown in Figure B-123.

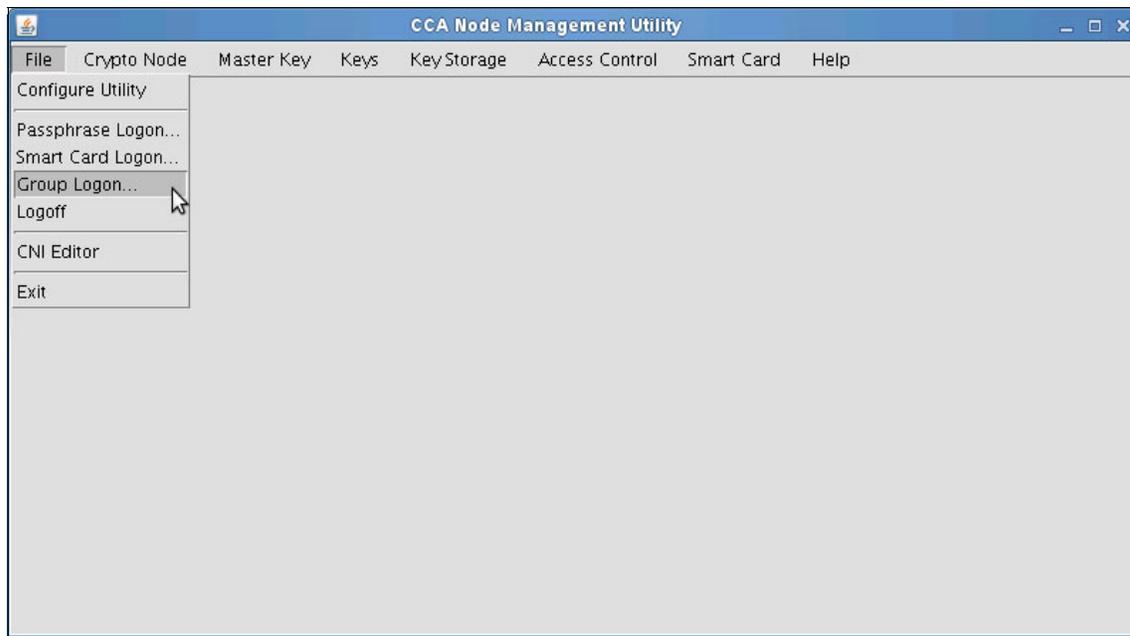


Figure B-123 Group Logon

3. ADM1n is prompted to select the group (of smart card groups) profile to log on to. Enter ADMIN as the Group ID (for SO1n, enter MANAGER). Click **OK**, as shown in Figure B-124.



Figure B-124 Enter the Group ID

4. The Smart Card Group Logon window shows the groups (of smart card groups, with associated smart cards) that can participate in this logon.  
ADM1n is prompted to select the ADM1n logon smart card from the ADM1 group of logon smart cards. ADM1n then clicks **Read Smart Card**, as shown in Figure B-125.

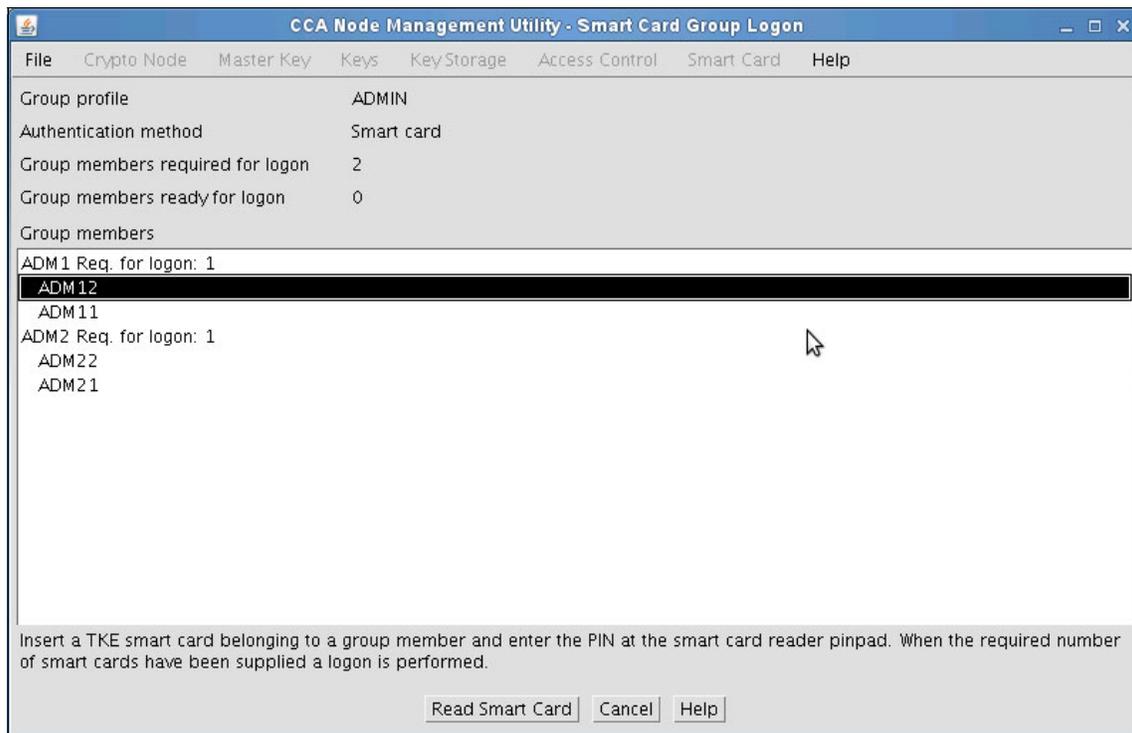


Figure B-125 Smart Card Group Logon for ADM1n

5. ADM1n is prompted to insert the ADM1n logon smart card in to reader 2. Click **OK** to continue.

6. ADM1n is then prompted to enter the 6-digit PIN of the ADM1n smart card in reader 2.
7. ADM1n must remove the ADM1n logon smart card from reader 2.  
ADM2n (or SO2n) must now *repeat* steps 5 on page 348 to 7, and in the process select an ADM2n logon smart card that belongs to the ADM2 group of logon smart cards, as shown in Figure B-126.

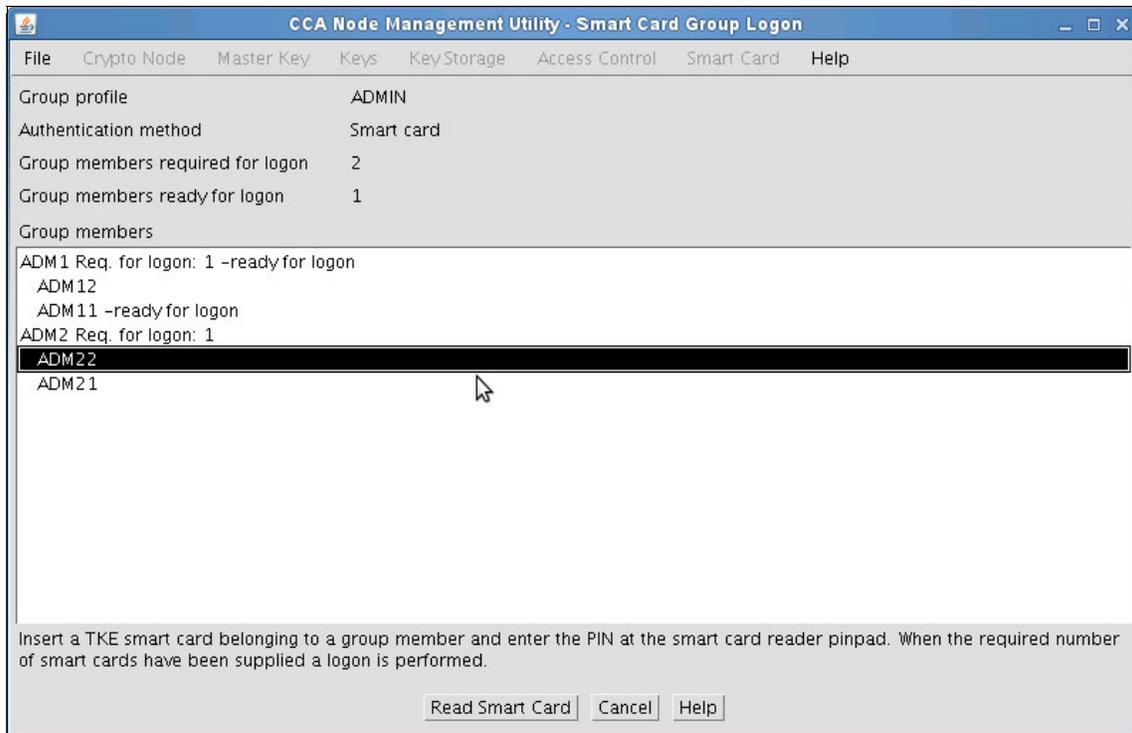


Figure B-126 Smart Card Group Logon for ADM2n

8. After ADM2n is authenticated, a prompt indicates that the group logon is successful. Click **OK**, as shown in Figure B-127.



Figure B-127 Group Logon completed

9. After you use the CNM Utility, log off from the group profile (or stop the CNM Utility program) by clicking **File** → **Logoff**.

10. When the user is prompted to confirm the logoff, click **Yes**.

## Using the CNM Utility to create, edit, or delete a role

The aim of this procedure is to create, edit, or delete a role in the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

### Participants and special requirements

First, perform the procedure in “Performing a CNM Utility logon by using a split passphrase” on page 339. If you already have smart card profiles set up for administrative purposes, perform the procedure in “Performing a CNM Utility group logon by using smart cards” on page 346 instead.

### Procedure: Using the CNM Utility to create, edit, or delete a role

Complete the following steps:

1. Click **Access Control** → **Roles**, as shown in Figure B-128.

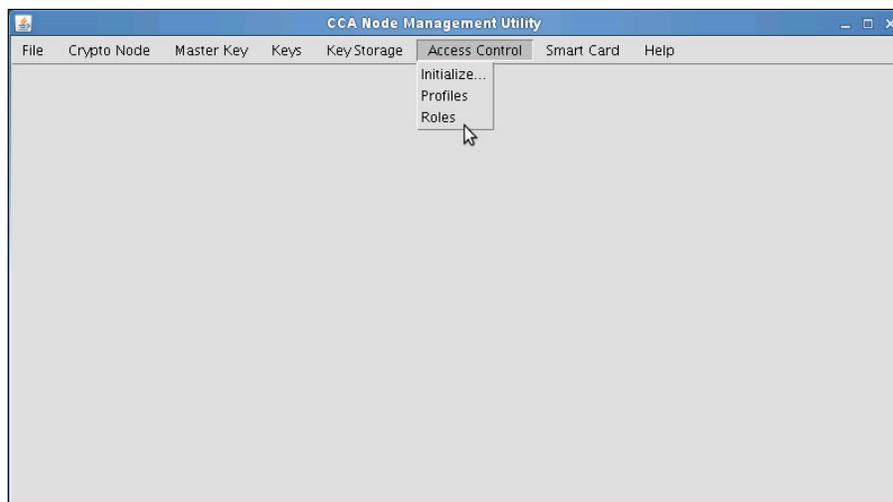


Figure B-128 Select Roles

2. Click **New** to create a role, as shown in Figure B-129 on page 351. You can also highlight a role in the list and click **Edit** if you must work on an existing role. All the following steps are the same as though you are creating a role.

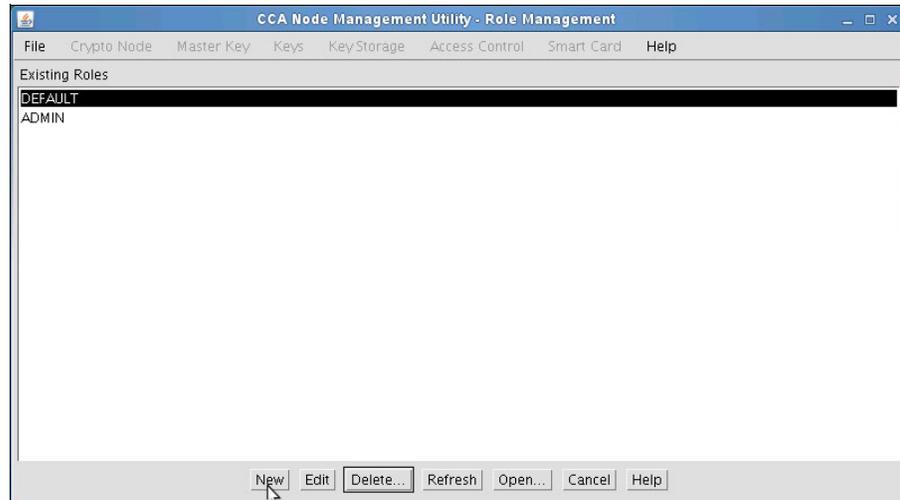


Figure B-129 Create a default role

3. Click **Open** to select an IBM provided file with access points for the role you are going to create (the files are on the DVD in the `ibm4765/default_roles` directory), as shown in Figure B-130.

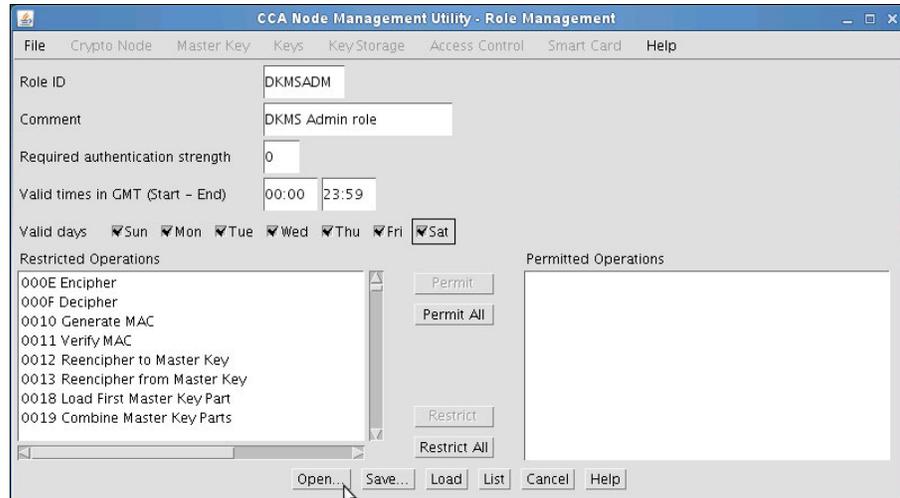


Figure B-130 Open a default role file

4. Using the Permit and Restrict buttons, you can change the list of access control points in the role, if needed.

5. Click **Load** to load the role in the IBM PCIe 4765 Cryptographic Coprocessor, as shown in Figure B-131.

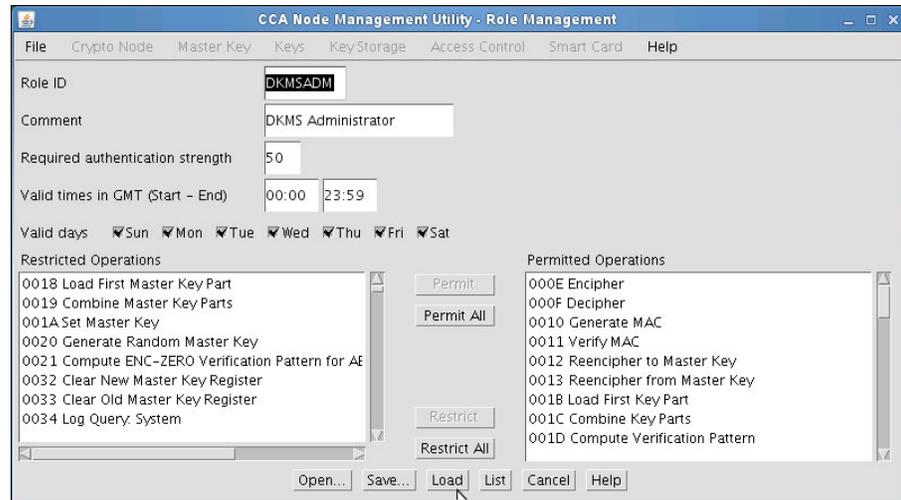


Figure B-131 Load the IBM PCIe 4765 Cryptographic Coprocessor role

6. Select **OK**.
7. Click **List** to return to the list of roles.
8. Click **Cancel** to return to the CNM Utility main window.

## Using the CNM Utility to create a smart card profile

The aim of this procedure is to create a smart card profile in the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

### Participants and special requirements

First, perform the procedure in “Performing a CNM Utility logon by using a split passphrase” on page 339. If you already have smart card profiles set up for administrative purposes, perform the procedure in “Performing a CNM Utility group logon by using smart cards” on page 346 instead.

You must have the smart card for which you want to create a profile. The PIN for the smart card is not needed.

## Procedure: Using the CNM Utility to create a smart card profile

Complete the following steps:

1. Click **Access Control** → **Profiles**, as shown in Figure B-132.

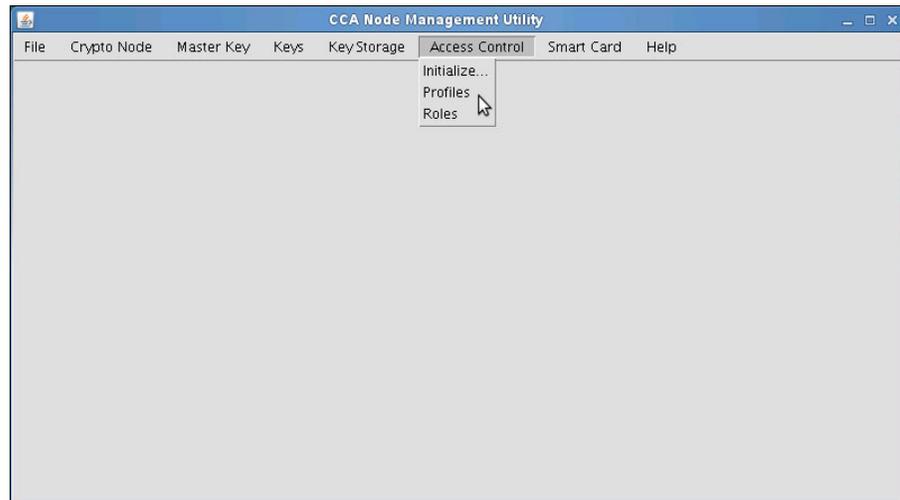


Figure B-132 Select Profiles

2. Click **New** to create a profile, as shown in Figure B-133.

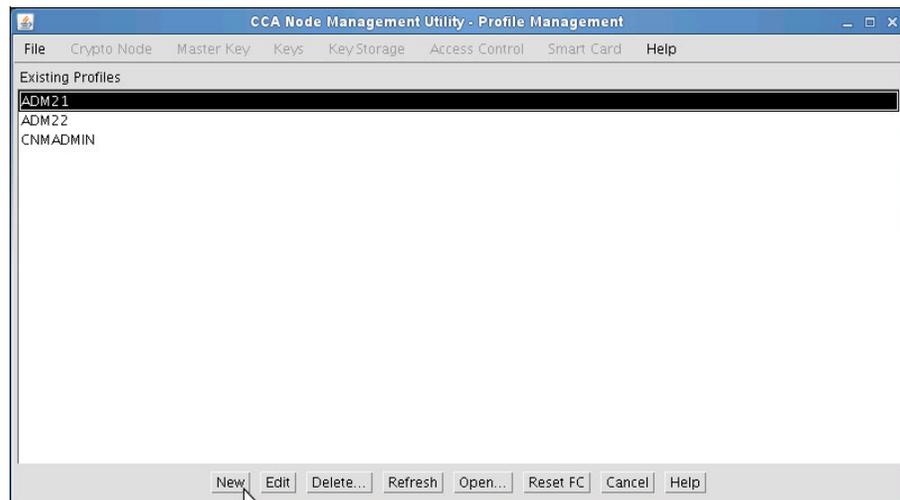


Figure B-133 Create a profile

3. Select **Smart card** and click **Continue** to add a smart card profile, as shown in Figure B-134.



Figure B-134 Select Smart card

4. Insert the smart card for the user whose profile you are creating.
5. Click **Load** to load the role in the IBM PCIe 4765 Cryptographic Coprocessor, as shown in Figure B-135.

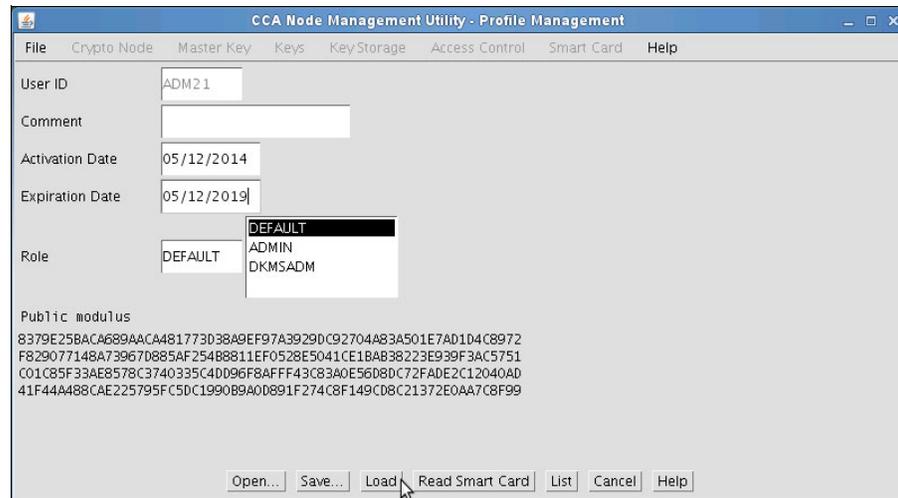


Figure B-135 Load the IBM PCIe 4765 Cryptographic Coprocessor role

6. Click **Load** and the profile is loaded.
7. Click **OK**.
8. Click **Cancel** to return to the CNM main window.

## Using the CNM Utility to create a smart card group profile

The aim of this procedure is to create a smart card group profile in the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

## Participants and special requirements

First, perform the procedure in “Performing a CNM Utility logon by using a split passphrase” on page 339. If you already have smart card profiles set up for administrative purposes, perform the procedure in “Performing a CNM Utility group logon by using smart cards” on page 346 instead.

## Procedure: Using the CNM Utility to create a smart card group profile

Complete the following steps:

1. Click **Access Control** → **Profiles**, as shown in “Using the CNM Utility to create a smart card profile” on page 352.
2. Click **New**.
3. Select **Group** and click **Continue**.
4. Enter the group name in to User ID, as shown in Figure B-136.

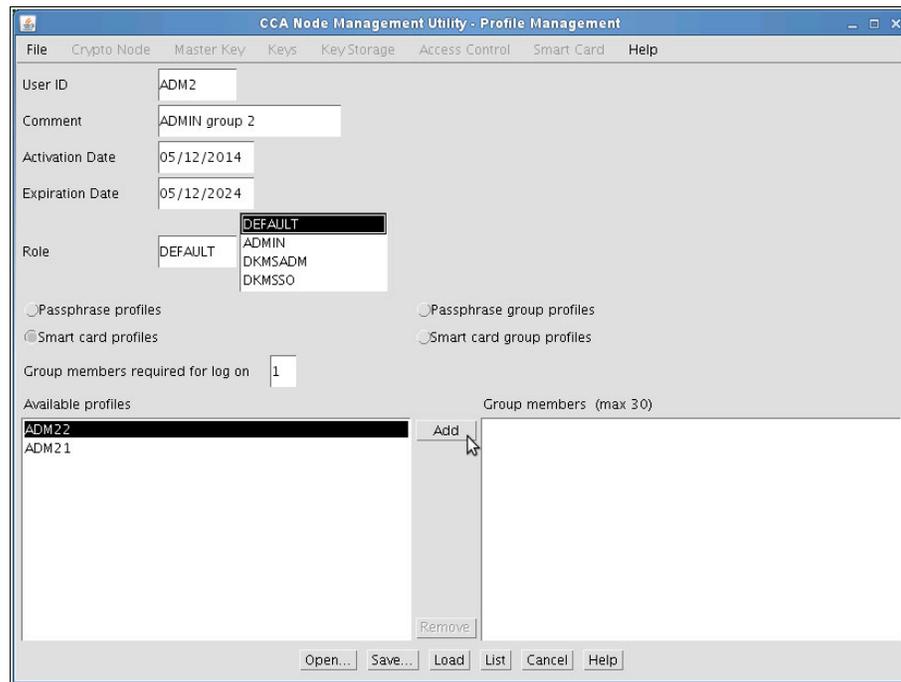


Figure B-136 Create group profile

5. Optionally, provide a Comment.
6. Change the dates as needed.

7. Select the **DEFAULT** role.
8. Click **Smart card profiles**.
9. Enter the number of group members to log on. Set this to 1 if you are using a group of group concept; otherwise, set it to at least 2.
10. Select the profiles that you want to add to the group (for the ADM2 group, include the ADM2\* profiles).
11. Click **Add**.
12. Click **Load**.
13. Click **OK**.

## Using the CNM Utility to create a group of groups profile

The aim of this procedure is to create a group profile consisting of other groups in the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

### Participants and special requirements

First, perform the procedure in “Performing a CNM Utility logon by using a split passphrase” on page 339. If you already have smart card profiles set up for administrative purposes, perform the procedure in “Performing a CNM Utility group logon by using smart cards” on page 346 instead.

### Procedure: Using the CNM Utility to create a group of groups profile

Complete the following steps:

1. Click **Access Control** → **Profiles**, as shown in “Using the CNM Utility to create a smart card profile” on page 352.
2. Click **New**.
3. Select **Group** and click **Continue**.
4. Enter the group name (for example, ADMIN) in to User ID, as shown in Figure B-137 on page 357.

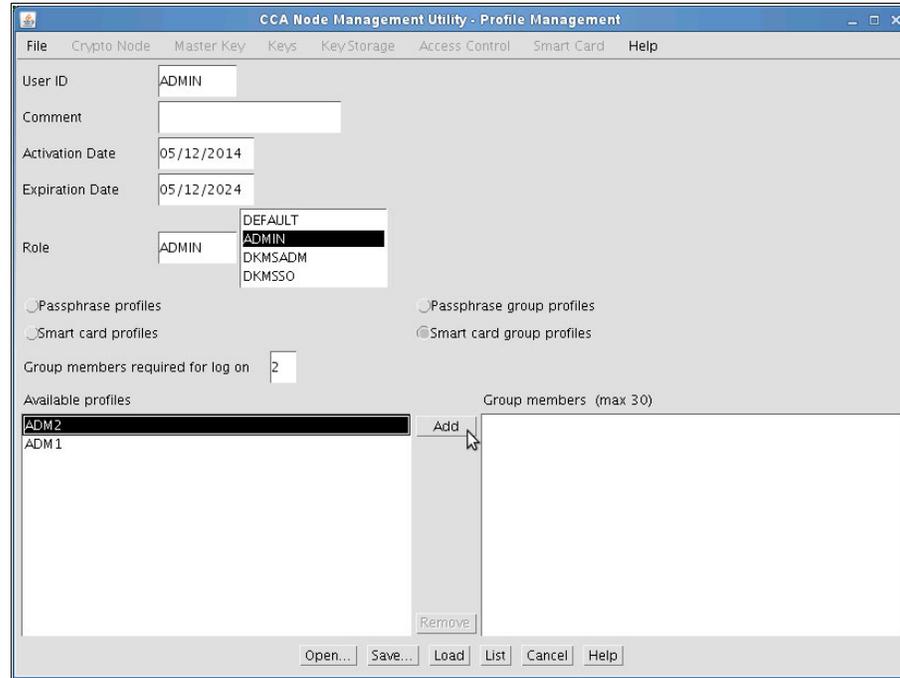


Figure B-137 Create a group of groups profile

5. Optionally, provide a Comment.
6. Change the dates as needed.
7. Select a role according to your role concept.
8. Click **Smart card group profiles**.
9. Set the number of group members to log on to 2.
10. Highlight the profiles that you want to add to the group (for the ADMIN group, include the ADM1 and ADM2 profiles).
11. Click **Add**.
12. Click **Load**.
13. Click **OK**.

## Using the CNM Utility to restrict the DEFAULT role

The aim of this procedure is to complete a secure setup by restricting the access points of the DEFAULT role in the IBM PCIe 4765 Cryptographic Coprocessor from the CNM Utility program by using logon smart cards.

## Participants and special requirements

First, perform the procedure that is outlined in “Performing a CNM Utility group logon by using smart cards” on page 346.

### Procedure: Using the CNM Utility to restrict the DEFAULT role

Complete the following steps:

1. Click **Access Control** → **Roles**, as shown in “Using the CNM Utility to create a smart card profile” on page 352.
2. Select **DEFAULT** in the list and click **Edit**, as shown in Figure B-138.

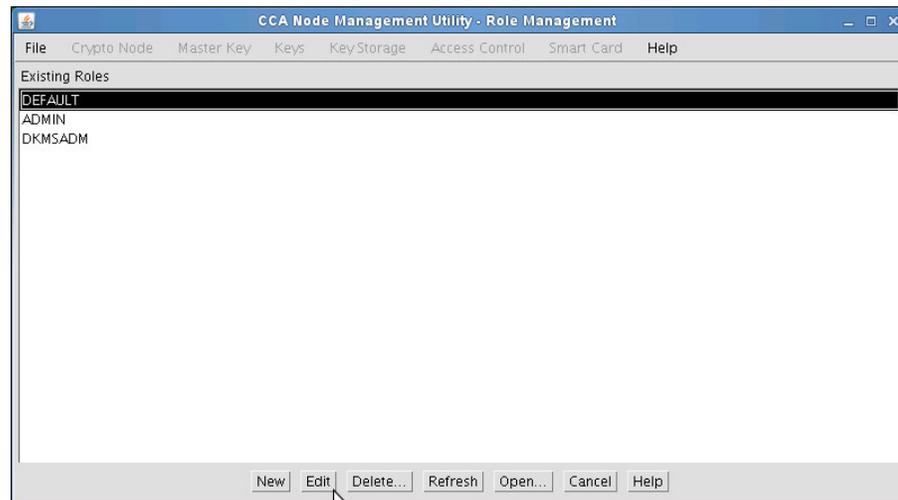


Figure B-138 Restrict the DEFAULT role

3. Mark all permitted operations except the four operations that are shown in Figure B-139 on page 359 and click **Restrict**. Then, click **Load** to change the DEFAULT role. Click **OK** to finalize this procedure.

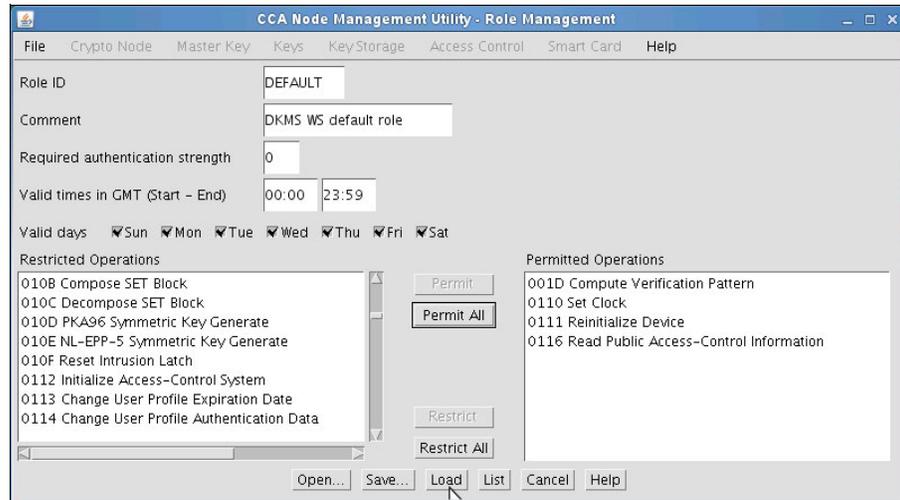


Figure B-139 Restrict the role operations

## Managing the application

This section describes how to manage the Key Management application (DKMS) application.

### Application group logon

The aim of this procedure is to log on to the DKMS application through a group of smart card group profiles.

Each of the two persons from either the Administrator Groups or the Security Officer Groups needs the following components:

- ▶ The labeled envelope with the ADM(n) smart card
- ▶ The envelope containing the PIN Form for the ADM(n) smart card

## Procedure: Application group logon

Complete the following steps:

1. Click **Computer** → **Applications** and locate and start DKMS, as shown in Figure B-140.

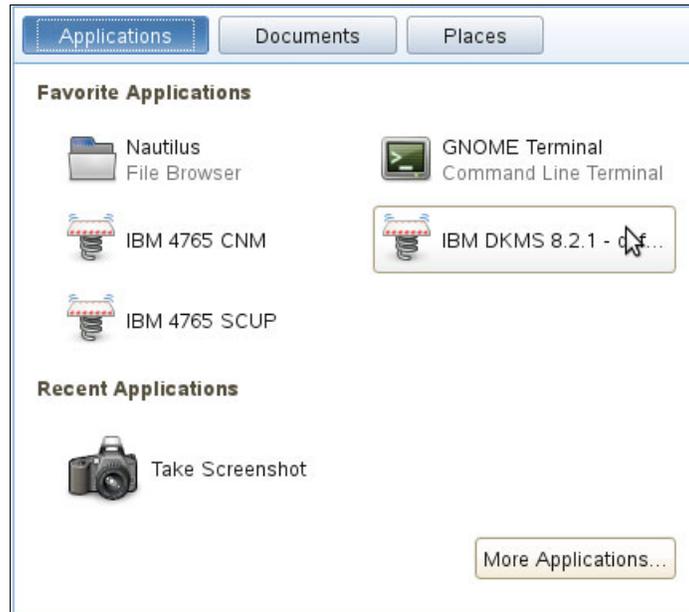


Figure B-140 Start DKMS

2. After a few seconds, the DKMS application start window opens, as shown in Figure B-141. Click **Start** to begin the logon sequence.

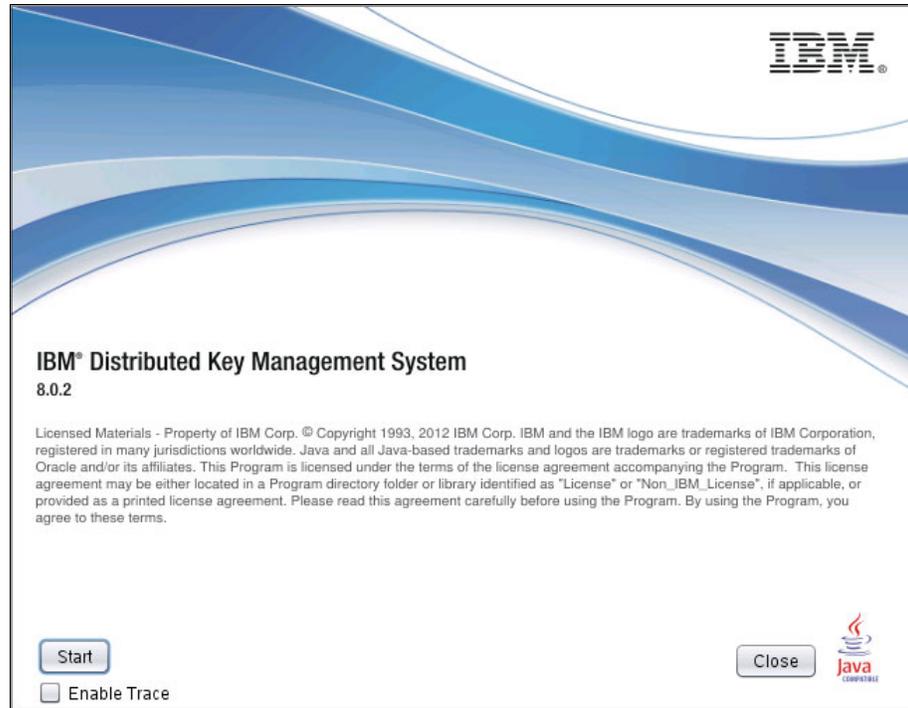


Figure B-141 The DKMS application start window

3. The IBM 4765 Logon window opens. Select a proper Profile ID:
  - Administrators: ADMIN2
  - Security Officers: DKMSSOClick **OK** to continue, as shown in Figure B-142.

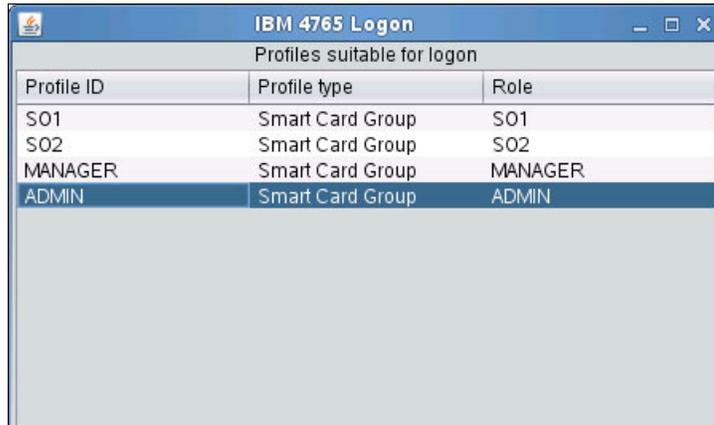


Figure B-142 IBM 4765 Logon window

4. The IBM 4765 Group Logon window opens. Select one of the Profile IDs corresponding to a member in Group 1 in the list of Group members. Click **OK**, as shown in Figure B-143 on page 363.

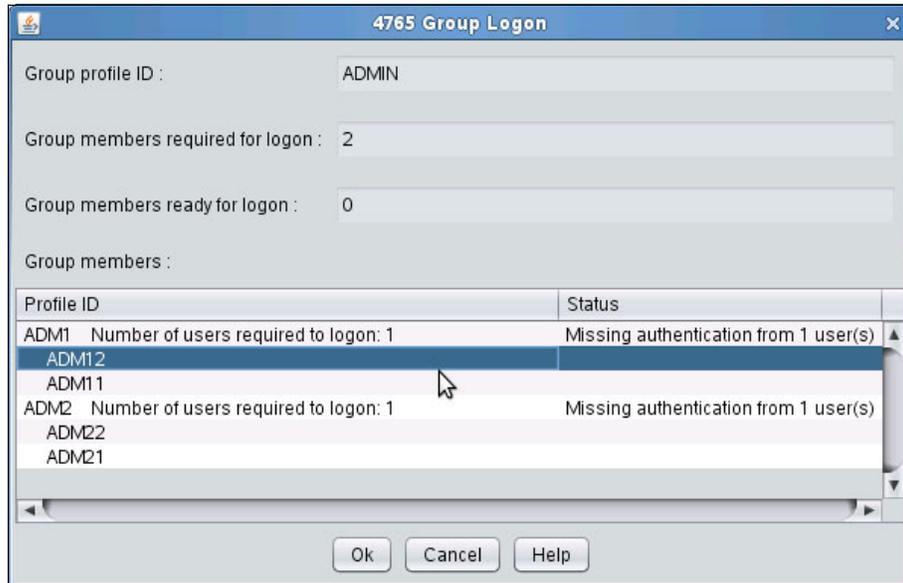


Figure B-143 Group Logon

- The Group 1 member that is responsible for the selected Profile ID is prompted to insert the corresponding ADM(n) / SO(n) smart card in to reader 1. Click **OK**, as shown in Figure B-144.



Figure B-144 Insert the smart card

- The same Group 1 member is prompted to enter the 6-digit PIN in to reader 1, as shown in Figure B-145.



Figure B-145 Enter PIN

- The Group 1 member must now remove the ADM(n) / SO(n) smart card from reader 1.

A member of *Group 2* must repeat steps 4 on page 362 to 7, as shown in Figure B-146.

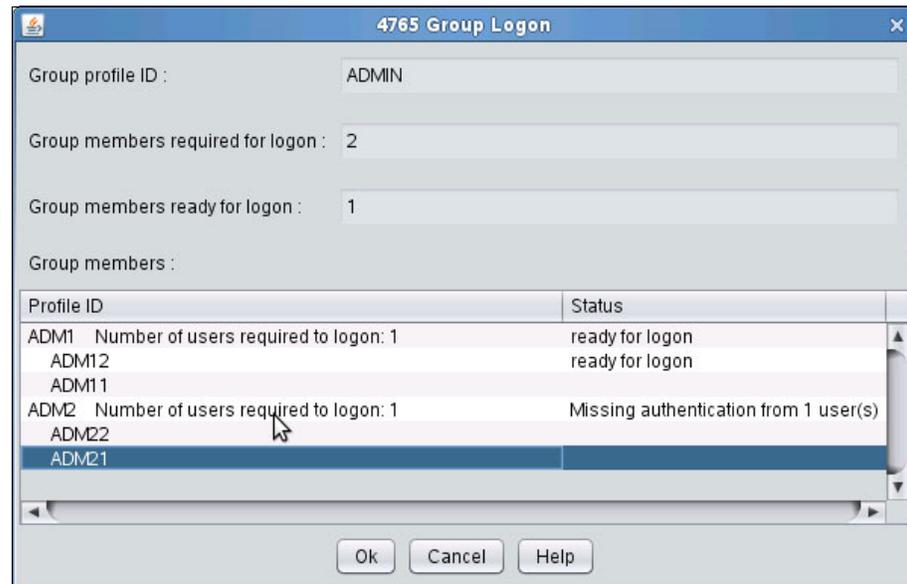


Figure B-146 Group Logon for a member of Group 2

- The first time that you start the application, you are prompted to specify the TCP/IP communication settings of the primary host Agent, as shown in Figure B-147 on page 365. This happens after a successful logon is authorized by the IBM 4765 access control system. If you keep **Prompt for communication parameters during start** checked, you are prompted for the parameters the next time you start the DKMS application.

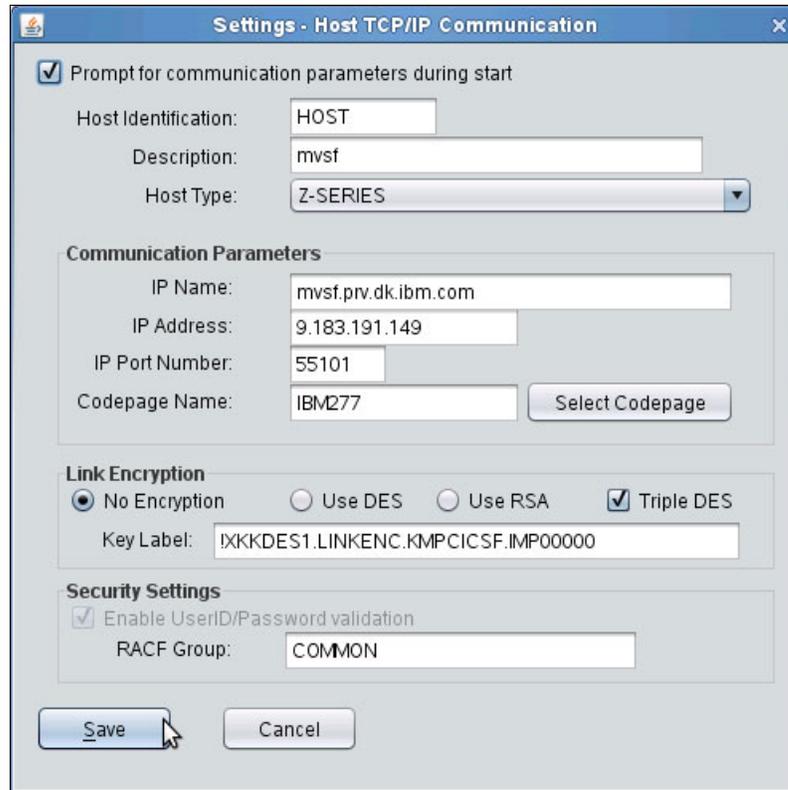


Figure B-147 TCP/IP communication

This setting can be useful because you most likely will need to change some of these settings during the first number of attempts to start the application. After all the settings are verified to be working correctly, you can clear the check mark to avoid further prompts of this window.

9. A series of prompts indicate the successful verification of the various configuration tables, as shown in Figure B-148.



Figure B-148 SignOn to Host verification

10. Finally, the DKMS Menu opens and indicates that the logon process is finished, as shown in Figure B-149.

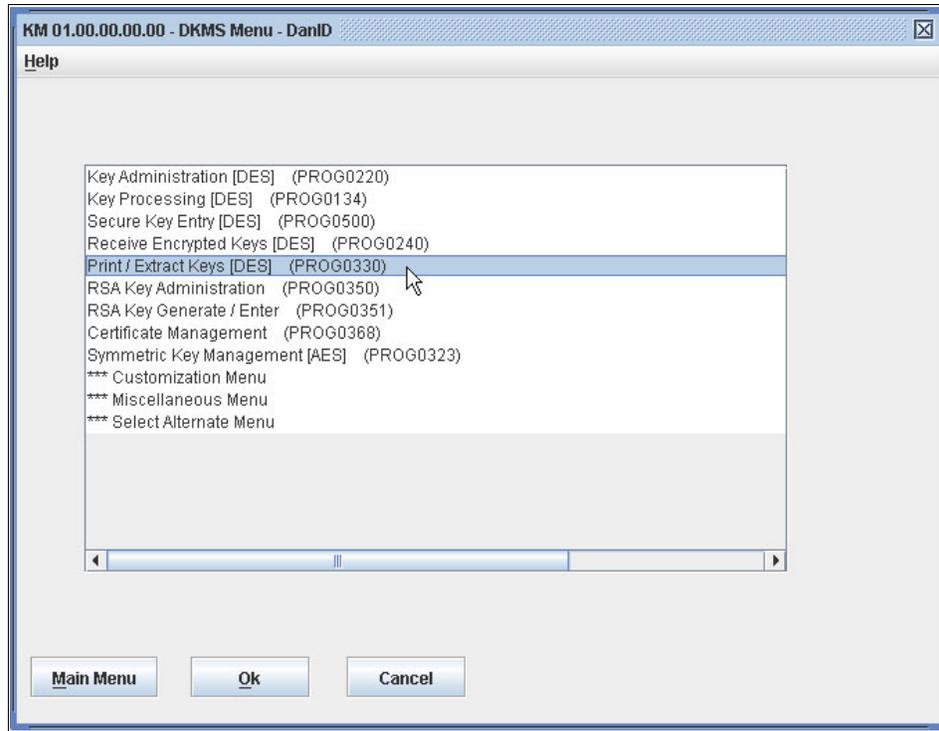


Figure B-149 Successful logon - DKMS Menu



# Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

## IBM Redbooks

The following IBM Redbooks publication provides additional information about the topics in this document. The publication that is referenced in this list might be available in softcopy only.

- ▶ *System z Crypto and TKE Update*, SG24-7848

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

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

## Product publications

The product manuals are available on the product CDs. This book mentions the following subset of manuals:

- ▶ *IBM Distributed Key Management System User's Guide, Volume 1, Customization and basic DES key management*, DKMS-2200
- ▶ *IBM Distributed Key Management System User's Guide — Key Templates and Symmetric Key Management*, DKMS-2231
- ▶ *IBM DKMS Key Management Workstation Installation Guide for SLES 11*, DKMS-4050

## Online resources

These websites are also relevant as further information sources:

- ▶ For more information about the IBM Enterprise Key Management Foundation, go to the website for the IBM Crypto Competence Center, Copenhagen:

<http://www.ibm.com/dk/security/cccc/>

- ▶ To find IBM System z Service Offerings for the IBM Enterprise Key Management Foundation, go to the following website:

[http://www.ibm.com/systems/services/labservices/platforms/labservices\\_z.html](http://www.ibm.com/systems/services/labservices/platforms/labservices_z.html)

## Help from IBM

IBM Support and downloads

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

IBM Global Services

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



## Key Management Deployment Guide: Using the IBM Enterprise Key Management Foundation

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages







# Key Management Deployment Guide

Using the IBM Enterprise Key Management Foundation



**Enterprise  
integration for  
centralized key  
management  
deployment**

**Complete  
information about  
architecture and  
components**

**Deployment scenario  
with hands-on  
details**

In an increasingly interconnected world, data breaches grab headlines. The security of sensitive information is vital, and new requirements and regulatory bodies such as the Payment Card Industry Data Security Standard (PCI-DSS), Health Insurance Portability and Accountability Act (HIPAA), and Sarbanes-Oxley (SOX) create challenges for enterprises that use encryption to protect their information. As encryption becomes more widely adopted, organizations also must contend with an ever-growing set of encryption keys. Effective management of these keys is essential to ensure both the availability and security of the encrypted information. Centralized management of keys and certificates is necessary to perform the complex tasks that are related to key and certificate generation, renewal, and backup and recovery.

The IBM Enterprise Key Management Foundation (EKMF) is a flexible and highly secure key management system for the enterprise. It provides centralized key management on IBM zEnterprise and distributed platforms for streamlined, efficient, and secure key and certificate management operations.

This IBM Redbooks publication introduces key concepts around a centralized key management infrastructure and depicts the proper planning, implementation, and management of such a system using the IBM Enterprise Key Management Foundation solution.

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