



# Key Recovery Server

## Installation and Usage Guide



Copyright© 1997 International Business Machines Corporation. All rights reserved.  
Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication, or disclosure is subject to restriction set forth in GSA ADP Schedule Contract with IBM Corp.  
IBM is a registered trademark of International Business Machines Corporation, Armonk, N.Y.

# Table of Contents

<b>CHAPTER 1. INTRODUCTION</b> .....	<b>1</b>
1.1 AUDIENCE .....	1
1.2 DOCUMENTATION SET .....	1
1.3 REFERENCES .....	2
<b>CHAPTER 2. TECHNOLOGY OVERVIEW</b> .....	<b>3</b>
<b>CHAPTER 3. KEY RECOVERY SCENARIOS</b> .....	<b>4</b>
<b>CHAPTER 4. KEY RECOVERY PHASES</b> .....	<b>6</b>
4.1 KEY RECOVERY CAPABILITY DEFINITION PHASE .....	6
4.2 KEY RECOVERY ENABLEMENT PHASE .....	6
4.3 KEY RECOVERY REQUEST PHASE .....	7
<b>CHAPTER 5. INSTALLATION AND CONFIGURATION</b> .....	<b>8</b>
5.1 KEY RECOVERY SERVER COMPONENTS .....	8
5.2 CHOOSING A CONFIGURATION .....	8
5.3 SETTING UP SECURE FACILITIES .....	8
5.4 INSTALLING THE SOFTWARE .....	9
5.5 OBTAINING CERTIFICATES AND PRIVATE KEYS .....	9
5.6 MASTER CONFIGURATION FILE .....	9
5.7 CONFIGURING A KEY RECOVERY OFFICER CLIENT .....	10
5.8 CONFIGURING A KEY RECOVERY COORDINATOR SERVER .....	10
5.9 CONFIGURING A KEY RECOVERY AGENT SERVER .....	11
<b>CHAPTER 6. OPERATION</b> .....	<b>12</b>
6.1 KEY RECOVERY OFFICER .....	12
6.1.1 <i>Sending an Enterprise Key Recovery Request</i> .....	12
6.1.2 <i>Sending a Law Enforcement Key Recovery Request</i> .....	14
6.1.3 <i>Monitoring the Status of Requests</i> .....	15
6.1.4 <i>Optional Decryption of Data</i> .....	16
6.2 KEY RECOVERY COORDINATOR .....	16
6.2.1 <i>Starting the Server</i> .....	16
6.2.2 <i>Monitoring the Status of Requests</i> .....	16
6.2.3 <i>Viewing the Audit Trail</i> .....	17
6.3 KEY RECOVERY AGENT .....	17
6.3.1 <i>Starting the Server</i> .....	18
6.3.2 <i>Monitoring the Status of Requests</i> .....	18
<b>APPENDIX A. PRE-INSTALLATION SETUP</b> .....	<b>19</b>
A.1 FACILITY REQUIREMENTS .....	19
A.1.1 <i>Key Recovery Server Facility Room</i> .....	19
A.1.2 <i>Limited Personnel Access</i> .....	19
A.2 SERVER PHYSICAL REQUIREMENTS .....	19
A.2.1 <i>Keyed Lock for System</i> .....	19
A.2.2 <i>Dedicated System</i> .....	19
A.2.3 <i>Removable Backup Media Connectivity</i> .....	19
A.3 INSTALLATION .....	20
<b>APPENDIX C. GLOSSARY</b> .....	<b>22</b>

## List Of Figures

Figure 1. KRO Password .....	12
Figure 2. Key Recovery Officer Request Monitor .....	13
Figure 3. KRB Browse Directory .....	13
Figure 4. Key Recovery Law Enforcement Request.....	14
Figure 5. Load Certificate Screen.....	15
Figure 6. Certificate Directory .....	15
Figure 7. Key Recovery Coordinator Request Monitor .....	17
Figure 8. Key Recovery Agent Name.....	18
Figure 9. Key Recovery Agent Request Monitor .....	18

## List Of Tables

Table 1. Comparison of Key Recovery Scenarios .....	4
---	---

# Chapter 1. Introduction

The IBM key recovery product suite consists of the IBM Secure Cryptography and Certificate Services (SCCS) Toolkit, the IBM Key Recovery Service Provider (KRSP), and the IBM Key Recovery Server (KRS). Components in IBM SCCS Toolkit and the IBM KRSP encapsulate keys such that only authorized parties using the IBM Key Recovery Server can recover the keys. This document describes the installation and usage of the IBM Key Recovery Server.

## 1.1 Audience

This document is intended to provide users and security administrators with an understanding of key recovery concepts, guidance in setting up a key recovery solution for an organization, and procedures for installing, configuring, and operating the IBM Key Recovery Server.

Security administrators are assumed to have a working knowledge of public key infrastructures and network computing environments.

This document is intended for use by:

- Enterprise Employees and Security Personnel
- Policy Enforcement Personnel
- Key Recovery Facilities Security Personnel

## 1.2 Documentation Set

The documentation set for the IBM Key Recovery Server and related products consists of the following manuals.

- *Secure Cryptography and Certificate Services Toolkit Developer's Guide*  
Document filename: sccs\_dev.pdf  
This document presents an overview of the Secure Cryptography and Certificate Services (SCCS) Toolkit. It explains how to integrate SCCS into applications and contains a sample SCCS application.
- *Secure Cryptography and Certificate Services Toolkit Application Programming Interface Specification*  
Document filename: sccs\_api.pdf  
This document defines the interface that applications developers employ to access security services provided by the SCCS Framework and service provider modules.
- *Secure Cryptography and Certificate Services Toolkit Service Provider Module Structure & Administration*  
Document filename: sccs\_mod.pdf  
This document describes the features common to all SCCS service provider modules. It should be used in conjunction with the individual SCCS Service Provider Interface Specifications in order to build a service provider module.
- *Secure Cryptography and Certificate Services Toolkit Cryptographic Service Provider Interface Specification*  
Document filename: sccs\_spi.pdf  
This document defines the interface to which cryptography service provider modules must conform in order to be accessible through SCCS.
- *Key Recovery Service Provider Interface Specification*  
Document filename: kr\_spi.pdf  
This document defines the interface to which key recovery service provider modules must conform in order to be accessible through SCCS.

- *Key Recovery Server Installation and Usage Guide*  
Document filename: krs\_gd.pdf  
This document describes how to install and use key recovery solutions using the components in the IBM Key Recovery Server.
- *Secure Cryptography and Certificate Services Toolkit Trust Policy Interface Specification*  
Document filename: sccs\_tp\_spi.pdf  
This document defines the interface to which policy makers, such as Certificate Authorities (CAs), Certificate Issuers, and policy-making application developers, must conform in order to extend SCCS with model or application specific policies.
- *Secure Cryptography and Certificate Services Toolkit Certificate Library Interface Specification*  
Document filename: sccs\_cl\_spi.pdf  
This document defines the interface to which certificate library developers must conform to provide format-specific certificate manipulation services to numerous SCCS applications and trust policy modules.
- *Secure Cryptography and Certificate Services Toolkit Data Storage Library Interface Specification*  
Document filename: sccs\_dl\_spi.pdf.  
This manual defines the interface to which library developers must conform to provide format-specific or format-independent persistent storage of certificates.

### 1.3 References

BSAFE*	<i>BSAFE Cryptography Toolkit</i> , RSA Data Security, Inc., Redwood City, CA
PKCS*	<i>The Public-Key Cryptography Standards</i> , RSA Laboratories, Redwood City, CA: RSA Data Security, Inc.
X.509	<i>CCITT. Recommendation X.509: The Directory – Authentication Framework</i> . 1988. CCITT stands for Comite Consultatif Internationale Telegraphique et Telphonique (International Telegraph and Telephone Consultative Committee)
Cryptography	<i>Applied Cryptography, Second Edition Protocols, Algorithms, and Source Code in C</i> , Bruce Schneier: John Wiley & Sons, Inc., 1996

## Chapter 2. Technology Overview

Because of the rise in popularity of the Internet and its huge potential for communication and commerce, the topic of security has percolated to the top of the list of major concerns of businesses, individuals, and governments. Fortunately, modern cryptographic techniques provide the foundation for security solutions. One of the most useful of these techniques, data encryption, involves encrypting information using **cryptographic keys** to keep the data confidential to parties who know the keys.

In environments where data encryption is used, these keys become critical items that are needed to access encrypted information. For example, an enterprise may heavily utilize data encryption in its mission-critical applications for confidentiality and the enterprise's employees may possess knowledge of keys relevant to their daily work.

The ability to recover the keys used for data encryption is a pivotal issue to the enterprise and potentially to policy (law) enforcement officials. Reasons for recovering encryption keys may include:

- An individual has encrypted important information that needs to be reused in clear form
- A business needs access to employee-encrypted non-personal information and the employee is not available
- Law enforcement personnel obtain a court order giving them the right to access information (e.g., a search warrant).

The ability to recover keys is provided by key recovery technologies that are in existence today. They fall under the following two categories:

- Techniques involving some form of escrow of the key or key parts with a trusted party (e.g., Royal Holloway), and
- Encapsulation techniques that involve creating key recovery fields that are mathematically related to, but not actually the key or parts of the key. These fields are associated with the encrypted data. Later, the key recovery fields can be used to recover the key.

IBM key recovery technology encapsulates keys. The IBM key encapsulation technique is based on the paradigm that a cryptographically encapsulated form of the key, a Key Recovery Field (KRF), is made available to parties that require key recovery. The technique ensures that only trusted principals, possibly third parties called *key recovery agents (KRAs)*, can perform unwrap operations on the encapsulated key block to retrieve the key material concealed inside.

When encapsulation schemes are used to conceal ephemeral versus long-term keys there is much greater privacy for the individuals; they can generate and keep their own long-term keys. Ephemeral keys can be recovered independently, so there is maximal granularity with respect to the recoverable data.

Advantages of ephemeral key encapsulation and recovery can therefore be summarized in the following:

- More privacy for individuals
- Fine granularity for recoverable keys
- No latency to obtain and use public keys
- No need for individuals to belong to any government-approved public key infrastructure (PKI)

For additional information about key encapsulation, refer to Section 4.1 in the *Secure Cryptography and Certificate Services Toolkit Developer's Guide*.

## Chapter 3. Key Recovery Scenarios

Three operational scenarios are supported by IBM key recovery technology:

- Law Enforcement Key Recovery
- Enterprise Key Recovery
- Individual Key Recovery

In the law enforcement scenario, key recovery is mandated by the jurisdictional law enforcement authorities in the interest of national security and law enforcement. For a specific cryptographic product, the key recovery policies for multiple jurisdictions may apply simultaneously. The policies (if any) of the jurisdiction(s) of manufacture of the product, as well as the jurisdiction of installation and use, need to be applied to the product such that the most restrictive combination of the multiple policies is used. Thus, law enforcement key recovery is based on mandatory key recovery policies; these policies are logically bound to the cryptographic product at the time the product is installed or shipped. Mechanisms for vendor-controlled updates of such law enforcement key recovery policies will be provided; however, organizations and end users of the product cannot modify this policy at their discretion. The Key Recovery Agents (KRAs) used for this scenario of key recovery need to be selected carefully to ensure that these agents meet the eligibility criteria for the relevant jurisdiction where the product is being used.

Enterprise key recovery allows enterprises to enforce stricter monitoring of the use of cryptography, and the recovery of enciphered data when the need arises. Enterprise key recovery is also based on a mandatory key recovery policy; however, this policy is set (through administrative means perhaps) by the organization or enterprise at the time of installation of a recovery enabled cryptographic product. The enterprise key recovery policy should not be modifiable or circumventable by the individual using the cryptographic product.

Individual key recovery is user-discretionary in nature, and is performed for the purpose of recovery of enciphered data by the owner of the data, if the cryptographic keys are lost or corrupted. Since this is a non-mandatory key recovery scenario, it is not based on any policy that is enforced by the cryptographic product; rather, the product may allow the user to specify when individual key recovery enablement is to be performed. There are few restrictions on the use of specific KRAs.

In each of the above scenarios key recovery may be desired. However, the detailed aspects or characteristics of these three scenarios are somewhat varied. Table 1 summarizes the specific characteristics of the different operational scenarios.

**Table 1. Comparison of Key Recovery Scenarios**

<b>Properties</b>	<b>Law Enforcement</b>	<b>Enterprise</b>	<b>Individual</b>
Mandatory key recovery	Yes	Yes	No
Escrow or recovery agents are approved	Yes	Yes	No
Recovery enablement needs to be noncircumventable	Yes	Yes	No
Dual-sided key recovery enablement	Maybe	Maybe	No
Use of workfactor when generating KRF	Maybe	No	No
KRF contains agent identification	Yes	Maybe	Yes
User registration needed at escrow or recovery agents	Maybe	Maybe	Maybe
User authentication information needed within KRF	Maybe	Yes	Yes
End-user knowledge/cooperation required	No	No	Yes



IBM's key recovery enabled cryptographic products are designed so that the key recovery enablement operation is mandatory and noncircumventable in the law enforcement and enterprise scenarios, and discretionary for the individual scenario. The KRAs that are used for law enforcement and enterprise scenarios must be tightly controlled so that the agents are validated to belong to a set of authorized or approved agents. In the law enforcement and enterprise scenarios, the key recovery process typically needs to be performed without the knowledge and cooperation of the parties involved in the cryptographic association. Such key recovery will be enabled by the law enforcement, enterprise officials communicating with approved key recovery agents.

The components of the Key Recovery Fields (KRFs) also vary somewhat between the three scenarios. In the law enforcement scenario, the KRFs must contain identification information for the escrow or recovery agent(s) whereas for the enterprise and individual scenarios, the agent identification information is not so critical, since this information may be available from the context of the recovery enablement operation. For the individual scenario, there needs to be a strong user authentication component in the KRFs, to allow the owner of the KRFs to authenticate themselves to the agents. For the law enforcement and enterprise scenarios, the KRFs may contain recovery information for both the generator and receiver of the enciphered data.

IBM key recovery products support all three of the above scenarios for key recovery. The law enforcement and enterprise scenarios for key recovery are mandatory in nature and are enforced. On the other hand, the individual scenario for key recovery is discretionary and no policy checks on the KRA credentials are performed. The application/user requests key recovery operations at their discretion.

## Chapter 4. Key Recovery Phases

Using IBM key recovery technology, the process of cryptographic key recovery involves three major phases. First, there is a *key recovery capability definition phase* where the parties that desire key recovery perform some initialization operations with a party authorized to issue key recovery requests; these operations include obtaining a public key certificate for relevant key recovery agent(s). Next, parties that are involved in cryptographic associations have to perform operations to enable key recovery (such as the generation of key recovery fields) - this is typically called the *key recovery enablement phase*. Finally, authorized parties that desire to recover the data keys, do so with the help of a recovery server and one or more recovery agents or - this is the *key recovery request phase*.

The entities involved in key recovery include the following:

- Key Recovery Officer (KRO) – the security officer representing an enterprise controlling the enterprise domain or a subdomain. The KRO is responsible for configuring the systems that use data encryption such that the appropriate key recovery fields are generated.
- Key Recovery Coordinator (KRC) – a software component that runs as a server listening for key recovery requests from authorized entities; typically this software is run within a secure facility.
- Key Recovery Agent (KRA) – a software component that runs as a server listening for requests from the KRC requesting it to compute secondary keys for the KRC. IBM key recovery technology allows an arbitrary number of KRAs. This software also is assumed to run within a secure and separate principals facility.

While it is strongly recommended that the KRAs involved in the recovery phase be as isolated as possible to prevent collusion attacks, the IBM Key Recovery Server (KRS) product is flexible enough to allow these entities to be configured on one or several machines connected via a TCP/IP network.

### 4.1 Key Recovery Capability Definition Phase

In this phase, each KRA, the KRC, and the KRO must be issued RSA key pairs in the form of public key certificates and private keys. Each of the parties must safely store their private keys and not divulge them.

The KRO must choose the KRAs that will be involved in key recovery and get their public key certificates. These public key certificates contain critical information that will be needed to generate the Key Recovery Fields (KRFs). The KRFs will be generated on key recovery enabled systems that have the IBM Key Recovery Service Provider (KRSP) installed and configured. These public key certificates must be sent to IBM for conversion into a form that the IBM KRSP can use the resulting **sealed** configuration files must then be installed on all key recovery enabled systems.

The certificates and private keys issued to the KRO and the KRC are used to digitally secure communication between them.

### 4.2 Key Recovery Enablement Phase

During the key recovery enablement phase, the IBM KRSP generates KRFs so that a given party that is authorized can recover the key associated within the KRFs. Since multiple parties may be authorized to recover a given key, a set of KRFs may be generated for a given key. These KRFs are contained in a single data block called a Key Recovery Block (KRB). Applications that have been written to the Application Programming Interface (API) provided by the IBM SCCS Toolkit will be able to access the services of the IBM Key Recovery Service Provider.

The purpose of a KRSP is to generate a correct KRB given an ephemeral symmetric key and to make sure that KRBs have not been tampered with subsequent to generation.

For additional information about key recovery-enabled application development, refer to the *Secure Cryptography and Certificate Services Toolkit Application Programming Interface Specification*.

For additional information about KRSPs, refer to the *Key Recovery Service Provider Interface Specification*, which defines the interface to which key recovery service providers must conform in order to be accessible through SCCS.

### **4.3 Key Recovery Request Phase**

When a key is unavailable for any of the reasons previously outlined, encrypted information is effectively “locked” until the key can be recovered. The KRB, which is associated with the encrypted data, is retrieved by the KRO. In the case of an encrypted file, the KRB is typically stored in a filename derived from the original encrypted file. In the case of an encrypted network communication, the encrypted data and the KRB are both sent on the network.

The KRO, using the IBM provided software, sends the KRB as part of a key recovery request to the KRC. The KRC acts as a front-end for coordinating key recovery requests. It distinguishes between law enforcement, enterprise, or individual originated key recovery requests, authenticating the requesting party when needed, and processes the KRB by sending requests to the appropriate KRAs involved in the key recovery phase. The identities of the KRAs are stored in the KRB.

Each KRA receives a request to compute a secondary key and send it back to the KRC. When the KRC receives positive responses from each KRA, it can perform a number of decryptions to compute the actual original encryption key. This original encryption key is then sent back to the KRO, completing the recovery phase.

The recovered key can then be used to decrypt the encrypted data. Note that it is not necessary for the party recovering the key to be the party decrypting the data, but this is often the case.

## Chapter 5. Installation and Configuration

### 5.1 Key Recovery Server Components

The following software components are included in the IBM Key Recovery Server (KRS):

- Key Recovery Officer Client – this client software allows the KRO to submit key recovery requests to the KRC.
- Key Recovery Coordinator Server – this server software listens for requests from the KRO and processes them as required, including authentication of KRO requests, communicating with a set of KRAs to obtain required secondary keys, and recovering the requested key information.
- Key Recovery Agent Server – this server software listens for requests from the KRC and computes secondary keys using its configured private key.
- Key Recovery Utility – this utility software allows the user to encrypt and decrypt an arbitrary file given a key. It can also generate the corresponding KRB if the system has the IBM KRSP installed and configured.
- Authorization Info Utility – this utility software allows the KRO to create an authorization information file that needs to be stored on each enterprise key recovery enabled system.

### 5.2 Choosing a Configuration

Every key recovery configuration involves a single KRC and an arbitrary number of KRAs. The first step in the installation and configuration process is to choose the number of KRAs that will be involved in the key recovery scenario. As previously stated, increasing the number of KRAs provides added levels of protection against collusion between independent KRAs.

Each system in the configuration must be configured with and be reachable using TCP/IP. Additionally, the KRC Server and KRA Servers listen for requests using TCP protocol. The TCP port numbers at which they listen for requests need to be chosen. These TCP/IP address and ports will be needed in later configuration steps to help all the parties to communicate.

Each KRA Server must be given a name by which to name its configuration data. This name must begin with the prefix “KRA\_”.

Enterprise KROs must also choose a password that will authorize them to receive recovered keys. Each key recovery enabled system in the enterprise needs to have an Authorization Information (AI) file that is derived from this password so that each KRB is associated with this AI. Each KRB generated by KRSPs on key recovery enabled systems contains this AI. When a KRB is submitted to the KRC as part of a key recovery request, the KRO’s password is checked against the KRB for validation of the request. To create the AI file, the KRO must use the AI Utility (SETAUTHINFO.EXE) on a system that has the KRSP installed and configured.

### 5.3 Setting Up Secure Facilities

The KRC Server and each of the KRA Servers should run in a Key Recovery Secure Facility (KRSF) as specified in Appendix A, Security Practices. This involves securing the site that houses the systems running the software, as well as configuring the Windows NT 4.0 operating system properly. It is strongly recommended that these steps be followed before installation of the KRS. The KRO Client system should be physically secured because of the sensitive information that will be processed.

## 5.4 Installing the Software

For each system in the configuration, the relevant software components should be installed by following the instructions on the IBM Key Recovery Server CD-ROM. The following are the minimum requirements for each system:

- Enterprise KRO – this system requires the Key Recovery Officer Client.
- Key Recovery Coordinator – this system requires the Key Recovery Coordinator Server and the Key Recovery Officer Client (for law enforcement requests).
- Key Recovery Agent – this system requires the Key Recovery Agent Server.

## 5.5 Obtaining Certificates and Private Keys

The enablement phase and the recovery phase of key recovery work together through the matching certificate and private key pairs that are configured on the systems involved.

A certificate and private key pair is needed for the following:

- Key Recovery Agent Server(s)
- Key Recovery Coordinator Server
- Key Recovery Officer Client

Only the certificates are needed for Trusted Anchors (used to verify the other certificates).

The IBM Key Recovery Server uses ASN/DER-encoded X.509 Version 3 certificates. The public/private key pair is comprised of 1024 bit RSA keys. The pub key is stored in the certificate. The private key is DER-encoded, encrypted, and stored in a file. The public key certificates must be sent to IBM for conversion into a form that the IBM KRSP can use. The resulting **sealed** configuration files must then be installed on all **key recovery enabled systems** where KRBs will be generated.

A sample set of certificates, private keys, and KRSP configuration files has been generated for a configuration with two KRA Servers. Refer to the instructions distributed with the CD-ROM for more details.

## 5.6 Master Configuration File

Many of the Key Recovery Server components rely on a master configuration file called KRS.CFG. The master configuration file is stored in the CFG subdirectory below the installation directory, which is chosen at install time. For example, if the installation directory was chosen to be C:\IBMKRS, the absolute path of the master configuration file would be C:\IBMKRS\CFG\KRS.CFG. An example of a master configuration file follows:

```
[KRO]
certfile = kro.cert
certprivkpair = kro.cert, d:\private\kro.priv

[KRC]
certprivkpair = ..\krc\krc.cert
netinfo = krc.company.com, 29003
```

Each section corresponds to a role within the key recovery phase. Each attribute for a given role can have one or more values separated by commas.

The valid section names are:

- [KRO] – for attributes related to the KRO Client
- [KRC] – for attributes related to the KRC Server
- [KRAName] – for attributes related to the named KRA. KRA names must begin with the prefix “KRA\_”.

The format of valid attribute/value pairs are:

- certfile = FileContainingCertificate
- certprivkpair = FileContainingCertificate,FileContainingPrivateKey
- netinfo = IPAddress Or Name,IPPort

The certfile value is the name of a file that contains an ASN/DER-encoded X.509 Version 3 public key certificate. The certprivkpair value is the name of two files: the file containing the certificate and the file containing the private key. The netinfo value contains the TCP/IP address of a server followed by the port number that the server will use to listen for incoming requests.

## 5.7 Configuring a Key Recovery Officer Client

The KRO Client requires access to certificates and private keys specified in the master configuration file. The KRO Client needs this to contact the KRC and communicate with it securely. Set the following attribute/value pairs after copying the named files onto the system:

[KRO]

certprivkpair = FileContainingCertificate, FileContainingPrivateKey

[KRC]

certfile = FileContainingCertificate

In the enterprise and law enforcement scenarios, the KRO needs to contact the KRC using TCP/IP. Therefore, the following attribute/value pair needs to be added to the master configuration file:

[KRC]

netinfo = NetworkAddress, NetworkPort

## 5.8 Configuring a Key Recovery Coordinator Server

The KRC Server requires the following attribute/value pairs in order for it to communicate securely and receive sealed information. The KRA information must be separated in sections that have the same name as the KRA. This name also will be needed at KRA startup time.

[KRC]

certprivkpair = FileContainingCertificate, FileContainingPrivateKey

[KRA\_1]

certfile = FileContainingCertificate

netinfo = NetworkAddress, NetworkPort

[KRA\_2]

certfile = FileContainingCertificate  
netinfo = NetworkAddress, NetworkPort

## 5.9 Configuring a Key Recovery Agent Server

The KRA Server requires the following attribute/value pairs in order for it to communicate securely and for it to compute secondary keys in the recovery sequence. The name of the section should be the name of the KRA itself.

[KRA\_1]

certprivkpair = FileContainingCertificate, FileContainingPrivateKey

## Chapter 6. Operation

The operational aspects of each of the components of the Key Recovery Server (KRS) are discussed in detail in the following subsections.

### 6.1 Key Recovery Officer

The Key Recovery Officer (KRO) consists of one main application, the KRO application, which may be used to initiate a key recovery request operation. Two types of scenarios are used to perform a key recovery request operation: the Enterprise scenario and the Law Enforcement scenario. In the Enterprise scenario, the KRO has both the key recovery block and the credentials that authenticate it to receive the recovered key. This information will be transmitted over the network to the Key Recovery Coordinator (KRC). In the Law Enforcement scenario, all recovery requests are brought to the KRO administrator using a physical medium such as a floppy diskette.

#### 6.1.1 Sending an Enterprise Key Recovery Request

The actual key recovery phase begins when the KRO administrator uses the KRO application to initiate a key recovery request to the appropriate KRC. At this time, the administrator selects a Key Recovery Block (KRB) to be sent for recovery, enters the Authentication Information (AI) that can be used to authenticate the request to the KRC, and submits the request. The KRO administrator manually starts up the KRO application based on the input the administrator receives from an enterprise employee who needs to recover a key embedded within a KRB. At the time the KRO application is started up, it expects the KRO administrator to provide a password to establish the administrator's access to the application. It is assumed that the KRO administrator has the means to look up the AI value for the owner of the KRB. This AI value is then used, along with the KRB, to send a recovery request to the KRC.

To send an online key recovery request:

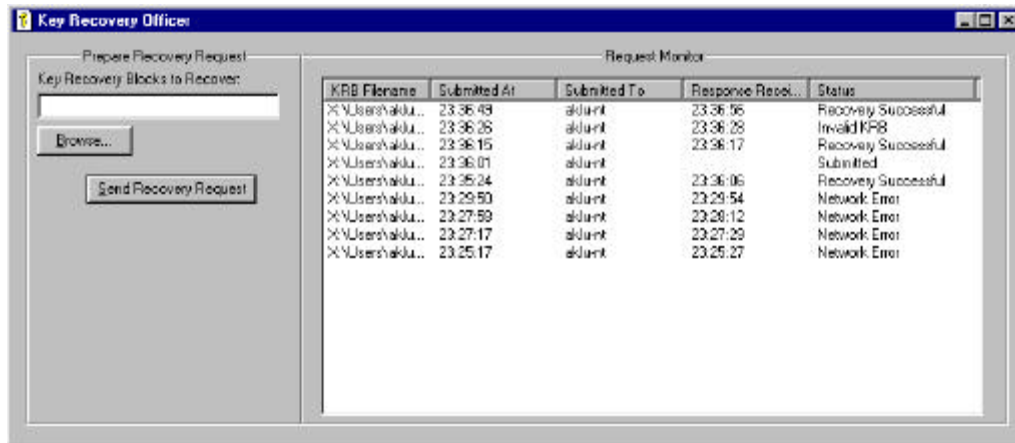
1. The KRO administrator receives a key recovery request from the enterprise in the form of a Key Recovery Block (where the KRB could be an email attachment).
2. The enterprise must also send the KRO administrator identification information that allows the KRO to search its AI Database for an AI value.
3. The KRO administrator enters the KRB Authentication password to access the KRO application (Figure 1) and clicks on the Continue button.



Figure 1. KRO Password

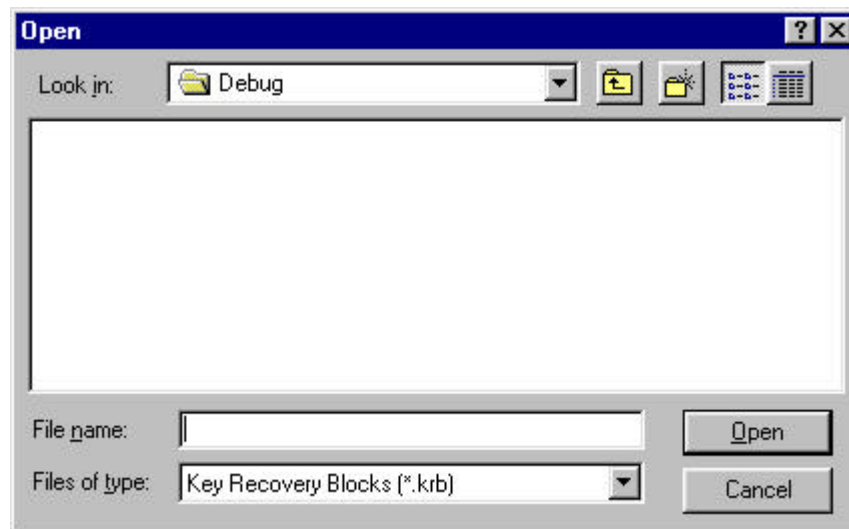


- Once the password is accepted, the KRO administrator can start the KRO application. The Key Recovery Officer Request Monitor screen appears (Figure 2).



**Figure 2. Key Recovery Officer Request Monitor**

- The KRO administrator clicks on the Browse button in the Key Recovery Officer Request Monitor screen to select the key recovery block file to be sent for recovery. The KRB Browse Directory appears (Figure 3).



**Figure 3. KRB Browse Directory**

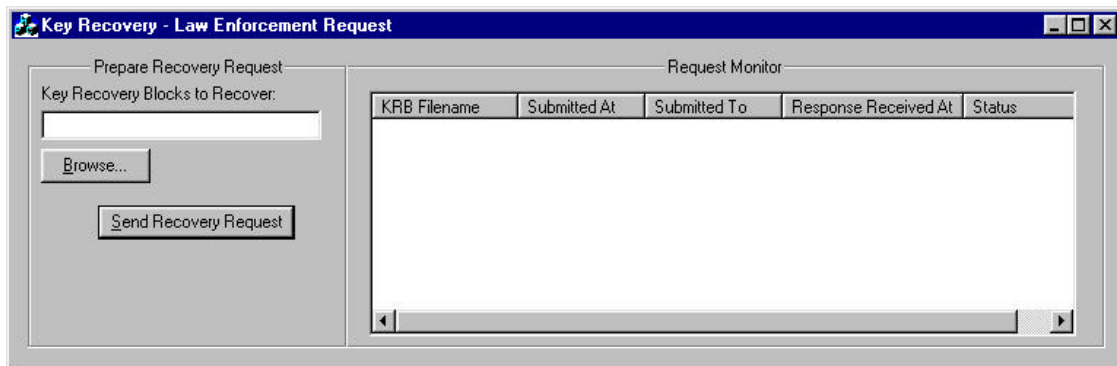
- The KRO administrator selects the desired KRB file from the KRB Browse Directory, clicks on the Open button, and is immediately returned to the Key Recovery Officer Request Monitor screen. The selected file will appear in the Key Recovery Blocks to Recover field.
- The KRO administrator clicks on the Send Recovery Request button to send the KRB file to the KRC.

## 6.1.2 Sending a Law Enforcement Key Recovery Request

Sending a Law Enforcement key recovery request involves bypassing the KRO stage and moving directly to the KRC.

To send a manual law enforcement key recovery request:

1. The law enforcement agent must present the KRB on a 3.5" diskette to the KRC operator and submit the necessary credentials that are in the physical form of a legal warrant. This warrant will specify any information available to the law enforcement agencies which can be used to tie the warrant to the identity of the user for whom KRBs were generated (i.e., username, hostname, IP address).
2. The KRC operator checks the credentials and begins the recovery of the original encryption key from the KRB with the help of its Key Recovery Agents (KRAs).
3. The KRC operator uses the KRB to initiate a key recovery operation. The KRC provides a graphical user interface (GUI) driven application that allows the KRC operator to initiate a law enforcement operation.
4. The KRC must verify whether the legal authorization documents contain information that can be verified against the AI information embedded within the KRB. The GUI at the KRC displays the AI information within a KRB to allow the KRC operator to verify that the AI in the KRB corresponds to information contained in the legal authorization document.
5. Once the AI information is verified, the KRC accesses the Key Recovery Law Enforcement Request screen (Figure 4). The KRC installs the 3.5" diskette, which contains the KRB files, into the A:\ drive.
6. The KRC operator clicks on the Browse button in the Key Recovery Law Enforcement Request screen to select the desired key recovery block file from the A:\ drive. The file appears in the Key Recovery Blocks to Recover field in the Key Recovery Law Enforcement Request screen.
7. The KRO operator clicks on the Send Recovery Request button. The Load Certificate screen appears (Figure 5).

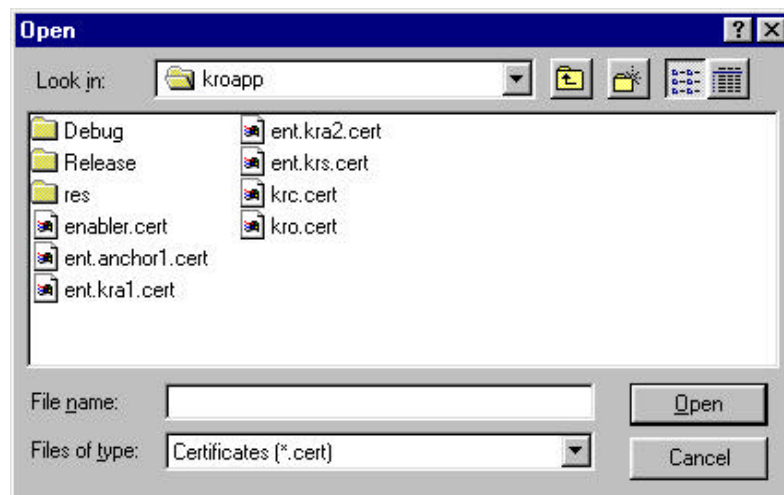


**Figure 4. Key Recovery Law Enforcement Request**



**Figure 5. Load Certificate Screen**

8. The KRC operator clicks on the Browse button in the Load Certificate screen. The Certificate Directory appears (Figure 6).
9. The KRC operator selects the desired certificate file, which appears in the Load Certificate screen. This selected certificate will be used to encrypt the recovery key.
10. The KRC operator clicks on the OK button to submit the key recovery request. The KRC Server will decode the KRB and send it to the appropriate KRAs.



**Figure 6. Certificate Directory**

### 6.1.3 Monitoring the Status of Requests

The Key Recovery Officer application provides a request monitoring facility (see Figure 2) that allows the KRO administrator to monitor the status of all recovery requests that are being serviced or have been serviced at any point in time. The following status information is displayed in the Key Recovery Officer Request Monitor screen:

- KRB Filename – Identifies the filename of the KRB sent by the KRO.
- Submitted At – Identifies the time of day the key recovery request was sent from the KRO. The time is shown in military format (hh/mm/ss).
- Submitted To – Identifies the IP address of the KRC.

- Response Received At – Identifies the time of day of when the response comes back from the KRC.
- Status – Identifies the current status of the recovery request (e.g., recovery successful, request aborted, server failure, etc).

#### **6.1.4 Optional Decryption of Data**

Once the key recovery request has been completed, and the recovered key returned to the Key Recovery Officer, the encryption utility can then be used to decrypt the original data. This can be done by selecting the encrypted file in the “Files” field, clicking on the “Use file as key” button, and selecting the recovered key file in the “Key file” field. The utility will automatically determine the algorithm and mode required to decrypt the data from the contents of the key file. The user will be prompted for a filename with which to save the original clear data. It will be according to particular enterprise policies whether the decrypted data is recovered at the site of the Key Recovery Officer, or at the original client’s desktop where the file was encrypted.

## **6.2 Key Recovery Coordinator**

The Key Recovery Coordinator (KRC) consists of one main application, the KRC application, which when started behaves as a daemon or server process that continuously listens for incoming requests, until the process is terminated either explicitly or at system shutdown.

The KRC application uses several files to perform its operation. These files are accessible through utilities that are available on the KRC system. The KRC Database file holds entries that contain certificates and network information for KRA entities that are known to the KRC. The KRC Database file also contains information about the KRC’s own certificates and the names of the files that contain the corresponding private keys. The private key files reside on disk encrypted with a password that the KRC operator provides when the KRC application is started.

### **6.2.1 Starting the Server**

The KRC application can be started using two different methods: the application is configured to start as part of system services, or the KRC operator can start up the KRC application manually.

To start the KRC application using system services:

1. Whenever the KRC application is started up, the KRC icon will appear in the status area of the taskbar where it continuously runs in the background.

To start the KRC server manually:

1. The KRC operator has the option of starting the application by accessing the KRC application icon (Key Recovery Coordinator) using the Windows NT startup menu. The KRC icon will appear in the status area of the taskbar where it continuously runs in the background.

### **6.2.2 Monitoring the Status of Requests**

The KRC application provides a request monitoring facility that allows the KRO administrator to monitor the status of all recovery requests that are being serviced or have been serviced at any point in time. The following status information is displayed in the Key Recovery Officer Request Monitor screen (Figure 7):

- Session Id – Provides a unique numeric value identifying a single key recovery request.
- Request Received At – Identifies the time of day a key recovery request was received from a KRO.

- Originator – Identifies the IP address of the KRO who initiated the key recovery request.
- Number of KRAs – Identifies the number of KRAs involved in recovering the key from the KRB.
- Responses From KRAs – Identifies the number of KRAs who have responded to the KRC.
- Response Posted At – Identifies the time of day that a response was posted to the originating KRO.
- Status – Identifies the current status of the recovery request (e.g., recovery successful, request aborted, KRC failure, etc).

The screenshot shows a window titled "Key Recovery Coordinator" with a sub-window titled "Request Monitor". The "Request Monitor" window contains a table with the following data:

Session Id	Request Received At	Originator	Number of KRAs	Responses From KRAs	Response Posted At	Status
18260968	23:38:21	9.210.116.1	2	2	23:38:21	Recovery Successful
18260967	23:37:57	9.210.116.1	0	0	23:37:57	Invalid KRB
18260966	23:37:46	9.210.116.1	2	2	23:37:47	Recovery Successful
18260965	23:37:32	9.210.116.1	2	2	23:37:33	Recovery Successful

**Figure 7. Key Recovery Coordinator Request Monitor**

### 6.2.3 Viewing the Audit Trail

The KRC operator is responsible for viewing the audit trail, which is available under the Microsoft Windows NT Event Viewer administrative tool. The audit trail logs the events that occur during the key recovery operation. These events include the following:

- Successful recoveries
- Communication between key recovery entities
- Authentication failures
- Cryptographic failures
- Network errors
- Server startup and shutdown

## 6.3 Key Recovery Agent

The Key Recovery Agent (KRA) consists of one main application, the KRA application, which when started behaves as a daemon or server process that continuously listens for incoming requests, until the process is terminated either explicitly or at system shutdown. The system, which serves as the KRA, may be configured to start the KRA application as part of system services; alternatively, the KRA operator can start up the KRA application manually.

The KRA application uses several files to perform its operation. These files are accessible through utilities that are available on the KRA system. The KRA Database file holds entries that contain information about the KRA's own certificates and the names of the files that contain the corresponding private keys. The private key files reside on disk encrypted with a password that the KRA operator provides when the KRA application is started.

### 6.3.1 Starting the Server

The KRA application can be started using two different methods: the application is configured to start as part of system services, or the KRA operator can start up the KRA application manually.

To start the KRA application using system services:

1. Whenever the KRA application is started up, the Key Recovery Agent Name screen appears (Figure 8) prompting for the KRA Name. The KRA icon will appear in the status area of the taskbar where it continuously runs in the background.

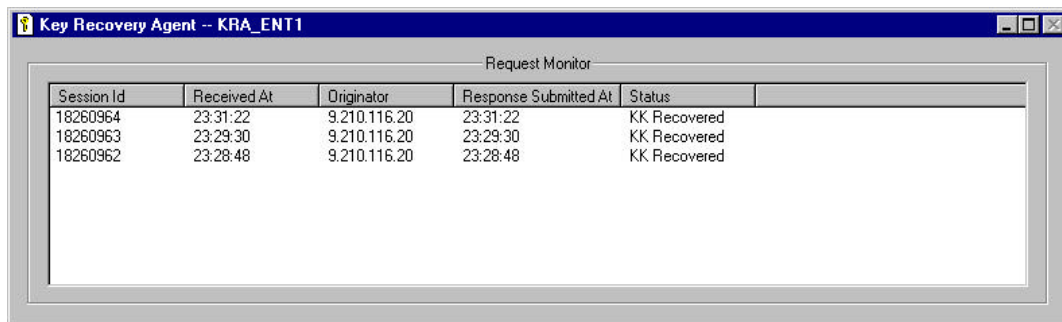


Figure 8. Key Recovery Agent Name

### 6.3.2 Monitoring the Status of Requests

The KRA application provides a request monitoring facility that allows the KRA operator to monitor the status of all recovery requests that are being serviced or have been serviced at any point in time. The following status information is displayed in the Key Recovery Agent Request Monitor screen (Figure 9):

- Session Id – Provides a unique numeric value identifying a single key recovery request.
- Received At – Identifies the time of day when a KK request was received from the KRC .
- Originator – Identifies the IP address of the KRO who initiated the key recovery request.
- Response Submitted At – Identifies the time of day the KRA responded to the KRC's request.
- Status – Identifies the current status of the recovery request (e.g., recovery successful, request aborted, KRA failure, etc).



Session Id	Received At	Originator	Response Submitted At	Status
18260964	23:31:22	9.210.116.20	23:31:22	KK Recovered
18260963	23:29:30	9.210.116.20	23:29:30	KK Recovered
18260962	23:28:48	9.210.116.20	23:28:48	KK Recovered

Figure 9. Key Recovery Agent Request Monitor

# Appendix A. Pre-Installation Setup

The Key Recovery Server (KRS) component facilities are high-value entities and are thus prime targets for attack. Thus, elaborate measures need to be taken to ensure that these facilities operate within a secure environment that is highly resistant to penetration and compromise. In addition, the KRC, KRA and KRO needs to hold, maintain, and use one or more RSA key pairs to perform their operations. These key pairs constitute very sensitive information, and need to be maintained and used in a manner that does not undermine the confidentiality of the private keys. This section provides guidelines to prepare a secure facility and a machine that has been loaded with a fresh copy of Windows NT 4.0 and before the KRS software is installed.

## A.1 Facility Requirements

This section will outline the physical and administrative security procedures that are to be followed at the Key Recovery Server Facility (KRSF). They include the following:

- Key Recovery Server Facility Room
- Limited Personnel Access

### A.1.1 Key Recovery Server Facility Room

The KRSF will be a room with a single entrance/exit whose doorway is secured by a combination lock. The room should have no windows and a secured ceiling.

### A.1.2 Limited Personnel Access

Access to the facility must be restricted to a limited number of personnel. At all times, a two-person rule will be imposed for access to the KRSF. The combination lock key for the KRSF should be split among two designated security officers. Both security officers have to be present in order to open the KRSF door.

## A.2 Server Physical Requirements

The system requirements to be followed in the KRSF include the following:

- Keyed Lock for System
- Dedicated System
- Removable Backup Media Connectivity
- Microsoft Windows NT 4.0

### A.2.1 Keyed Lock for System

The system will have its own keyed lock to assure no tampering.

### A.2.2 Dedicated System

The IBM KRSF will house a dedicated system that runs only the IBM Key Recovery Server.

### A.2.3 Removable Backup Media Connectivity

The system will be connected to a removable backup media (Jaz drive.) to assure timely backup source for database redundancy.

#### **A.2.4 Microsoft Windows NT 4.0**

The KRSF uses a secure platform (NT Workstation 4.0) which supports identification and authentication as well as filesystem level access control. A single non-administrative account on this system should be set up; the account password should be broken up into two portions and each portion is revealed to one of the two security officers. The goal is to make sure that both security officers are present to log in to the KRSF system and perform KRSF operations.

There will be at least 10 characters in the password so that the password can be split between dual entities for access to the system. Passwords must be changed frequently to ensure higher level security. Account lockout will occur after a small number(s) of login attempts, and will stay locked until the security administrator unlocks the account.

The Windows NT 4.0 Auditing Policies will provide information relevant to the following events:

- User logs on in order to change password.
- All logon and logoff to conduct recovery operations
- Changes to user rights
- User and group management rights changes
- Security policy changes

### **A.3 Installation**

Once the above steps are completed to prepare the system, the installation of the key recovery system may proceed. Steps described on the CD should be followed for the complete installation.

The process described above is a recommendation based on IBM experience in using this product. IBM DISCLAIMS ALL WARRANTIES EXPRESS AND IMPLIED WITH RESPECT TO THE USE OF THESE RECOMMENDATIONS INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.



## Appendix B. List of Acronyms

AI	Authentication Information
API	Application Program Interface
CA	Certificate Authority
CDSA	Common Data Security Architecture
CLI	Certificate Library Service Provider Interface
CRL	Certificate Revocation Lists
CSP	Cryptographic Service Provider
CSSM	Common Security Services Manger
DLI	Data Store Library Service Provider Interface
EDI	Electronic Data Interchange
GUI	Graphical User Interface
ISV	Independent Software Vendor
KRA	Key Recovery Agent
KRB	Key Recovery Block
KRC	Key Recovery Coordinator
KRF	Key Recovery Field
KRMM	Key Recovery Module Manager
KRO	Key Recovery Officer
KRS	Key Recovery Server
KRSF	Key Recovery Security Facility
KRSP	Key Recovery Service Provider
PKI	Public Key Infrastructure
SCCS	Secure Cryptography and Certificate Services
SET	Secure Electronic Transaction
SP	Service Provider
SPI	Service Provider Interface
SSL	Secure Socket Layer
TP	Trust Policy

## Appendix C. Glossary

Asymmetric algorithms	Cryptographic algorithms where one key is used to encrypt, and a second key is used to decrypt. They are often called public-key algorithms. One key is called the public key, and the other is called the private key or secret key. RSA (Rivest-Shamir-Adelman) is the most commonly used public-key algorithm. It can be used for encryption and for signing.
Authentication Information (AI)	Information that is verified for authentication. For example, a Key Recovery Officer (KRO) selects a password which will be used for authentication with the Key Recovery Coordinator (KRC). A KRO operator who has identification information must search the Authentication Information Database to locate an AI value that corresponds to the individual who generated the information.
Certificate Authority (CA)	An entity that guarantees or sponsors a certificate. For example, a credit card company signs a cardholder's certificate to assure that the cardholder is who he or she claims to be. The credit card company is a certificate authority. Certificate authorities issue, verify, and revoke certificates.
Certificate	See Digital certificate.
Certificate chain	The hierarchical chain of all the other certificates used to sign the current certificate. This includes the Certificate Authority (CA) who signs the certificate, the CA who signed that CA's certificate, and so on. There is no limit to the depth of the certificate chain.
Certificate signing	The Certificate Authority (CA) can sign certificates it issues or co-sign certificates issued by another CA. In a general signing model, an object signs an arbitrary set of one or more objects. Hence, any number of signers can attest to an arbitrary set of objects. The arbitrary objects could be, for example, pieces of a document for libraries of executable code.
Certificate validity date	A start date and a stop date for the validity of the certificate. If a certificate expires, the Certificate Authority (CA) may issue a new certificate.
Cryptographic Service Providers (CSPs)	Modules that provide secure key storage and cryptographic functions. The modules may be software only or hardware with software drivers. The cryptographic functions provided may include: <ul style="list-style-type: none"><li>• Bulk encryption and decryption</li><li>• Digital signing</li><li>• Cryptographic hash</li><li>• Random number generation</li><li>• Key exchange</li></ul>
Cryptographic algorithm	A method or defined mathematical process for implementing a cryptography operation. A cryptographic algorithm may specify the procedure for encrypting and decrypting a byte stream, digitally signing an object, computing the hash of an object, generating a random number, etc. SCCS accommodates DES, RC2, RC4, IDEA and other encryption algorithms.

Cryptoki	Short for cryptographic token interface. See Token.
Digital certificate	The binding of some identification to a public key in a particular domain, as attested to directly or indirectly by the digital signature of the owner of that domain. A digital certificate is an unforgettable credential in cyberspace. The certificate is issued by a trusted authority, covered by that party's digital signature. The certificate may attest to the certificate holder's identity, or may authorize certain actions by the certificate holder. A certificate may include multiple signatures and may attest to multiple objects or multiple actions.
Digital signature	<p>A data block that was created by applying a cryptographic signing algorithm to some other data using a secret key. Digital signatures may be used to:</p> <ul style="list-style-type: none"> <li>• Authenticate the source of a message, data, or document</li> <li>• Verify that the contents of a message hasn't been modified since it was signed by the sender</li> <li>• Verify that a public key belongs to a particular person</li> </ul> <p>Typical digital signing algorithms include MD5 with RSA encryption, and DSS, the proposed Digital Signature Standard defined as part of the U.S. Government Capstone project.</p>
Enterprise	A company or individual who is authorized to submit on-line requests to the Key Recovery Officer (KRO). In the enterprise key recovery scenario, a process at the enterprise called the KRO is responsible for preparing key recovery requests and communicating them to the KRC. The KRO, acting on behalf of an enterprise or individual, sends off an on-line request to the Key Recovery Coordinator (KRC) to recover a key from a Key Recovery Block (KRB).
Graphical User Interface	A type of display format that enables the user to choose commands, start programs, and see lists of files and other options by pointing to pictorial representations (icons) and lists of menu items on the screen. Graphical user interfaces are used by the Microsoft Windows program for IBM-compatible microcomputers and by other systems.
Hash algorithm	A cryptographic algorithm used to hash a variable-size input stream into a unique, fixed-sized output value. Hashing is typically used in digital signing algorithms. Example hash algorithms include MD and MD2 from RSA Data Security. MD5, also from RSA Data Security, hashes a variable-size input stream into a 128 bit output value. SHA, a Secure Hash Algorithm published by the U.S. Government, produces a 160-bit hash value from a variable-size input stream.

Key Recovery Agent	<p>The Key Recovery Agent (KRA) acts as the back end for a key recovery operation. The KRA can only be accessed through an online communication protocol via the Key Recovery Coordinator (KRC). KRAs are considered outside parties involved in the key recovery process; they are analogous to the neighbors who each hold one digit of the combination of the lock box containing the key. The authorized parties (i.e., enterprise or Law Enforcement) have the freedom to choose the number of specific KRAs that they wish to use. The authorized party requests that each KRA decrypt its section of the Key Recovery Fields (KRFs) that is associated with the transmission. Then those pieces of information are used in the process that derives the session key. The KRA will only be able to recover a portion of the key, and reading the original message will require searching the remaining key space in order to find the key that will decrypt the message. The number of KRAs on each end of the communication does not have to be equal.</p>
Key Recovery Block	<p>The Key Recovery Block (KRB) is a piece of encrypted information that is contained within a block. The KRS components (i.e., KRO, KRC, KRA) work collectively to recover a session key from a provided key recovery block. In the enterprise scenario, the KRO has both the KRB and the credentials that authenticate it to receive the recovered key. This information will be transmitted over the network to the KRC. In the law enforcement scenario, the KRB is presented on a 3.5" diskette, and the credentials are in the physical form of a legal warrant. This warrant will specify any information available to the law enforcement agents which can be used to tie the warrant to the identity of the use for whom KRBs were generated (i.e., username, hostname, IP address.) The KRC has the ability to check credentials and derive the original encryption key from the KRB with the help of its KRAs.</p>
Key Recovery Coordinator	<p>The Key Recovery Coordinator (KRC) acts as the front end for the key recovery operation. The KRO acting on behalf of an enterprise or individual sends off an on-line request to the KRC to recover a key from a KRB. The KRC receives the on-line request and services it by interacting with the appropriate set of KRAs as specified within the KRB. The recovered key is then sent back to the KRO by the KRC using an on-line protocol. The KRC consists of one main application, which when started, behaves as a server process. The system, which serves as the KRC, may be configured to start the KRC application as part of system services; alternatively, the KRC operator can start up the KRC application manually. The KRC application performs the following operations:</p> <ul style="list-style-type: none"> <li>• Listens for on-line recovery requests from KRO</li> <li>• Can be used to launch an embedded application that allows manual key recovery for law enforcement</li> <li>• Monitors and displays the status of the recovery requests being serviced</li> </ul>

Key Recovery Fields	A Key Recovery Field (KRF) is a block of data which is created from a symmetric key and key recovery profile information. The Key Recovery Service Provider (KRSP) is invoked from the SCCS framework to create the key recovery fields. There are two major pieces of the key recovery fields: block 1 contains information that is unrelated to the session key of the transmitted message and encrypted with the public keys of the selected key recovery agents; block 2 contains information that is related to the session key of the transmission. The KRSP generates the KRFs for the session key. This information is NOT the key or any portion of the key, but is information that can be used to recover the key. The KRSP has access to location-unique jurisdiction policy information that controls and modifies some of the steps in the generation of the KRFs. Only once the KRFs are generated, and both the client and server sides have access to them, can the encrypted message flow begin. KRFs are generated so that they can be used by a KRA to recover the original symmetric key, either because the user who generated the message has lost the key, or at the warranted request of law enforcement agents.
Key Recovery Module Manager	The Key Recovery Module Manager (KRMM) enables key recovery for cryptographic services obtained through the SCCS. It mediates all cryptographic services provided by the SCCS and applies the appropriate key recovery policy on all such operations. The KRMM contains a Key Recovery Policy Table (KRPT), which defines the applicable key recovery policy for all cryptographic products. KRMM routes the KR-API function calls made by an application to the appropriate KR-SPI functions. The KRMM also enforces the key recovery policy on all cryptographic operations that are obtained through the SCCS. It maintains key recovery state in the form of key recovery contexts.
Key Recovery Officer	An entity called the Key Recovery Officer (KRO) is the focal point of the key recovery process. In the enterprise key recovery scenario, the KRO is responsible for preparing key recovery requests and communicating them to the KRC. The KRO has both the KRB and the credentials that authenticate it to receive the recovered key. The Key Recovery Officer is the entity that acts on behalf of an enterprise to initiate a key recovery request operation. An employee within an enterprise who desires key recovery will send a request to the KRO with the KRB that is to be recovered. The actual Key recovery phase begins when the KRO operator uses the KRO application to initiate a key recovery request to the appropriate KRC. At this time, the operator selects a KRB to be sent for recovery, enters the AI information that can be used to authenticate the request to the KRC, and submits the request.
Key Recovery Policy	Key recovery policies are mandatory policies that are typically derived from jurisdiction-based regulations on the use of cryptographic products for data confidentiality. Often, the jurisdictions for key recovery policies coincide with the political boundaries of countries, in order to serve the law enforcement and intelligence needs of these political jurisdictions. Political jurisdictions may choose to define key recovery policies for cryptographic products based on export, import, or use controls. Enterprises may define internal and external jurisdictions, and may mandate key recovery policies on the cryptographic products within their own jurisdictions.

Key recovery policies come in two flavors: *key recovery enablement policies* and *key recovery inter-operability policies*. Key recovery enablement policies specify the exact cryptographic protocol suites (algorithms, modes, key lengths etc.) and perhaps usage scenarios, where key recovery enablement is mandated. Furthermore, these policies may also define the number of bits of the cryptographic key that may be left out of the key recovery enablement operation; this is typically referred to as the *workfactor*. Key recovery inter-operability policies specify to what degree a key-recovery-enabled cryptographic product is allowed to inter-operate with other cryptographic products.

Key Recovery Server Facility	The Key Recovery Server Facility (KRSF) is a facility room which houses the KRS component facilities ensuring they operate within a secure environment that is highly resistant to penetration and compromise. Several physical and administrative security procedures must be followed at the KRSF such as a combination keyed lock, limited personnel, standalone system, operating system with security features (Microsoft NT Workstation 4.0), NTFS Filesystem, and account and auditing policies.
Key Recovery Server	The Key Recovery Server (KRS) consists of three major entities: Key Recovery Coordinator (KRC), Key Recovery Agent (KRA), and Key Recovery Officer (KRO). The KRS is intended to be used by enterprise employees and security personnel, law enforcement personnel, and Key Recovery Facilities Server (KRSF) Personnel. The key recovery server interacts with one or more local or remote key recovery agents to reconstruct the secret key that can be used to decrypt the ciphertext.
Key Recovery Service Providers (KRSPs)	Modules that provide key recovery enablement functions. The cryptographic functions provided may include: <ul style="list-style-type: none"><li>• Key recovery field generation</li><li>• Key recovery field processing</li></ul>
Law Enforcement	A type of scenario where key recovery is mandated by the jurisdictional law enforcement authorities in the interest of national security and law enforcement. In the law enforcement scenario, the KRB is presented on a 3.5" diskette, and the credentials are in the physical form of a legal warrant. This warrant will specify any information available to the law enforcement agents which can be used to tie the warrant to the identity of the use for whom KRBs were generated (i.e., username, hostname, IP address).
Leaf Certificate	The certificate in a certificate chain that has not been used to sign another certificate in that chain. The leaf certificate is signed directly or transitively by all other certificates in the chain.
Message digest	The digital fingerprint of an input stream. A cryptographic hash function is applied to an input message arbitrary length and returns a fixed-size output, which is called the digest value.
Owned certificate	A certificate whose associated secret or private key resides in a local CSP. Digital-signing algorithms require using owned certificates when signing data for purposes of authentication and non-repudiation. A system may use certificates it does not own for purposes other than signing.

Private key	The cryptographic key used to decipher messages in public-key cryptography. This key is kept secret by its owner.
Public key	The cryptographic key used to encrypt messages in public-key cryptography. The public key is available to multiple users (i.e., the public).
Random number generators	A function that generates cryptographically strong random numbers that cannot be easily guessed by an attacker. Random numbers are often used to generate session keys.
Root certificate	The prime certificate, such as the official certificate of a corporation or government entity. The root certificate is positioned at the top of the certificate hierarchy in its domain, and it guarantees the other certificates in its certificate chain. Each Certificate Authority (CA) has a self-signed root certificate. The root certificate's public key is the foundation of signature verification in its domain.
Secure Cryptography and Certificate Services Architecture	A set of layered security services that address communications and data security problems in the emerging PC business space.
Secure Cryptography and Certificate Services Framework	<p>The SCCS framework defines five key service components:</p> <ul style="list-style-type: none"> <li>• Cryptographic Module Manager</li> <li>• Key Recovery Module Manager</li> <li>• Trust Policy Module Manager</li> <li>• Certificate Library Module Manager</li> <li>• Data Storage Library Module Manager</li> </ul> <p>The SCCS binds together all the security services required by PC applications. In particular, it facilitates linking digital certificates to cryptographic actions and trust protocols.</p>
Security Context	A control structure that retains state information shared between a cryptographic service provider and the application agent requesting service from the CSP. Only one context can be active for an application at any given time, but the application is free to switch among contexts at will, or as required. A security context specifies CSP and application-specific values, such as required key length and desired hash functions.
Security-relevant event	An event where a CSP-provided function is performed, a security module is loaded, or a breach of system security is detected.
Session key	A cryptographic key used to encrypt and decrypt data. The key is shared by two or more communicating parties, who use the key to ensure privacy of the exchanged data.
Signature	See Digital signature.
Signature chain	The hierarchical chain of signers, from the root certificate to the leaf certificate, in a certificate chain.

Symmetric algorithms	Cryptographic algorithms that use a single secret key for encryption and decryption. Both the sender and receiver must know the secret key. Well-known symmetric functions include DES (Data Encryption Standard) and IDEA. The U.S. Government endorsed DES as a standard in 1977. It's an encryption block cipher that operates on 64-bit blocks with a 56-bit key. It is designed to be implemented in hardware, and works well for bulk encryption. IDEA (International Data Encryption Algorithm), one of the best known public algorithms, uses a 128-bit key.
Token	The logical view of a cryptographic device, as defined by a CSP's interface. A token can be hardware, a physical object, or software. A token contains information about its owner in digital form, and about the services it provides for electronic-commerce and other communication applications. A token is a secure device. It may provide a limited or a broad range of cryptographic functions. Examples of hardware tokens are smartcards and PCMCIA cards.
Verification	The process of comparing two message digests. One message digest is generated by the message sender and included in the message. The message recipient computes the digest again. If the message digests are exactly the same, it shows or proves there was no tampering of the message contents by a third party (between the sender and the receiver).
Web of trust	A trust network among people who know and communicate with each other. Digital certificates are used to represent entities in the web of trust. Any pair of entities can determine the extent of trust between the two, based on their relationship in the web. Based on the trust level, secret keys may be shared and used to encrypt and decrypt all messages exchanged between the two parties. Encrypted exchanges are private, trusted communications.