EMC® Documentum®
Trusted Content Services

Version 7.3

Cryptography and FIPS 140-2 Guide

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Table of Contents

List of Figures

List of Tables

Preface

This guide describes information about the cryptographic modules used in Documentum Trusted Content Services, and in dependent applications. It lists information about the modules used, where they are used, and the FIPS 140–2 certifications of those modules. This guide also describes where to obtain information about using FIPS 140–2 modules in the third-party application server that hosts Trusted Content Services-dependent applications.

Intended Audience

This manual is written for system and repository administrators, and for application server administrators. The system administrator is the person who generally installs and owns the Documentum installation. Repository administrators are the users who own and are responsible for one or more repositories. Application server administrators are the users who own and are responsible for the application server that hosts Trusted Content Services-dependent applications. Readers should be familiar with the general principles of client/server architecture and networking. In addition, they should know and understand the Windows and Linux operating systems.

Revision History

Revision Date	**Description**
November 2016	Initial publication.

Chapter 1

FIPS 140-2 and Trusted Content Services

Introduction

The Federal Information Processing Standard (FIPS) Publication 140-2 is a U.S. government computer security standard used to validate cryptographic modules. Starting with the 7.0 release, EMC Documentum Trusted Content Services installs only FIPS 140-2 validated cryptographic modules. Any exceptions are listed in this document.

FIPS 140-2 certification is handled by third parties who certify the FIPS 140-2 libraries. FIPS 140-2 compliance is achieved by using these FIPS 140-2 certified libraries for our cryptographic functions.

FIPS 140-2 compliant and enabled products

This section contains information on FIPS 140-2 compliant and enabled product definitions, converting EMC products from FIPS 140-2 enabled to FIPS 140-2 compliant status, and custom client applications.

Definitions

EMC Documentum products and applications are either FIPS 140-2 compliant or FIPS 140-2 enabled, according to these definitions:

- A FIPS 140-2 compliant application is one that uses only FIPS 140-2 approved cryptographic algorithms and only a FIPS 140-2 validated implementation of those algorithms. FIPS 140-2 compliant applications require no additional configuration and are enabled to use FIPS-140–2 algorithms upon installation.
- A FIPS 140-2 enabled Java application is one that uses the Java Cryptographic Extension (JCE) and/or Java Secure Sockets Extension (JSSE) APIs for all its cryptographic and SSL functions.

Converting EMC products from FIPS 140-2 enabled to FIPS 140-2 compliant status

A FIPS 140-2 enabled application can be made FIPS 140-2 compliant by deploying and running the application in a Java Runtime Environment (JRE) that has been configured to use FIPS 140-2 validated JCE and JSSE providers.

To configure FIPS-140 compliant mode, you must first install a Java Cryptography Extension (JCE) API provider, such as RSA BSAFE Crypto-J. The following procedure uses the RSA JCE provider only as an example; you can use any compliant provider you want.

1. Install the JCE API implementation, such as RSA BSAFE Crypto-J, on the same computer on which you have installed your Documentum product. Follow the installation instructions of the JCE provider.

   In this procedure, it is assumed that you installed RSA BSAFE Crypto-J in the `$CRYPTOJ_HOME` directory.

2. From the computer on which you installed your Documentum product, open a terminal window as the installation owner.

3. Statically register the Crypto-J JCE provider by copying the `$CRYPTOJ_HOME/cryptoj/lib/cryptojFIPS.jar` JAR file to the `$JAVA_HOME/jre/lib/ext` directory. For example:

   ```
   prompt$ cp $CRYPTOJ_HOME/cryptoj/lib/cryptojFIPS.jar $JAVA_HOME/jre/lib/ext
   ```

4. Edit the `$JAVA_HOME/lib/security/java.security` file as follows:

   a. Configure the Crypto-J JCE provider to be the default provider by adding the following line:

      ```
      security.provider.1=com.rsa.jsafe.provider.JsafeJCE
      ```

      If other security providers are already configured with this property, change their identifying numbers so that they are unique, as shown in the following example:

      ```
      security.provider.1=com.rsa.jsafe.provider.JsafeJCE
      security.provider.2=sun.security.provider.Sun
      ```

   b. Add the following properties as required specifically by the Crypto-J JCE provider:

      ```
      com.rsa.cryptoj.fips140initialmode=FIPS140_MODE
      com.rsa.cryptoj.kat.strategy=on.load
      ```

5. If you are using the evaluation mode of the RSA BSAFE Crypto-J module, install the RSA evaluation license as follows:

   ```
   prompt$ cp $CRYPTOJ_HOME/cryptoj/lib/rsamisc.jar $JAVA_HOME/jre/lib/ext
   ```

Instructions to configure a JRE to use FIPS 140-2 validated JCE and JSSE vary depending on the vendor. Consult vendor documentation for specific instructions and guidelines.

If the Documentum Foundation Classes (DFC) API is used for all cryptographic operations, then the application is FIPS 140-2 enabled and can be made FIPS 140-2 compliant by deploying and running it in a properly configured JRE environment.

Special considerations apply for applications deployed on top of FIPS 140-2 compliant applications. For example, the Content Server application is FIPS 140-2 compliant, but Documentum Administrator, which must be deployed on the Content Server, is only FIPS 140-2 enabled. To ensure that both components are FIPS 140-2 compliant, Documentum Administrator must be deployed and running in a JRE that has been properly configured to use FIPS 140–2 compliant cryptographic algorithms.

The following table lists the EMC Documentum products and their FIPS 140-2 compliant or FIPS 140-2 enabled status:

Table 1. Documentum products, custom client applications, and FIPS 140-2 status

Product	FIPS 140-2 status
Accelerated Content Services (ACS)	Compliant (using Content Server JBoss)

Product	FIPS 140-2 status
Branch Office Caching System (BOCS)	Compliant
Business Activity Monitor (BAM)	Enabled
Connector for Network Appliance	Not using FIPS 140-2 compliant technology
Content Intelligence Services (CIS)	Enabled
Content Management Interoperability Services (CMIS)	Enabled
Content Server (CS)	Compliant
Content Services for Centera	Refer to ATMOS and Centera filestores, page 14 for more information
Content Storage Services	Enabled
Content Transformation Services (CTS)	Enabled
Custom client applications	Refer to Custom client applications, page 12 for more information
Documentum Administrator (DA)	Enabled
Documentum Foundation Classes (DFC)	Enabled (for local CS environment, compliant)
Documentum Foundation Services (DFS)	Enabled
Documentum Imaging Services (DIS)	Enabled
Documentum Messaging Server (DMS)	Compliant (using Content Server JBoss)
High-Volume Server	Compliant
Process Engine (PE)	Compliant
Process Integrator (PI)	Enabled
Thumbnail Server	Compliant
Trusted Content Services (TCS)	Compliant
UCF Client Java	Enabled
UCF Client .NET	Enabled
UCF Server	Enabled
xCelerated Composition Platform (xCP) 2.0	Enabled
xCelerated Management System (xMS)	Enabled[1]
XML Store	Enabled
xPlore	Compliant, with exceptions Refer to Full-text indexing, page 15 for more information

1. xMS uses externally implemented crypto using Java JCE. The JVM is not pre-configured to use RSA BSAFE. The code uses RSA Crypto-J 5.0 FIPS-compliant JARs as the implementation provider of Java JCE. This crypto-J FIPS JAR is set at runtime. Therefore, this is a dynamic configuration of JVM to use an RSA Crypto-J FIPS-compliant JAR that is shipped along with the product.

Custom client applications

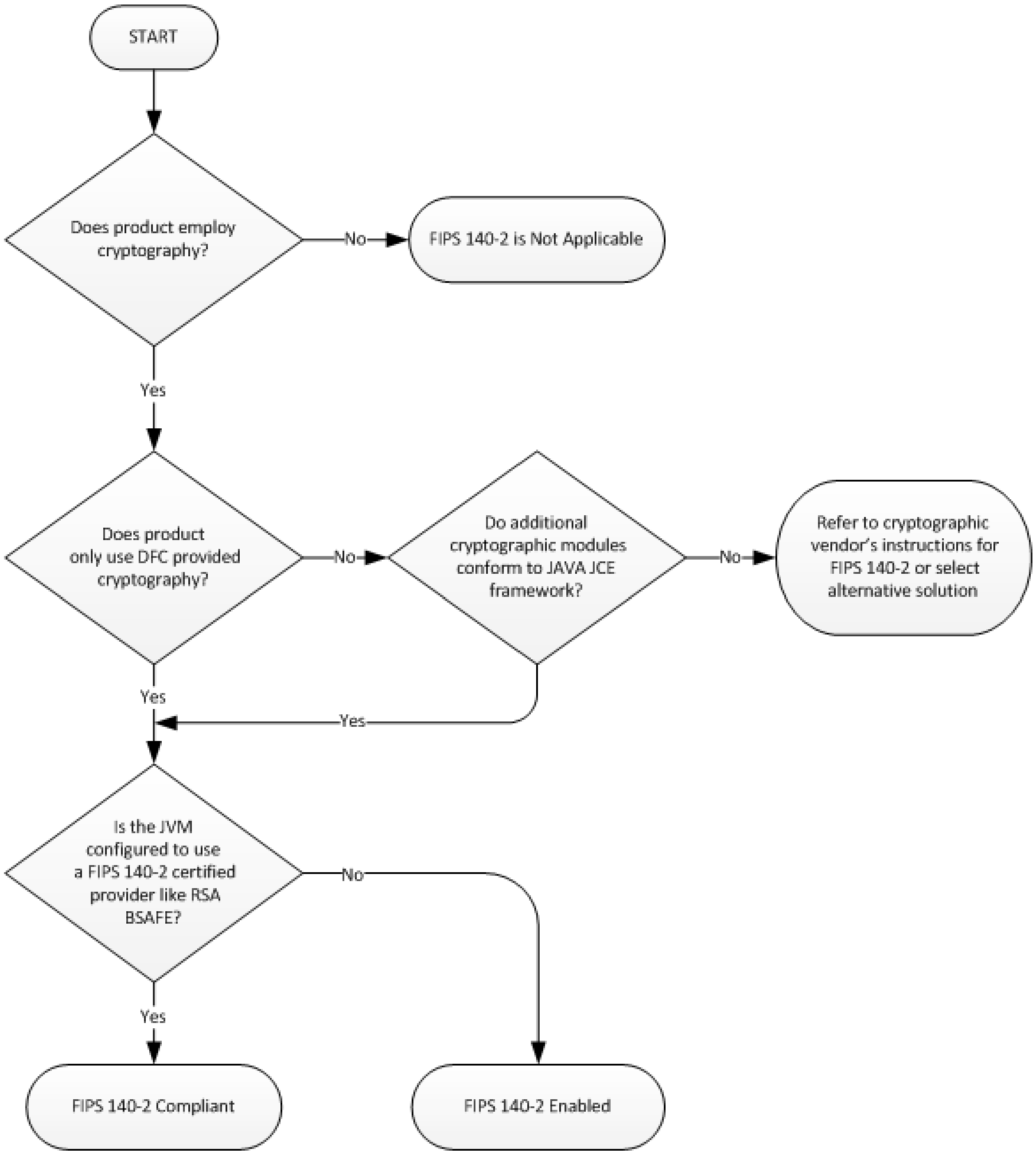
Custom applications built using WDK, DFC, DFS, or any other technologies, could use encryption that does not meet FIPS 140–2 criteria. Therefore, the creator of the custom application is responsible for evaluating how any encryption is used by the application, and whether it meets FIPS 140–2 criteria. Even if all the EMC-provided software meets the criteria, a custom application could still be an exception.

If you are delegating your cryptographic operations (including hash) to DFC, it is safe to say that you are FIPS 140-2 enabled. However, if you are using any cryptographic functions (including HASH) in your portion of the code, you need to do an analysis for FIPS 140-2 readiness.

If custom APIs or applications are used in the system, then it is necessary to work with third-party vendors to replace the cryptographic libraries used with FIPS 140-2 enabled cryptographic algorithms before the application can be made FIPS 140-2 compliant. Before end-to-end FIPS 140-2 compliance can be achieved, an inventory of all products deployed with the Content Server must be made to ensure that there are no independent cryptographic operations that require third-party certification for FIPS 140-2 compliance.

Use the following flowchart to determine the FIPS 140-2 readiness of your custom application.

Figure 1. FIPS 140-2 classification procedure


START
Does product employ cryptography?
No
FIPS 140-2 is Not Applicable
Yes
Does product only use DFC provided cryptography?
No
Do additional cryptographic modules conform to JAVA JCE framework?
No
Refer to cryptographic vendor's instructions for FIPS 140-2 or select alternative solution
Yes
Yes
Is the JVM configured to use a FIPS 140-2 certified provider like RSA BSAFE?
No
Yes
FIPS 140-2 Compliant
FIPS 140-2 Enabled

Remember that product environments and infrastructure can also affect FIPS 140-2 compliance. For example, web applications need to be supported by a FIPS 140-2 compliant application server for end-to-end compliant environments. Refer to the FIPS 140-2 documentation for all your infrastructure components.

FIPS 140-2 modules

The FIPS 140–2 standard does not specify what data must be encrypted or in what circumstances, so a system that uses FIPS 140–2 modules may encrypt only some data, or only for some cases. As a specific example, Content Server can be configured to communicate with a connection broker in native mode (unsecured, non-encrypted) or in secured mode (TLS/SSL encrypted). Only the TLS/SSL mode, and not the native mode, is relevant to FIPS 140–2.

A cryptographic hash algorithm is used in the Documentum system to securely manage and authenticate information by providing a unique hash key for a given unit of information. The Documentum system uses hash keys for signing and validating audit records and esignatures, generating passphrase-based keys, and managing content deduplication, a process that relies on the content hash key to ensure that stored or embedded content is not duplicated in the repository.

Except for those modules listed in the exceptions section of this chapter, all encryption done by Trusted Content Services Content Server uses FIPS 140–2 validated modules.

FIPS 140-2 exceptions

There are some cases where encryption modules are used by Trusted Content Services that do not satisfy FIPS 140–2 criteria. This section identifies those cases.

Pre-7.0 Content Server and applications

Pre-7.0 Content Servers and applications do not use validated FIPS 140-2 modules. A 7.0 release (and later) Content Server can communicate with pre-7.0 release Content Servers, connection brokers, DFC instances, and other pre-7.0 applications, and will do so using validated FIPS 140-2 cryptographic modules. Content encrypted with pre-7.0 Content Servers will be decrypted by a 7.0 Content Server using validated FIPS 140-2 cryptographic modules.

Operating environments

The operating environment (for example, application servers) should also be FIPS 140-2 enabled. For example, if you are deploying your application to a server that uses MD5, you will not be able to achieve FIPS 140-2 compliance through that particular application server flavor. You need to make sure the third parties required by EMC products are FIPS 140-2 compliant/enabled.

ATMOS and Centera filestores

ATMOS and Centera filestores use encryption that is not based on FIPS 140–2 validated cryptographic modules. Neither ATMOS or Centera filestores can be used in a way that meets FIPS 140–2 criteria.

Relational databases

Repositories managed by Content Server use a relational database to store metadata. We link to the vendor library for the databases we support. When we deliver data to the database, we no longer control how it is handled. It may be handled in a way that does not meet FIPS 140–2 criteria, dependent on the configuration of the database. You will need to review your database vendor's instructions for any configuration restrictions or procedures.

Single sign-on (SSO)

The SSO products used by the Content Server use encryption that is not based on FIPS 140–2 validated cryptographic modules. Therefore, the following cannot be used in a way that meets FIPS 140–2:

- Kerberos
- EMC RSA Plugin for Documentum
- CA SiteMinder (Netegrity)

Full-text indexing

The xPlore query plugin uses a TLS/SSL implementation from OpenSSL that is not based on FIPS 140–2 validated cryptographic modules. Therefore, when xPlore is configured to use SSL for communication, it cannot be used in a way that meets FIPS 140–2 criteria.

Electronic signatures

The software for encrypting electronic signatures uses MD5 encryption and is not based on FIPS 140–2 validated modules. Therefore, electronic signatures cannot be used in a way that meets FIPS 140–2 criteria

Chapter 2

Cryptography Used in the Documentum System

The Documentum System uses cryptography in several different contexts. For example, content files in a filestore can be encrypted, communications between Content Server and a connection broker (also referred to as a docbroker) can be protected with TLS/SSL connections, and access keys for different elements of Content Server can be protected cryptographically. Branch Office Caching Services (BOCS) caches content files, and can be configured to encrypt them.

TLS/SSL cryptography

TLS/SSL communication can be used between several Documentum components and between Documentum components and third-party products. This section describes which modules are used for each component when TLS/SSL communication is used. For backward compatibility, TLS/SSL communications with components from releases earlier than release 7.0 use the 3DES algorithm. TLS/SSL communications with components from release 7.0 and later use AES-128.

Content Server TLS/SSL

Content Server uses RSA BSAFE® Micro Edition Suite (BSAFE MES) version 3.2.4 to connect to connection brokers, DFC instances, and LDAP servers when using TLS/SSL. For encryption, BSAFE MES depends on the FIPS 140–2 validated modules, RSA BSAFE® Crypto-C Micro Edition 3.0.0.0 and 3.0.0.1. You cannot specify a non-FIPS 140–2 mode for this connection. The algorithms used are AES-128 and 3DES.

Remote Key Manager TLS/SSL

Content Server uses the RKM C Client Library Version 2.7.1.11 to establish a TLS/SSL connection to the RSA Data Protection Manager.

Connection broker TLS/SSL

Connection broker (docbroker) uses RSA BSAFE Micro Edition Suite (MES) version 3.2.4 to connect to Content Server and to DFC when using TLS/SSL. For encryption, this suite depends on the FIPS 140–2 validated modules, RSA BSAFE Crypto-C Micro Edition 3.0.0.0 and 3.0.0.1. You cannot specify a non-FIPS 140–2 mode for this connection. The algorithms used are AES-128 and 3DES.

Java Method Server TLS/SSL

Java Method Server (JMS) uses the RSA BSAFE Crypto-J 5.0 module for encryption when using TLS/SSL. The algorithms used are AES-128 and 3DES.

JMS is based on the JBoss application server and uses a Java Virtual Machine (JVM). In order for the JMS to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during JMS installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

DFC TLS/SSL

There are two separate types of DFC installations, and they require different handling to use FIPS 140–2 encryption. One type of DFC installation is the embedded DFC installed when Content Server is installed. The other type of DFC installation is a remote installation on a non-Content Server host. Both types of DFC installations rely on a JVM to perform encryption.

Embedded DFC

An instance of DFC is installed during Content Server installation, on the same host as Content Server. This instance of DFC uses the RSA BSAFE Crypto-J 5.0 module for encryption. The algorithms used are AES-128 and 3DES.

This instance of DFC uses a Java Virtual Machine (JVM) installed by Content Server. In order for the JMS to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

Remote DFC

An instance of DFC can be installed on a remote, non-Content Server host. This instance of DFC uses the JVM installed on that host. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by your JVM are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor. EMC recommends the RSA BSAFE SSL-J modules.

DFS TLS/SSL

Documentum Foundation Services (DFS) is installed into an application server, and relies on the JVM installed on that server when using TLS/SSL. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by your application server JVM are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor. EMC recommends the RSA BSAFE SSL-J modules.

Documentum Administrator TLS/SSL

Documentum Administrator is installed into an application server, and relies on the JVM installed on that server when using TLS/SSL. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by your application server JVM are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor. EMC recommends the RSA BSAFE SSL-J modules.

ACS TLS/SSL

Accelerated Content Services (ACS) server uses the RSA BSAFE Crypto-J 5.0 module for encryption when using TLS/SSL. The algorithms used are AES-128 and 3DES.

ACS is installed on the JBoss application server and uses a Java Virtual Machine (JVM). In order for the ACS server to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

UCF Server TLS/SSL

A UCF server uses the RSA BSAFE Crypto-J 5.0 module for encryption when using TLS/SSL. You cannot specify a non-FIPS 140–2 mode for this connection. The algorithms used are AES-128 and 3DES.

UCF client TLS/SSL

There are two types of UCF clients: Java and .NET.

Java UCF client

A Java UCF client relies on the JVM installed on the end user's host when using TLS/SSL. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by that JVM are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor. EMC recommends the RSA BSAFE SSL-J modules.

.NET UCF client

A .NET UCF client relies on the .NET runtime installed on the end user's host when using TLS/SSL. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by that .NET runtime are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor, Microsoft. The *Microsoft documentation* contains more information.

BOCS TLS/SSL

A Branch Office Caching Services (BOCS) server uses the RSA BSAFE Crypto-J 5.0 module for encryption when using TLS/SSL. You cannot specify a non-FIPS 140–2 mode for this connection. The algorithms used are AES-128 and 3DES.

BOCS is installed on a JBoss application server and uses a Java Virtual Machine (JVM). In order for the BOCS server to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

DMS TLS/SSL

Documentum Messaging Service (DMS) uses the RSA BSAFE Crypto-J 5.0 module for encryption when using TLS/SSL. The algorithms used are AES-128 and 3DES.

DMS uses a private Java Virtual Machine (JVM) installed by DMS. In order for the DMS to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

CMIS TLS/SSL

Content Management Interoperability Services (CMIS) is installed into an application server, and relies on the JVM installed on that server when using TLS/SSL. To operate in FIPS 140–2 compliant mode you must insure that the cryptography modules used by your application server JVM are FIPS 140–2 validated. Install the required modules according to the instructions provided by the module vendor. EMC recommends the RSA BSAFE SSL-J modules.

Key generation

Several symmetric keys are used for encrypting and for signing data:

- Application Encryption Key (AEK)
- DocBase Key (DBK)
- Login Ticket Key (LTK)
- File Store Key (FSK)
- File Encryption Key (FEK)

For systems that do not use remote key management, all these keys are generated by Content Server. For systems that do use remote key management, only the AEK is generated by Content Server, and the rest are generated and managed by the remote key server. The AEK is the root key for the system.

Content Server keys

Content Server uses RSA BSAFE Micro Edition Suite (BSAFE MES) version 3.2.4 to generate and encrypt the keys. For this, BSAFE MES depends on the FIPS 140–2 validated modules, RSA BSAFE Crypto-C Micro Edition 3.0.0.0 and 3.0.0.1. You cannot specify a non-FIPS 140–2 mode for this encryption. The algorithms used are AES-128, AES-192, AES-256, and 3DES. 3DES is used only to decrypt keys for the releases prior to 7.0.

For systems that use remote key management, the keys are generated and protected in RSA Data Protection Manager. Please see the RSA documentation for encryption modules used by Data Protection Manager.

DFC keys

DFC uses the AEK for encryption and decryption of data independently from a repository. A DFC instance can be either an embedded or remote DFC. The details are described in DFC TLS/SSL, page 18.

File encryption

Content files can be protected by encryption when in transit, using TLS/SSL. These files can also be protected at rest by using encryption either in a filestore, or in a file cache in a BOCS server.

Filestore encryption

Content files are stored in a file store. Before the files are stored, Content Server encrypts them using the file encryption key (FEK), The FEK itself is encrypted using the file store key (FSK). The encrypted FEK is stored in the content header of the encrypted file, and the file is then stored.

Content Server uses RSA BSAFE Micro Edition Suite (BSAFE MES) version 3.2.4 to encrypt the files and keys. For this, BSAFE MES depends on the FIPS 140–2 validated modules, RSA BSAFE Crypto-C Micro Edition 3.0.0.0 and 3.0.0.1. You cannot specify a non-FIPS 140–2 mode for this encryption. The algorithms used are AES-128, AES-192, AES-256, and 3DES. 3DES is used only to decrypt keys for the releases prior to 7.0. For the 7.0 and 7.1 releases, AES-128 is used. For the 7.2 release and later, AES-256 is used.

BOCS encryption

A Branch Office Caching Services (BOCS) server uses the RSA BSAFE Crypto-J 5.0 module for encryption for cached content files. You cannot specify a non-FIPS 140–2 mode for this encryption. The algorithm used is AES-256.

BOCS is installed on a JBoss application server and uses a Java Virtual Machine (JVM). In order for the BOCS server to use only FIPS 140–2 validated encryption modules, we replace the Java supplied encryption libraries and Java security policy files with RSA BSAFE Crypto-J 5.0 libraries and required policy files during installation. As installed, you cannot specify a non-FIPS 140–2 mode for this connection. However, an administrator can change the Java security policy files, and install non-FIPS 140–2 Java encryption libraries.

Chapter 3

Module Certificates

The following is a list of the FIPS 140–2 validated modules used in Documentum products. The links point to the validation details including links to the PDF documents of the actual certificates. All other RSA products that we use (Micro Edition Suite 3.2.4, CertJ 5.2, and SSLJ 5.1.1.1) are built on top of these modules and do not contain any cryptographic code and so do not need validation.

- **RSA BSAFE Crypto-J 5.0** (Java): For Certificate 1502 and Certificate 1503, the *National Institute of Standards and Technology* website contains more information.
- **RSA BSAFE Crypto-C Micro Edition 3.0.0.0** (32-bit: AIX, HPUX, Solaris. 64-bit: AIX, HPUX, Solaris, Linux, Windows): For Certificate 1058: the *National Institute of Standards and Technology* website contains more information.
- **RSA BSAFE Crypto-C Micro Edition 3.0.0.1** (32-bit: UNIX, Windows): For Certificate 1092, the *National Institute of Standards and Technology* website contains more information