Adobe® Flash® Access™

Version 3.0

August 2011

Secure Deployment Guidelines

© 2010 Adobe Systems Incorporated. All rights reserved.

Adobe® Flash® Access™ 3.0 Secure Deployment Guidelines

This guide is protected under copyright law, furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the user guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the user guide; and (2) any reuse or distribution of the user guide contains a notice that use of the user guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/

Adobe, the Adobe logo, Adobe AIR, and Flash Access are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

Updated Information/Additional Third Party Code Information available at http://www.adobe.com/go/thirdparty.

Portions include software under the following terms:

This product contains either BSAFE and/or TIPEM software by RSA Security Inc.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Last updated 8/27/2011

Contents

Chapter 1: Network topology
Network layer security 1
Firewall rules 2
Network protocols used by Flash Access 3
Ports for application servers 3
Configuring SSL 4

Chapter 2: Vendor-specific security information
Operating system security information 5
Application server security information 6

Chapter 3: Physical security and access
Overview 7

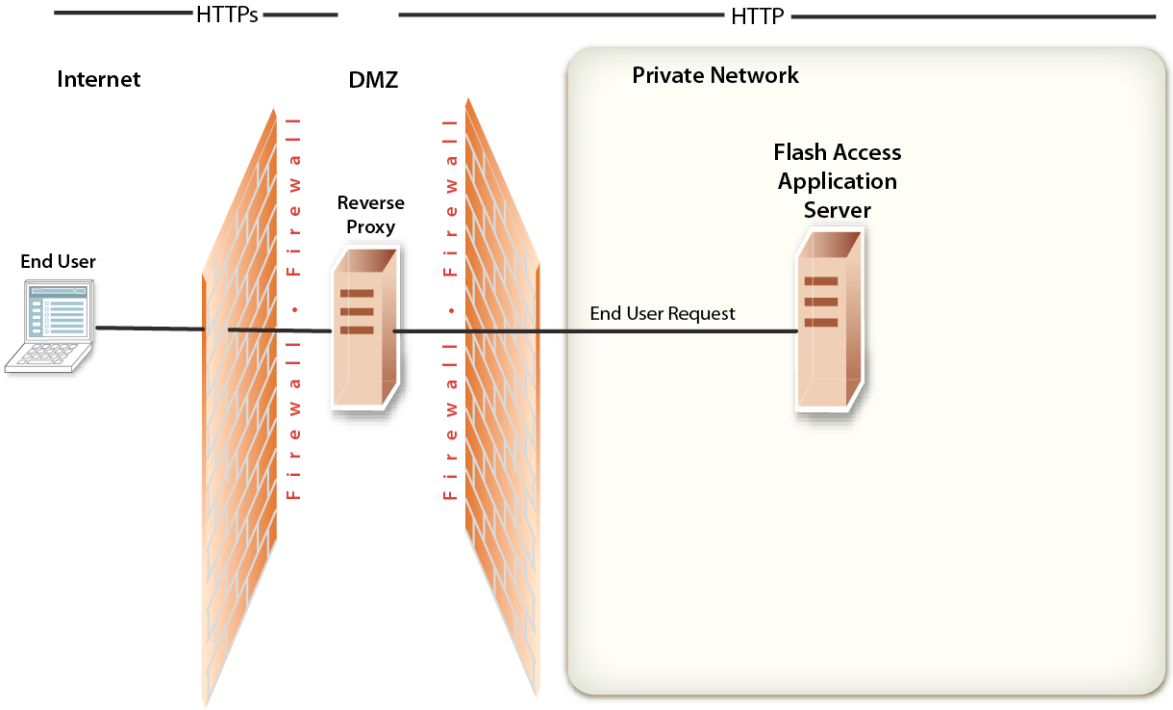
Chapter 4: Packaging and protecting content
Securing the server 8
Securely packaging content 8
Securely storing policies 9
Asymmetric key encryption 9
Ensuring compatibility with Flash Media Rights Management Server 1.x 9

Chapter 5: Issuing licenses
Protecting the License Server 11
Pre-generating licenses 13
Managing Domains 13

Chapter 1: Network topology

After you successfully deploy Flash Access, it is important to maintain the security of your environment. This section describes the tasks that are necessary to maintain the security of your Flash Access production server.

Use a *reverse proxy* to ensure that different sets of URLs for Flash Access web applications are available to both external and internal users. This configuration is more secure than allowing users to connect directly to the application server on which Flash Access is running. The reverse proxy performs all HTTP requests for the application server that is running Flash Access. Users only have network access to the reverse proxy and can attempt only the URL connections that are supported by the reverse proxy.



Network layer security

Network security vulnerabilities are among the first threats to any Internet-facing or intranet-facing application server. This section describes the process of hardening hosts on the network against these vulnerabilities. It addresses network segmentation, Transmission Control Protocol/Internet Protocol (TCP/IP) stack hardening, and the use of firewalls for host protection.

This table describes common techniques that reduce network security vulnerabilities.

Technique	Description
Demilitarized zones (DMZs)	Segmentation must exist in at least two levels with the application server used to run Flash Access placed behind the inner firewall. Separate the external network from the DMZ that contains the web servers, which in turn must be separated from the internal network. Use firewalls to implement the layers of separation. Categorize and control the traffic that passes through each network layer to ensure that only the absolute minimum of required data is allowed.
Private IP addresses	Use Network Address Translation (NAT) with RFC 1918 private IP addresses on Flash Access application servers. Assign private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) to make it more difficult for an attacker to route traffic to and from a NAT internal host through the Internet.
Firewalls	Use the following criteria to select a firewall solution: • Implement firewalls that support proxy servers and/or stateful inspection instead of simple packet-filtering solutions. • Use a firewall that supports a security paradigm in which you can deny all services except those explicitly permitted. • Implement a firewall solution that is dual-homed or multi-homed. This architecture provides the greatest level of security and helps to prevent unauthorized users from bypassing the firewall security.

Firewall rules

Incoming URLs

Configure your outer firewall so that it exposes only the URLs for application functionality that you want to provide to end users. Allow external users access through the outer firewall only to the URLs listed in the following table:

Root URL	Purpose
/flashaccess/getServerVersion/v3	URL for determining the server version.
/flashaccess/authn/v1/* /flashaccess/authn/v3/*	URLs for user authentication. This URL must be accessible only if you use Flash Access Client APIs to perform user authentication.
/flashaccess/license/v1/* /flashaccess/license/v3/*	URLs for issuing licenses to end users.
/flashaccess/sync/v3	URL for synchronization requests. This URL must be accessible only if you specify the synchronization requirements in your licenses.
/flashaccess/domain/v3	URL for domain registration. This URL must be accessible only if you implement domain support.

Root URL	Purpose
/flashaccess/dereg/v3	URL for domain de-registration. This URL must be accessible only if you implement the domain support.
/flashaccess/headerconversion/v1/* /flashaccess/headerconversion/v3/*	URLs for use by the client to convert FMRMS 1.x DRM metadata to Flash Access DRM metadata. Note: *This URL must use SSL (HTTPS)*.
/edcws/services/urn:EDCLicenseService/*	LiveCycle Rights Management ES web service URL. If content was published using an earlier version of FMRMS, this URL allows older clients to connect to the server and be prompted to upgrade to Flash Access 3.0. Note: *This URL must use SSL (HTTPS)*.

Note: *The internal firewall must only allow connections to be made to the Flash Access license server through the reverse proxy, and only to the URLs listed above. To improve scalability, the connections between the reverse proxy and Flash Access can be over HTTP.*

Outgoing URLs

The license server requires access through the firewall to download the following CRLs from Adobe:

- http://crl2.adobe.com/Adobe/FlashAccessRootCA.crl
- http://crl2.adobe.com/Adobe/FlashAccessIntermediateCA.crl
- http://crl3.adobe.com/AdobeSystemsIncorporatedFlashAccessRuntime/LatestCRL.crl
- http://crl2.adobe.com/Adobe/FlashAccessIndividualizationCA.crl

Network protocols used by Flash Access

When you configure a secure network architecture, the network protocols in the following table are required for interaction between Flash Access and other systems in your enterprise network.

Protocol	Use
HTTP	Flash Player and Adobe AIR® communicate with Flash Access over HTTP.
HTTPS (Optional)	Flash Player and Adobe AIR can use HTTPS for communication with Flash Access, however, HTTPS (SSL) is not required unless you need support for FMRMS 1.x clients. See the notes in the table "Incoming URLs" and "Configuring SSL".

Ports for application servers

You can configure the Flash Access license server to use any network port. These ports must be enabled or disabled on the inner firewall, depending on the network functionality you want to allow for clients that connect to the application server that is running Flash Access.

Configuring SSL

Secure Sockets Layer (SSL) is only necessary if you require support for Flash Media Rights Management Server 1.x clients.

Chapter 2: Vendor-specific security information

This section contains security-related information about operating systems and application servers that are incorporated into your Flash Access solution.

Use the links provided in this section to find vendor-specific security information for your operating system and application server.

Operating system security information

When securing your operating system, carefully implement the measures that are described by your operating system vendor, including these:

- Defining and controlling users, roles, and privileges
- Monitoring logs and audit trails
- Removing unnecessary services and applications
- Backing up files

For security information about operating systems that Flash Access supports, see the resources in this table.

Operating System	Security Resource
Microsoft® Windows Server® 2008 Enterprise or Standard Edition	*Windows Server 2008 Security Guide*
Red Hat® Enterprise Linux® 5.4, 5.5, and 5.6.	*Red Hat Enterprise Linux 5 Security Guide*

The following table describes some potential approaches to minimizing security vulnerabilities that are found in the operating system.

Item	Description
Security patches	There is an increased risk that an unauthorized user may gain access to the application server if vendor security patches and upgrades are not applied in a timely fashion. Test security patches before you apply them to production servers. Also, create policies and procedures to check for and install patches on a regular basis.
Virus protection software	Virus scanners can identify infected files by scanning for a signature or watching for unusual behavior. Scanners keep their virus signatures in a file, which is usually stored on the local hard drive. Because new viruses are discovered often, you must frequently update this file in order for the virus scanner to identify all current viruses.
Network Time Protocol (NTP)	For both proper operation and forensic analysis, keep accurate time on Flash Access servers and Flash Access packagers. Use a secure version of NTP to synchronize the time on all systems that are connected to the Internet.

Application server security information

When securing your application server, you must implement the measures that are described by your server vendor, including these:

- Using non-obvious administrator user name
- Disabling unnecessary services
- Securing the console manager
- Enabling secure cookies
- Closing unneeded ports
- Limiting administrative interfaces by IP addresses or domains
- Using the Java™ Security Manager

Chapter 3: Physical security and access

The physical security for your Flash Access environment can range from the server being placed in a secure room under lock and key, along with other equipment, to being in a secured alarmed cage, isolated from other computers, with two-part authentication such as badge and fingerprint, monitored continuously by Closed Circuit Television (CCTV). The degree of security that you implement depends on your organizations' policies, the risk involved (potential of loss and severity if lost), and other legal compliance requirements.

Overview

As a general rule, it is recommended that your Flash Access environment be located in a secure server room where access is electronically controlled (with a card reader at a minimum), alarmed and monitored by security or someone who will respond rapidly to any breach or incident, with access continuously recorded on CCTV. The goal of this security recommendation is for your security team to know who is in the room and when they entered. If the server room is large and has a long list of people with access, the server must also be in a secure cage or rack to limit further access.

To implement an extremely high level of security, the protection must extend to the power supply, uninterruptable power supply (UPS), network equipment, and other related equipment. Any disruption to these items affects the server, especially if it must be up and running at all times. Access must be two-part, such as badge and PIN or badge and fingerprint. Also, anti-tailgating devices must be installed on the door to prevent authorized people from bringing in unauthorized people with them.

For general information and recommendations about physical security standards, see the ISO FAQs site.

Chapter 4: Packaging and protecting content

The information in this chapter will help you protect your content.

Securing the server

You need to physically secure the computer on which policy management and content packaging occurs. (See "Physical security and access".)

In addition, if your content packaging implementation requires network connectivity, you must harden your operating system and implement an appropriate firewall solution. For information, see "Network topology" on page 1.

Securely packaging content

The configuration file for the Flash Access Media Packager command line tool requires a PKCS12 credential that is used during packaging.

In the Reference Implementation Command Line tools, the password for the PKCS12 credentials file is stored in the flashaccess.properties file in clear text. For this reason, take extra care when securing the computer hosting this file, and ensure that it is in a secure environment. (See "Physical security and access" on page 7.)

The packager also uses the License Server and License Server Transport certificates. Both the integrity and confidentiality of this information must be protected. Only authorized entities should be permitted to use the packager. If any of your private keys are compromised, immediately inform Adobe Systems Incorporated so that the certificate can be revoked.

Note: *The API allows you to use the same key for multiple pieces of content. To ensure the highest level of security, we recommend this feature only be used for multi-bit rate FMS content. It is not recommended to use the same key for multiple files representing different content.*

The Flash Access Packaging API issues warnings under certain conditions. You must review these warnings to determine whether your files have been successfully encrypted. The warning messages may state that the policy has expired, an unrecognized tag or track will not be encrypted, movie fragments will not be encrypted (and references inside those fragments may become invalid), and metadata will not be encrypted.

If content is packaged using a policy with incorrect attributes, the policy should be updated, and the updated policy must be made available to the License Server though a policy update list or another mechanism for delivering the updated policy to the server. Some policy attributes cannot be changed after the policy is created. If these attributes are incorrect, pull the content back from the distribution sites, revoke the policy so that no future licenses can be granted, and re-encrypt the content.

When packaging is complete, the packaging key is not explicitly destroyed; however, it is garbage-collected. Therefore, the packaging key does remain present in memory for a period of time; you must guard against unauthorized access to the machine and take steps to ensure that you do not expose any files, such as core dumps, that may reveal this information.

Securely storing policies

Flash Access SDK provides a great deal of flexibility in the development of applications for use in content packaging and policy creation. When creating such applications, you may want to allow some users to create and modify policies, and limit other users such that they can only apply existing policies to content. If this is the case, you must implement the necessary access controls to create user accounts with different privileges for policy creation and the application of policies to content.

Policies are not signed or otherwise protected from modification until they are used in packaging. If you are concerned about users of your packaging tools modifying policies, you should consider signing the policies to ensure that they cannot be modified.

For more information on creating applications using the SDK, see the *Flash Access API Reference*.

Asymmetric key encryption

Asymmetric key encryption (also called public key encryption) uses pairs of keys. One key is used for encryption; the other for decryption. The decryption key is kept secret, and is referred to as a *private key*. The encryption key, referred to as the *public key*, is made available to anyone authorized to encrypt content. Anyone with access to the public key is able to encrypt content, but only someone with access to the private key can decrypt the content. The private key cannot be reconstructed from the public key.

When packaging content, the License Server's public key is used to encrypt the content encryption key (CEK) in the DRM metadata. You must ensure that only the License Server has access to the License Server's private key; if someone else has the key they can decrypt and view the content.

Caution: *Ensure that you obtain the License Server's certificate (containing the public key) from a trusted source so that you can be certain it is the License Server's key, and not a rogue public key. If an attacker were to substitute their public key for the License Server's key, they would be able to decrypt your content.*

For more information on packaging content, see *Flash Access: Protecting Content*.

Ensuring compatibility with Flash Media Rights Management Server 1.x

Flash Media Rights Management Server 1.x and Flash Access use different metadata for packaging content and requesting licenses. For Flash Access to use FMRMS version 1.x content, the metadata must be converted.

The Flash Access SDK supports two options for converting metadata:

- Offline (recommended)

  Generate the Flash Access metadata in an offline process and store it for use when a client requests it. Adobe recommends this option. If metadata is generated offline, the license server does not need access to the 1.x CEKs or the packager credential. In addition, converting offline offers better performance because the License Server doesn't need to generate the metadata in real time.

- On-demand

Generate the Flash Access metadata on-demand when a client requests it. When a Flash Access client downloads version 1.x content, it sends the DRM metadata (also known as the DRM header) to the Flash Access server. The server converts the DRM metadata to the current format. The server encrypts and embeds the content encryption key (CEK) in the metadata and sends the content back to the Flash Access client. (The Flash Access 1.x metadata does not contain the CEK.)

To convert the metadata, the Flash Access server requires access to the Flash Access 1.x content encryption keys. When you migrate from Flash Media Rights Management Server 1.x, you can continue to store the content encryption keys in the LiveCycle ES database, or you can implement a custom solution to securely store the content encryption keys elsewhere. If you choose to continue storing the content encryption keys in the LiveCycle ES database, follow the recommendations outlined in "Protecting access to sensitive content in the database" in *Hardening and Security for LiveCycle ES*.

For more information on ensuring compatibility with content packaged using Flash Media Rights Management Server 1.x, see the *Flash Access API Reference*.

Chapter 5: Issuing licenses

The information in this chapter will help you securely issue licenses.

Protecting the License Server

The following sections discuss best practices to protect the License Server.

Consuming locally generated CRLs

To consume locally generated certificate revocation lists (CRLs) and policy update lists, use Flash Access APIs to verify the signature. The APIs verify that the lists have not been tampered with and that they were signed by the correct License Server.

- Call `RevocationList.verifySignature` to check the signature before providing the RevocationList to any APIs.

  For more information, see `RevocationListFactory` in the *Flash Access API Reference*.
- Call `PolicyUpdateList.verifySignature` to check the signature before providing the PolicyUpdateList to any APIs.

  For more information, see `PolicyUpdateList` in the *Flash Access API Reference*.

Consuming CRLs published by Adobe

The SDK periodically downloads CRLs published by Adobe. Do not block access to these files or prevent enforcement of these CRLs.

The SDK has a configuration option to ignore errors when retrieving the Adobe CRLs. This option may only be used in development environments. In production environments, the license server must be able to retrieve the CRLs from Adobe. Failure to obtain a valid CRL is an error.

Generating CRLs to supplement those published by Adobe

Flash Access lets you create CRLs to supplement the machine CRL published by Adobe. Flash Access SDK checks and enforces the Adobe CRLs, however, you can disallow additional client machines by creating a CRL that revokes additional machine credentials. To do this, you must pass the CRL to Flash Access SDK, then, when issuing a license, the SDK checks both the Adobe CRL and your own CRL.

To learn more about generating CRLs, see `RevocationListFactory` in *Flash Access API Reference*.

Rollback detection

If your implementation of Flash Access uses business rules that require the client to maintain state (for example, the playback window interval), Adobe highly recommends that the server keep track of the rollback counter and use AIR or SWF whitelisting.

The rollback counter is sent to the server in most requests from the client. If your implementation of Flash Access does not require the rollback counter, it can be ignored. Otherwise, Adobe recommends that the server store the random machine ID—obtained using `MachineToken.getMachineId().getUniqueId()`—and the current counter value in a database. For more information on incrementing and tracking the rollback counter, see ClientState in *Flash Access API Reference* and *Rollback detection* in *Using the Flash Access Server for Protecting Content*.

Machine count when issuing licenses

If the business rules require that the number of machines for a user be tracked, the License Server or Domain Server must store the machine IDs associated with the user. The most robust way to track machine IDs is to store the value returned by the `MachineId.getBytes()` method in a database. When a new request comes in, compare the machine ID in the request against the known machine IDs using `MachineId.matches()`.

`MachineId.matches()` performs a comparison of IDs to determine if they represent the same machine. This comparison is only practical if there is a small number of machine IDs to compare against. For example, if a user is allowed five machines within their domain, you can search the database for the machine IDs associated with the user's username and obtain a small set of data to compare against.

Note: *This comparison is not practical for deployments allowing anonymous access. In such cases* `MachineId.getUniqueID()` *can be used, however, this ID will not be the same if the user accesses content from both Flash and Adobe AIR® runtimes, and will not survive if the user reformats their hard drive.*

To learn more about `MachineToken.getMachineId()` and `MachineId.matches()`, see the *Flash Access API Reference*.

Replay protection

Replay protection prevents an attacker from replaying a license request message and potentially causing a denial-of-service (DoS) attack against the client (A *denial-of-service* attack is an attempt by attackers to prevent legitimate users of a service from using that service.). For example, a replay attack using the rollback counter could be used to "trick" the License Server into thinking that the DRM client is rolling back its state, causing a suspension of the account.

To learn more about replay protection, see `AbstractRequestMessage.getMessageId()` the *Flash Access API Reference*.

Maintain a whitelist of trusted content packagers

A *whitelist* is a list of trusted entities. In the case of content packagers, these are organizations trusted by the content owner to package (or encrypt) the FLV/F4V video files, creating DRM protected content. When deploying Flash Access, it is recommended that you maintain a whitelist of trusted content packagers, and that you verify the identity of the content packager contained in the DRM metadata (the DRM header) of a DRM protected file prior to issuing a license.

To learn more about obtaining information about the entity that packaged the content, see `V2ContentMetaData.getPackagerInfo()` in the *Flash Access API Reference*.

Timeout for authentication tokens

All authentication tokens generated by the Flash Access SDK have a timeout interval to protect application security. The expiration for the authentication token is specified use the Flash Access SDK when handling an authentication request. Once the expiration has passed, the authentication token is no longer valid and the user must re-authenticate with the license server.

To learn more about authentication requests, see AuthenticationHandler in the *Flash Access API Reference*.

Overriding policy options

When issuing a license, it is possible for the license server to override the usage rules specified in the policy. If the policy specifies a start date, a license is not generated before that start date. However, it is possible to set a future start date in the license after the license is generated. This option should be used with caution, as the client will not prevent the user from moving the system time forward to circumvent the start date.

Pre-generating licenses

If you are pre-generating licenses that contain time-based usage rules, it is highly recommended that the license includes synchronization requirements (See '*Using the Flash Access Server for Protecting Content*' guide), so the license expiration can be enforced securely. Implementing a 'heart beat' mechanism between the client and the server is highly recommended if you have any time-based restrictions in the license, since the heart beat will synchronize the client time with the server time.

Managing Domains

To prevent users from being able to backup and restore their files in an effort to bypass domain de-registration, it is recommended that one of the following approaches be implemented for domain management:

- Limit the amount of time the domain credentials are valid. Clients will need to contact the domain server to re-acquire domain credentials when they expire. At that time, the Domain Server can ensure that the machine is still authorized to be a member of the domain.
- Rollover the domain keys each time a user de-registers. The License Server should only issue licenses to clients that have the latest domain key. This assumes that the License Server can co-ordinate with the Domain Server to know which key is the latest. Rolling over the domain keys involves generating a new key pair for the domain. When rolling over the keys for a particular domain, be sure to increment the key version in `generateDomainCredential`. For more information on implementing a key rollover, see *RefImplDomainReqHandler* in the Reference Implementation.
- If the domain server is the same as the license server, the server can use the rollback counter to detect backup and restore. See *Processing Flash Access requests* in *Using the Flash Access Server for Protecting Content.*