

Digitally Signed Cisco Software

The Digitally Signed Cisco Software feature describes how to identify digitally signed Cisco software, gather software authentication information related to digitally signed images, and perform key revocation. Digitally Signed Cisco software is software that is digitally signed using secure asymmetrical (public-key) cryptography.

The purpose of digitally signed Cisco software is to ensure that customers are confident that the software running within their systems is secure and has not been tampered with, and that the software running in those systems originated from the trusted source as claimed.

For customers concerned about software updates involving digitally signed Cisco software--no action is necessary for customers to take advantage of the increased protection. The system operation is largely transparent to existing practices. Some minor changes in system displays reflect the use of digitally signed Cisco software.

- Restrictions for Digitally Signed Cisco Software, on page 1
- Information About Digitally Signed Cisco Software, on page 1
- How to Work with Digitally Signed Cisco Software Images, on page 5
- Configuration Examples for Digitally Signed Cisco Software, on page 8
- Additional References, on page 12
- Feature Information for Digitally Signed Cisco Software, on page 13

Restrictions for Digitally Signed Cisco Software

The Cisco Catalyst 4500 E+Series switches running Cisco IOS XE software include the functionality described in this document, except for Digitally Signed Software Key Revocation and Replacement.

Information About Digitally Signed Cisco Software

Features and Benefits of Digitally Signed Cisco Software

Three main factors drive digitally signed Cisco software and software integrity verification:

- The U.S. government is introducing a new version of the Federal Information Processing Standard (FIPS) 140. FIPS-140-3 is the latest draft and is scheduled for ratification in 2010 and to be effective in 2011. This standard requires software to be digitally signed and to be verified for authenticity and integrity prior to load and execution.

- The focus on product security provides increased protection from attacks and threats to Cisco products. Digitally signed Cisco software offers increased protection from the installation and loading of software that has been corrupted or modified.
- Digitally signed Cisco software provides counterfeit protection, which provides further assurance for customers that the equipment they purchase is as claimed.

Digitally Signed Cisco Software Identification

Digitally signed Cisco IOS software is identified by a three-character extension in the image name. The Cisco software build process creates a Cisco IOS image file that contains a file extension based on the signing key that was used to sign images. These file extensions are:

- .SPA
- .SSA

The significance of each character in the file extension is explained in the table below.

Table 1: Digitally Signed Cisco Software Images File Extension Character Meanings

File Extension Character	Character Meaning
S (first character)	Stands for digitally signed software.
P or S (second character)	P and S stand for a production and special (development) image, respectively. A production image is Cisco software approved for general release; a special image is development software provided under special conditions for limited use.
A (third character)	Indicates the key version used to digitally sign the image. A key version is identified by an alphabetical character - for example, A,B,C...

Digitally Signed Cisco Software Key Types and Versions

Digitally signed Cisco software keys are identified by the type and version of the key. A key can be a special, production, or rollover key type. Special and production keys can be revoked. A rollover key is used to revoke a production or special key. The second character in the file extension indicates whether the key type is a special or production key. The key type can be “P” for a production key or an “S” for a special key.

Production and special key types have an associated key version. The key version is defined by the third character in the file extension, in the form of an alphabetical character; for example A, B or C. When a key is replaced, the key version is incremented alphabetically. For example, after a key revocation of a key type “P” (production key) with a key version of “A”, the new image will be signed with key version “B”. Key type and key version are stored as part of the key record in the key storage of the device.

Digitally Signed Cisco Software Key Revocation and Replacement



Note Key revocation and replacement is not supported on Catalyst 4500 E+Series switches running IOS XE software.

Key Revocation

Key revocation is the process of removing a key from operational use in digitally signed Cisco software.

Key revocation takes place when a key becomes compromised or is no longer used. Key revocation and replacement is only necessary in the event of a certain type of vulnerability or catastrophic loss to Cisco's secure key infrastructure. Operational steps to remedy the situation would only be necessary if notified and directed by Cisco. Notification and direction would occur through posting of advisories or field notices on www.cisco.com.

There are two different key revocation processes depending on the type of key to be revoked:

- Production key replacement uses a revocation image and a production image
- Special key replacement uses a production image

Key Replacement

Key replacement is the process of providing a new key to replace a compromised key. The new key is added before the compromised key is revoked. Key replacement is a two-step process:

1. A new key is added to the key storage to replace the revoked key.
2. After the image is verified as operating correctly with the new key, the compromised key is revoked from the key storage.

Key Revocation Image

A revocation image is a basic version of the normal image whose function is to add a new production key to the key storage area. A revocation image has no other capabilities. When a key is to be revoked and replaced, one revocation image per key is provided.

A revocation image contains a new production key bundled within it.

A rollover key stored on the platform is used to verify the signature of the revocation image--a valid revocation image is signed using the same rollover key.



Note A revocation image can be used only in production key revocation.

Important Tasks Concerning the Revocation Image

There are two important tasks concerning the revocation image:

- Adding the new production key to the key storage area.
- Performing a production key upgrade check. For more information, see Step 2 in the “Production Key Revocation”.

Adding the New Production Key to the Key Storage Area:

The revocation image adds the bundled production key to the key storage. The key is written to the primary and backup key storage areas after the revocation image checks that the key is already not part of the existing set of keys in the key storage.

Performing a Key Upgrade Check:

After the new key is added and the customer has upgraded the software (Cisco IOS and ROMmon), the show software authenticity upgrade-status command should be run. The user can review the command output to determine if the production key is successfully upgraded, and can be selected for the next boot.

Production Key Revocation

A production key (also called the release key) is revoked and replaced using a revocation image signed with a rollover key, because the images signed using the compromised production key cannot be trusted. The ROMmon can boot any image signed using a rollover key. The production key revocation and replacement process involves four steps:

1. Add the new production key to the key storage. The new production key is bundled within the revocation image.
2. Perform a software upgrade check using the show software authenticity upgrade-status command to verify the following:
   - The new production key version is installed.
   - The new production key is added to the primary key storage (if not, issue the software authenticity key add production command again with the existing revocation image).
   - The new production key is added to the backup key storage (if not, issue the software authenticity key add production command again with the existing revocation image).
   - The image is configured for autoboot (with the boot system command) signed with the new production key (if not, make sure the new production image is copied into the box and modify the boot system command to point to the new image).
   - The upgradable ROMmon is signed with the new production key (if not, upgrade the ROMmon to the one signed with the new production key).
3. Once everything is verified, the user may load the production image signed with the new production key by using the reload command.
4. Once the new production image is loaded, the user may revoke the compromised key using the software authenticity key revoke production command.

Steps 1 and 2 are done using the special revocation image. It is important for the user to do verifications in Step 2 because after a reboot (in Step 3), an old key will not be revoked if any of the software is still using the old key. The verifications help to ensure that the new key is fully installed and the next reboot (in Step 3) will use the new release software and new ROMmon. Revoking the old production key (Step 4) can be done only after the new key and the new software are installed to the system.

Special Key Revocation

A special key is revoked using a production image signed with a production key. Each production image used for special key revocation has a bundled special key that is the latest at the time of building the production image. The special key revocation and replacement process involves three steps:

1. Add the bundled new special key to the key storage area.
2. Upgrade the ROMmon that is signed using the compromise special key, to the new ROMmon signed with the new special key.
3. Revoke the compromised key from the key storage.

Note that Step 3 does not require any reboot and will be done using the production image itself. This is because the customer is already running a production image and invalidation itself happens from the running production image. Special images do not have the capability to add or invalidate any key.

How to Work with Digitally Signed Cisco Software Images

Identifying Digitally Signed Cisco Software

Perform this task to identify digitally signed Cisco software by examining the image filename in the command output from the show version command, and judging it on the criteria described in the "Digitally Signed Cisco Software Identification" section.



Note If the image file has been renamed by the user, it may not be possible to identify the image because the user may have overwritten the criteria used to indicate that the image is digitally signed.

SUMMARY STEPS

1. **enable**
2. **show version**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show version** **Example:** `Device# show version`	Displays information about the Cisco IOS software version running on a routing device, the ROM Monitor and Bootflash software versions, and the hardware configuration, including the amount of system memory.

Displaying Digitally Signed Cisco Software Signature Information

Perform this task to display information related to software authentication for the current ROMmon and the Cisco IOS image file used for booting. The display includes image credential information, the key type used for verification, signature information, and other attributes in the signature envelope.

SUMMARY STEPS

1. **enable**
2. **show software authenticity running**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show software authenticity running** **Example:** `Device# show software authenticity running`	Displays software authenticity-related information for the current ROMmon and the Cisco IOS image file used for booting.

Displaying Digital Signature Information for a Specific Image File

Perform this task to display the digital signature information related to software authentication for a specific image file.

SUMMARY STEPS

1. **enable**
2. **show software authenticity file {flash0:filename | flash1:filename | flash:filename | nvram:filename | flash0:filename | flash1:filename}**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show software authenticity file {flash0:filename | flash1:filename | flash:filename | nvram:filename | flash0:filename | flash1:filename}** **Example:**	Displays digital signature and software authenticity-related information for a specific image file.

	Command or Action	Purpose
	`Device# show software authenticity file flash0:c3900-universalk9-mz.SPA`	

Displaying Digitally Signed Cisco Software Key Information

Perform this task to display digitally signed Cisco software key information. The information details the software public keys that are in storage with the key types.

SUMMARY STEPS

1. **enable**
2. **show software authenticity keys**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show software authenticity keys** **Example:** `Device# show software authenticity keys`	Displays the software public keys that are in storage with the key types for digitally signed Cisco software.

Troubleshooting Digitally Signed Cisco Software Images

Perform this task to troubleshoot digitally signed Cisco software images.

SUMMARY STEPS

1. **enable**
2. **debug software- authenticity errors {envelope | errors | key | revocation | show | verbose}**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**debug software- authenticity errors {envelope | errors | key | revocation | show | verbose}** **Example:** `Device# debug software-authenticity errors`	Enables the display of debug messages for digitally signed Cisco software.

Configuration Examples for Digitally Signed Cisco Software

Identifying Digitally Signed Cisco Software Example

The following example displays the digitally signed Cisco software image filename and allows a user to identify it based on the digitally signed Cisco software identification criteria:

```
Device# show version
Cisco IOS Software, C3900 Software (C3900-UNIVERSALK9-M),
12.4(20090904:044027) [i12 577]
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Fri 04-Sep-09 09:22 by xxx
ROM: System Bootstrap, Version 12.4(20090303:092436)
C3900-2 uptime is 8 hours, 41 minutes
System returned to ROM by reload at 08:40:40 UTC Tue May 21 1901!
System image file is "xxx.SPA"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Cisco xxx (revision 1.0) with CISCxxx with 987136K/61440K bytes of memory.
Processor board ID xxx
3 Gigabit Ethernet interfaces
1 terminal line
1 Virtual Private Network (VPN) Module
1 cisco Integrated Service Engine(s)
DRAM configuration is 72 bits wide with parity enabled.
255K bytes of non-volatile configuration memory.
1020584K bytes of USB Flash usbflash0 (Read/Write)
1020584K bytes of USB Flash usbflash1 (Read/Write)
500472K bytes of ATA System CompactFlash 0 (Read/Write)
License Info:
License UDI:
------------------------------------------------
```

```
Device#   PID                   SN
-------------------------------------------------
xx        xxx                   xxxx
Technology Package License Information for Module:'xxx'
-----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent     ipbasek9
security      securityk9    Evaluation    securityk9
uc            None          None          None
data          None          None          None
Configuration register is 0x2102
```

Note the digitally signed image file is identified in the following line:

```
System image file is "xxx.SPA"
```

The image has a three-character extension in the filename (.SPA) characteristic of digitally signed Cisco software. Based on the guidelines in the “Digitally Signed Cisco Software Identification” section the first character in the file extension “S” indicates that the image is a digitally signed software image, the second character “P” indicates that the image is digitally signed using a production key, and the third character “A” indicates that the key version is version A.

Displaying Digitally Signed Cisco Software Signature Information Example

The following example shows how to display information related to software authentication for the current ROMmon and Cisco IOS image file used for booting:

```
Device# show software authenticity running
SYSTEM IMAGE
-------------------
Image type                    : Development
    Signer Information
        Common Name           : xxx
        Organization Unit     : xxx
        Organization Name     : xxx
    Certificate Serial Number : xxx
    Hash Algorithm            : xxx
    Signature Algorithm       : 2048-bit RSA
    Key Version               : xxx

    Verifier Information
        Verifier Name         : ROMMON 2
        Verifier Version      :  System Bootstrap, Version 12.4(20090409:084310)
ROMMON 2
---------------
Image type                    : xxx
    Signer Information
        Common Name           : xxx
        Organization Unit     : xxx
        Organization Name     : xxx
    Certificate Serial Number : xxx
    Hash Algorithm            : xxx
    Signature Algorithm       : 2048-bit RSA
    Key Version               : xx

    Verifier Information
        Verifier Name         : ROMMON 2
        Verifier Version      :  System Bootstrap, Version 12.4(20090409:084310) [
```

The table below describes the significant fields shown in the display.

Table 2: show software authenticity running Field Descriptions

Field	Description
SYSTEM IMAGE	Section of the output displaying the system image information.
Image type	Displays the type of image.
Common Name	Displays the name of the software manufacturer.
Organization Unit	Displays the hardware the software image is deployed on.
Organization Name	Displays the owner of the software image.
Certificate Serial Number	Displays the certificate serial number for the digital signature.
Hash Algorithm	Displays the type of hash algorithm used in digital signature verification.
Signature Algorithm	Displays the type of signature algorithm used in digital signature verification.
Key Version	Displays the key version used for verification.
Verifier Name	Name of the program responsible for performing the digital signature verification.
Verifier Version	Version of the program responsible for performing the digital signature verification.
ROMMON 2	Section of the output displaying the current ROMmon information.

Displaying the Digital Signature Information for a Specific Image File Example

The following example shows how to display the digital signature information related to software authentication for a specific image file:

Device# **show software authenticity file flash0:**c3900-universalk9-mz.SSA

```
File Name                    : flash0:c3900-universalk9-mz.SSA
Image type                   : Development
    Signer Information
        Common Name          : xxx
        Organization Unit    : xxx
        Organization Name    : xxx
    Certificate Serial Number : xxx
    Hash Algorithm           : SHA512
    Signature Algorithm      : 2048-bit RSA
    Key Version              : A

The table below describes the significant fields shown in the display.
```

Table 3: show software authenticity file Field Descriptions

Field	Description
File Name	Name of the filename in the memory. For example, flash0:c3900-universalk9-mz.SSA refers to filename c3900-universalk9-mz.SSA in flash memory (flash0:).
Image type	Displays the type of image.
Signer Information	Signature information.
Common Name	Displays the name of the software manufacturer.
Organization Unit	Displays the hardware the software image is deployed on.
Organization Name	Displays the owner of the software image.
Certificate Serial Number	Displays the certificate serial number for the digital signature.
Hash Algorithm	Displays the type of hash algorithm used in digital signature verification.
Signature Algorithm	Displays the type of signature algorithm used in digital signature verification.
Key Version	Displays the key version used for verification.

Displaying Digitally Signed Cisco Software Key Information Example

The following example displays digitally signed Cisco software key information. The information details the software public keys that are in storage, including their key types.

```
Device# show software authenticity keys
Public Key #1 Information
-------------------------
Key Type             : Release  (Primary)
Public Key Algorithm : RSA
Modulus :
        CC:CA:40:55:8C:71:E2:4A:3A:B6:9D:5C:94:1D:02:BA:
       ...
        26:04:6B:33:EB:70:2B:18:24:C7:D9:31:3E:77:24:85
Exponent : xxx
Key Version          : A
Public Key #2 Information
-------------------------
Key Type             : Development  (Primary)
Public Key Algorithm : RSA
Modulus :
        CC:CA:40:55:8C:71:E2:4A:3A:B6:9D:5C:94:1D:02:BA:
       ....
        26:04:6B:33:EB:70:2B:18:24:C7:D9:31:3E:77:24:85
Exponent : xxx
Key Version          : A

The table below describes the significant fields shown in the display.
```

Table 4: show software authenticity keys Field Descriptions

Field	Description
Public Key #	Public key number.
Key Type	Displays the key type used for image verification.
Public Key Algorithm	Displays the name of the algorithm used for public key cryptography.
Modulus	Modulus of the public key algorithm.
Exponent	Exponent of the public key algorithm
Key Version	Displays the key version used for verification.

Enabling Debugging of Digitally Signed Cisco Software Image Key Information Example

The following example shows how to enable debugging of software authentication events relating to key information for digitally signed Cisco software:

```
Device# debug software authenticity key
```

Additional References

The following sections provide references related to the Digitally Signed Cisco Software feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
System Management Command Reference	http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/software/release/3se/system_management/c

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Digitally Signed Cisco Software

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Digitally Signed Cisco Software

Feature Name	Releases	Feature Information
Digitally Signed Cisco Software		The Digitally Signed Cisco Software feature describes how to identify digitally signed Cisco software, gather software authentication information related to digitally signed images, and perform key revocation. Digitally Signed Cisco software is software that is digitally signed using secure asymmetrical (public-key) cryptography. The following commands were introduced or modified: **debug software authenticity, show software authenticity file, show software authenticity keys, show software authenticity running.**

Feature Name	Releases	Feature Information
Key Revocation Feature Support		Key revocation feature support was added. Key revocation removes a key from a platform's key storage. A platform can host a production or special image, and a production key (from a production image) or special key (from a special image) may be revoked during key revocation. The following section provides information about this feature: • Digitally Signed Cisco Software Key Revocation and Replacement The following commands were introduced or modified: **debug software authenticity, show software authenticity upgrade-status, software authenticity key add, software authenticity key revoke, upgrade rom-monitor file.**