Resize Default SSH RSA Keys on Cisco IOS XE SD-WAN Edges

Contents

Introduction

This document describes how to increase the default SSH RSA Keys used for secure protocols to a stronger length on Cisco IOS® XE SD-WAN Edges.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Catalyst Software-Defined Wide Area Network (SD-WAN)
- SSH Keys and Certificate basic operation
- RSA Algorithm

Components Used

- Cisco IOS® XE Catalyst SD-WAN Edges 17.9.4a

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Secure Shell (SSH) is a network protocol that allows users to establish remote connections to devices even over an unprotected network. The protocol secures the sessions using standard cryptographic mechanisms based on a client-server architecture.

RSA is Rivest, Shamir, Adleman: Encryption Algorithm (public-key cryptographic system) that uses two keys: Public and Private Key, also known as key pair. The Public RSA key is the encryption key and Private

RSA key is the decryption key.

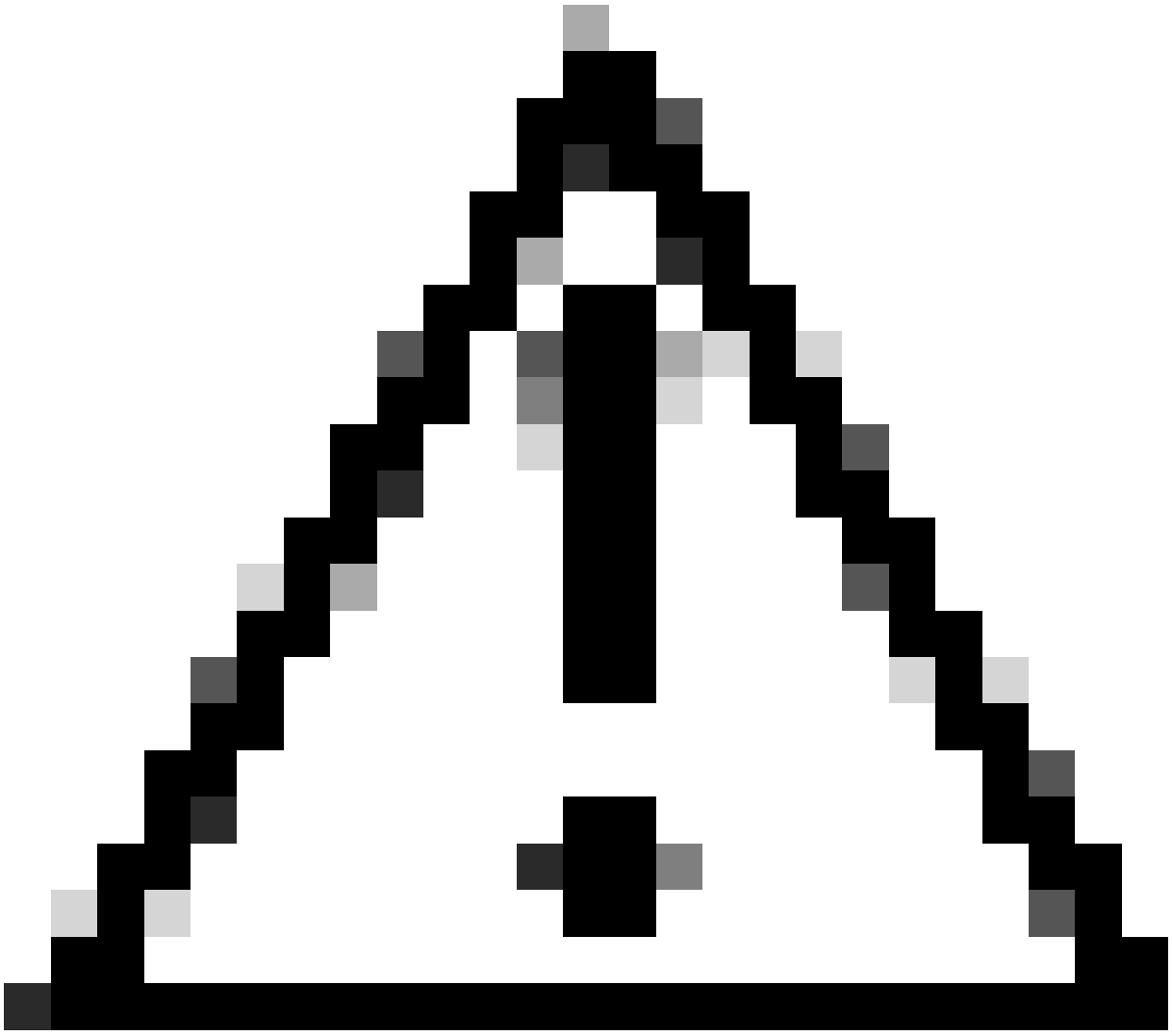
RSA Keys have a defined length, in bits, of the modulus. When an RSA key is said to have length 2048 bits, it really means that the modulus value lies between 22047 and 22048. Since the public and private keys of a given pair share the same modulus, they also have, by definition, the same length.

A trustpoint certificate is a self-signed certificate, hence the name trustpoint, since it does not rely on the trust of anyone else or other party.

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), Secure Shell (SSH), and Secure Socket Layer (SSL).

SSH RSA Keys are important on Cisco Catalyst SD-WAN because they are used by the SSH protocol to establish the communication between SD-WAN Manager and SD-WAN Edge devices, since SD-WAN Manager uses Netconf protocol, that works over SSH to manage, configure and monitor devices.

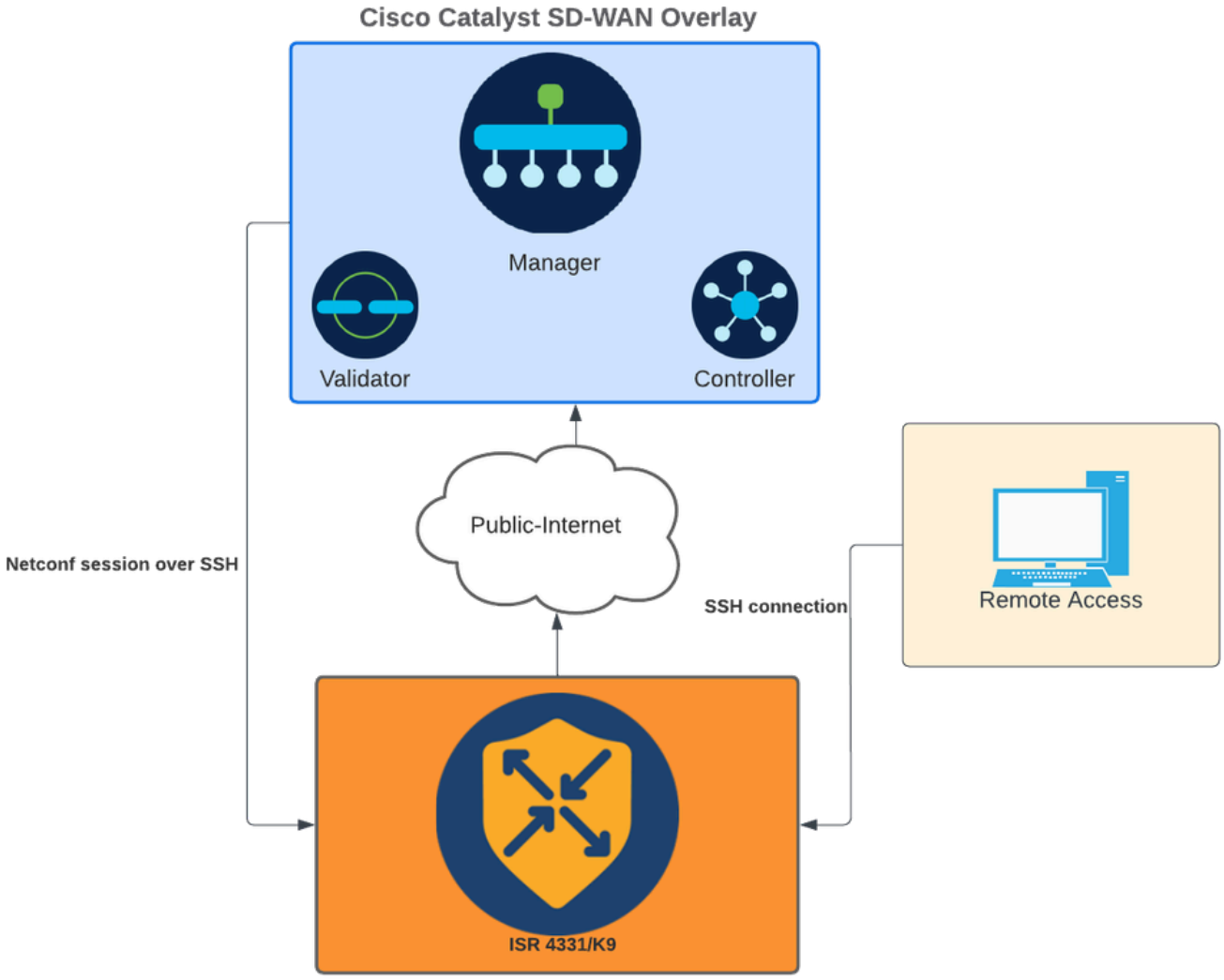
Due this fact, it is necessary that keys are synchronized and updated all the time. If by compliance and audit, it is needed to modify the key length for security, it is necessary to complete the process described on this document to resizing the keys and synchronize them correctly on the certificate to avoid disconnection between the SD-WAN Manager and SD-WAN Edge devices.



Caution: Please complete all the steps in the process to avoid loss of access to the device. If the device is in production, it is recommended to perform it in a maintenance window and have console access to the device.

Configure

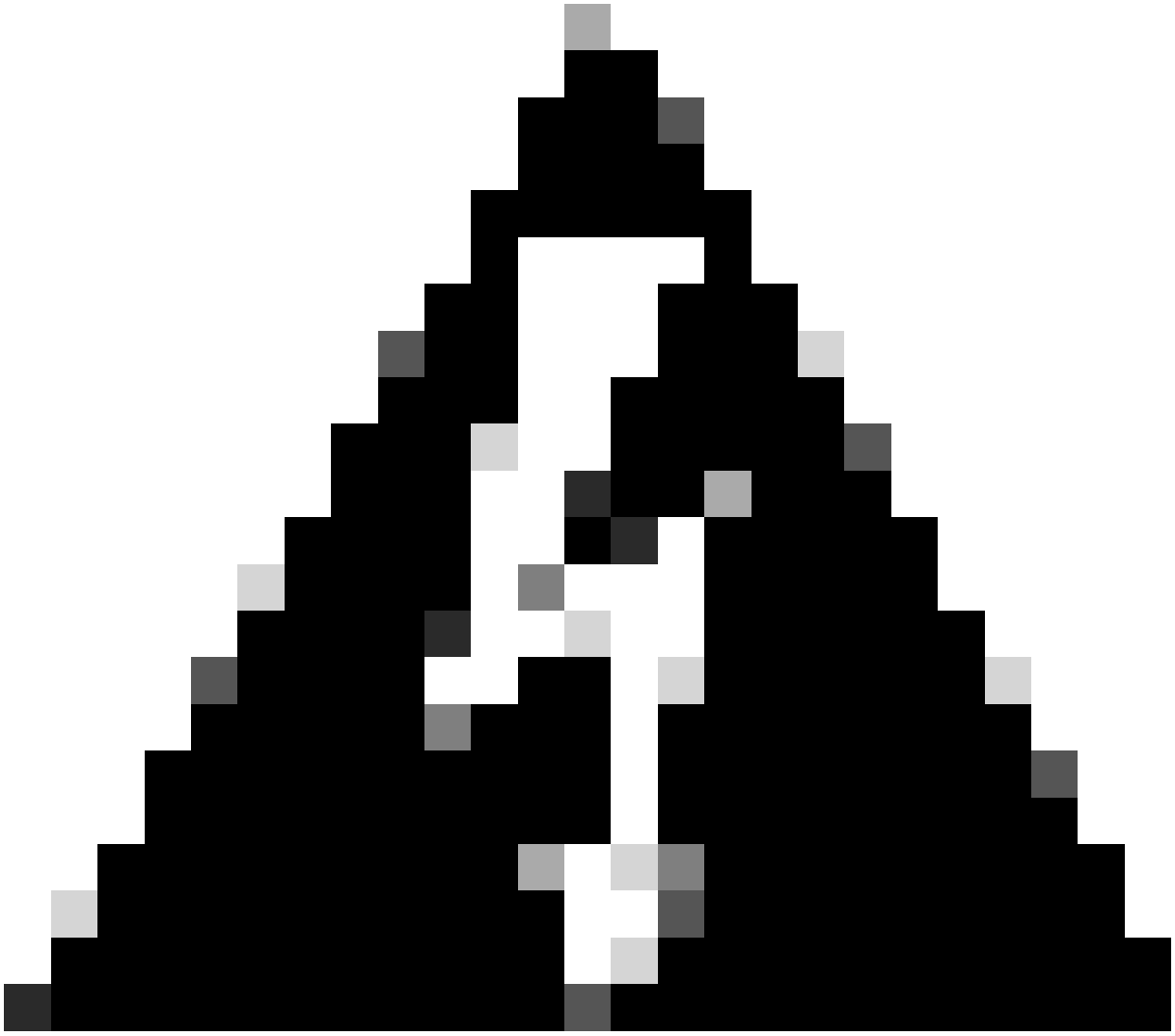
Network Diagram



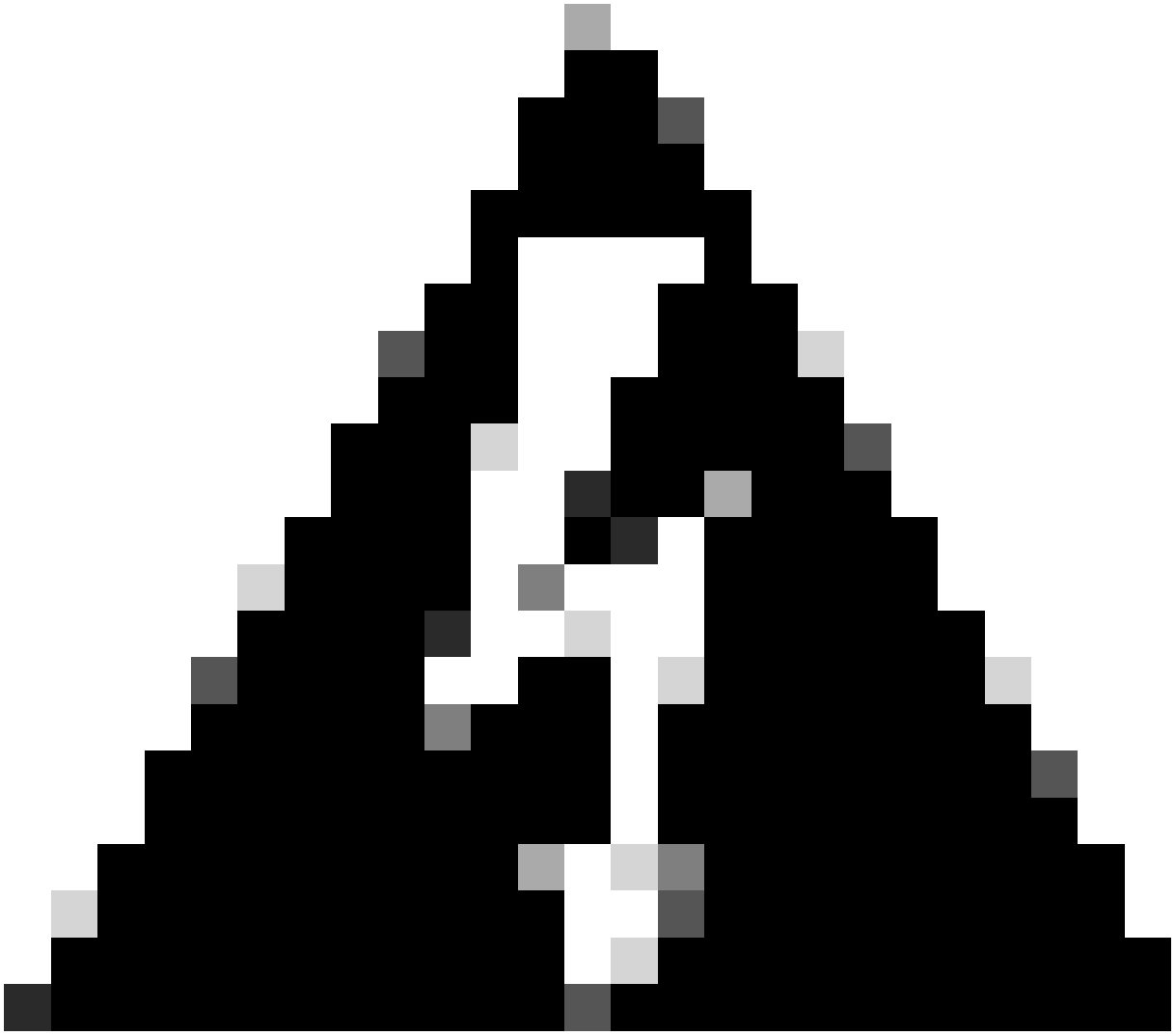
Network Diagram

Configurations

The RSA Keys in the WAN edge devices can only be modified using the Command-Line Interface (CLI); CLI Add-On Feature templates cannot be used to update the keys.



Warning: It is recommended to do the process with the use of the console as the SD-WAN Manager SSH Tool is unavailable until the process is finished.



Warning: This process requires a device restart. If the device is in production, it is recommended to perform it in a maintenance window and have console access to the device. If no console access, configure temporarily another remote access protocol as telnet.

This configuration example shows how to remove RSA 2048 and use RSA 4096 key.

1 - Get the current **SSH key name**.

<#root>

Device#

show ip ssh

```
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecds
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,
```

```
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
```

TP-self-signed-1072201169 <<<< RSA Key Name

Modulus Size : 2048 bits

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDZ5urq7f/X+AZJjUnMOdF9pLX+VOjPR8arK6bLSU7d
iGeSDDwW2MPNck/U5HBry9P/L4nKyZ1oevAhfy7cJVVmoHD41NQW9wb/hLtimuujnRRYkKuIWLmoI7AH
y6YQoetew8XVglVIjva+JzQ5ZX1JGm8AzN6a95RbRNhGRzgz9cTFmD7m6ArIKZPMyYQabXfrY+m/HuQ2
aytbHtJMgmOQk2fLPakO3PnQNYXpiDP3CmOEh3LJg82FZQ1eohmhm+mAInwU4mlLHUouigyBuq1KEBVe
z3vxjB9X8rGF3qzUcx2lpHmhXaNPXWen2QQbyfAIDo8WXVoff24uLYlwCVkv
```

2 - Get the current **trustpoint self-signed certificate**.

<#root>

```
Device#
```

show crypto pki trustpoint

Trustpoint TP-self-signed-1072201169: <<<< Self-signed Trustpoint name

```
Subject Name:
cn=IOS-Self-Signed-Certificate-1072201169
Serial Number (hex): 01
Persistent self-signed certificate trust point
Using key label
```

TP-self-signed-1072201169

Both value-names must match.

3 - Delete the **current key**.

<#root>

```
Device#
```

crypto key zeroize rsa <old_key_name>

4 - Validate that **old key** was deleted successfully.

<#root>

```
Device#
```

show ip ssh

5 - Generate the **new key**.

<#root>

```
Device#
```

crypto key generate rsa modulus 4096 label <old_key_name/trustpoint_name>

```
The name for the keys will be: TP-self-signed-1072201169
% The key modulus size is 4096 bits
% Generating crypto RSA keys in background ...
*Jun 25 21:35:18.919: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been gen
*Jun 25 21:35:18.924: %SSH-5-ENABLED: SSH 2.0 has been enabled
*Jun 25 21:35:23.205: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been gen
*Jun 25 21:35:29.674: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config file
```

This process can take 2 to 5 minutes to be completed.

6 - Validate the **new key** generated.

<#root>

```
Device#
```

show ip ssh

```
SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecds
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits
```

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169

Modulus Size : 4096 bits <<<< Key Size

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDEOt/SX3oQKN6zOWvOaFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm8O8mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796cOB1YLtK3eFOOH1AFmFy5ec8Own7ikO
JjKtwEozImFMjHZfUEUjFuhPJELBO6yYEipPWMRaZYFfTRbNJm8/7SOJGlFkgFVW5nITTIgISoMV8EJv
bLl8cVgATDb1OckeDb7uU6PDXm3zonmZCOyqHtF1OAOJxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5etOO49UHOyObMgPuWrJpwtk1iYA3+t6N/QdlC5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKyvkcqXi6nDfAKb8o+Z8/43xbvWlDIKAuj1rbdyqPAJB4llTZJkOHk8zRP5gZ8u4jTjNKQHb
vNa3ieg4RLEDOx4lqCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOul4YDJYr
D+jnyoZr4PnfWAgk4M9U89deWSlIRPMIXYd35YmLvD6OeQ5EQALNiNPUEkpdPKs4orYysEVOpRoY+HQ
```

Now, new key is generated. However, at the moment that old key was deleted, self-signed certificate that is in use by Netconf sessions is also deleted from the trustpoint.

```
<#root>

Device#

sh crypto pki trustpoint status


Trustpoint TP-self-signed-1072201169:
Issuing CA certificate configured::
Issuing CA certificate configured:
Subject Name:
cn=Cisco Licensing Root CA,o=Cisco
Fingerprint MD5: 1468DC18 250BDFCF 769C29DF E1F7E5A8
Fingerprint SHA1: 5CA95FB6 E2980EC1 5AFB681B BB7E62B5 AD3FA8B8
State:

Keys generated ..... No <<<< Depending on the version, it can erase the key or even that, delete


Issuing CA authenticated ..... Yes
Certificate request(s) ..... None
```

Once the new 4096 key is generated, the keys are not automatically updated on the self-signed certificate, and it is necessary complete extra steps to update it.

Note: If only the key is generated, but is not updated in the certificate, the SD-WAN Manager loses the Netconf sessions, and that could break all management activities to the device (templates, configuration, and so on).

There are two ways to generate the certificate/assign the key:

1 - Reload the **device**.

```
<#root>

Device#

reload
```

2 - Restart **HTTP secure-server**. This option is only available if the device is on CLI mode.

```
<#root>

Device (config)#

no ip http secure-server


Device (config)#
```

```
commit

Device (config)#
ip http secure-server

Device (config)#
commit
```

Verify

After the reload, validate that **new key** is generated and the **certificate** is under trustpoint with the same name.

```
<#root>

Device#
 show ip ssh

SSH Enabled - version 2.0
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecds
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169


Modulus Size : 4096 bits


ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDEOt/SX3oQKN6z0WvOaFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm8O8mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796cOB1YLtK3eFOOH1AFmFy5ec8Own7ikO
JjKtwEozImFMjHZfUEUjFuhPJELBO6yYEipPWMRaZYFfTRbNJm8/7SOJGlFkgFVW5nITTIgISoMV8EJv
bLl8cVgATDb1OckeDb7uU6PDXm3zonmZCOyqHtF1OAOJxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5etOO49UHOyObMgPuWrJpwtk1iYA3+t6N/QdlC5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKyvkcqXi6nDfAKb8o+Z8/43xbvWlDIKAuj1rbdyqPAJB4llTZJkOHk8zRP5gZ8u4jTjNKQHb
vNa3ieg4RLEDOx4lqCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOul4YDJYr
D+jnyoZr4PnfWAgk4M9U89deWSlIRPMIXYd35YmLvD6OeQ5EQALNiNPUEkpdPKs4orYysEVOpRoY+HQ


<#root>

Device#
show crypto pki trustpoint
```

Trustpoint TP-self-signed-1072201169: <<<< Trustpoint name

Subject Name:
cn=IOS-Self-Signed-Certificate-1072201169
Serial Number (hex): 01
Persistent self-signed certificate trust point

Using key label TP-self-signed-107220116

<#root>

Device#

show crypto pki certificates

Router Self-Signed Certificate
Status: Available
Certificate Serial Number (hex): 01
Certificate Usage: General Purpose
Issuer:
cn=IOS-Self-Signed-Certificate-1072201169
Subject:
Name: IOS-Self-Signed-Certificate-1072201169
cn=IOS-Self-Signed-Certificate-1072201169
Validity Date:
start date: 21:07:33 UTC Dec 27 2023
end date: 21:07:33 UTC Dec 26 2033

Associated Trustpoints: TP-self-signed-1072201169

Storage: nvram:IOS-Self-Sig#4.cer

Confirm that SD-WAN Manager can apply configuration changes to the device router.