Cisco IOS XE SD-WANエッジのデフォルトSSH RSAキーのサイズ変更

内容

はじめに

前提条件

- 要件
- 使用するコンポーネント

背景説明

設定

- ネットワーク図
- コンフィギュレーション

確認

はじめに

このドキュメントでは、Cisco IOS® XE SD-WANエッジで、セキュアプロトコルに使用されるデフォルトのSSH RSAキーをより強力な長さに増やす方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco Catalystソフトウェア定義型ワイドエリアネットワーク(SD-WAN)
- SSHキーと証明書の基本操作
- RSAアルゴリズム

使用するコンポーネント

- Cisco IOS® XE Catalyst SD-WANエッジ17.9.4a

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

セキュアシェル(SSH)は、ユーザが保護されていないネットワーク上でもデバイスへのリモート

接続を確立できるネットワークプロトコルです。このプロトコルは、クライアント/サーバアーキテクチャに基づく標準的な暗号化メカニズムを使用してセッションを保護します。

RSAはRivest, Shamir, Adlemanの略で、公開鍵と秘密鍵の2つの鍵を使用する暗号化アルゴリズム（公開鍵暗号化システム）です。公開鍵と秘密鍵は鍵ペアとも呼ばれます。公開RSAキーは暗号化キーで、秘密RSAキーは復号化キーです。

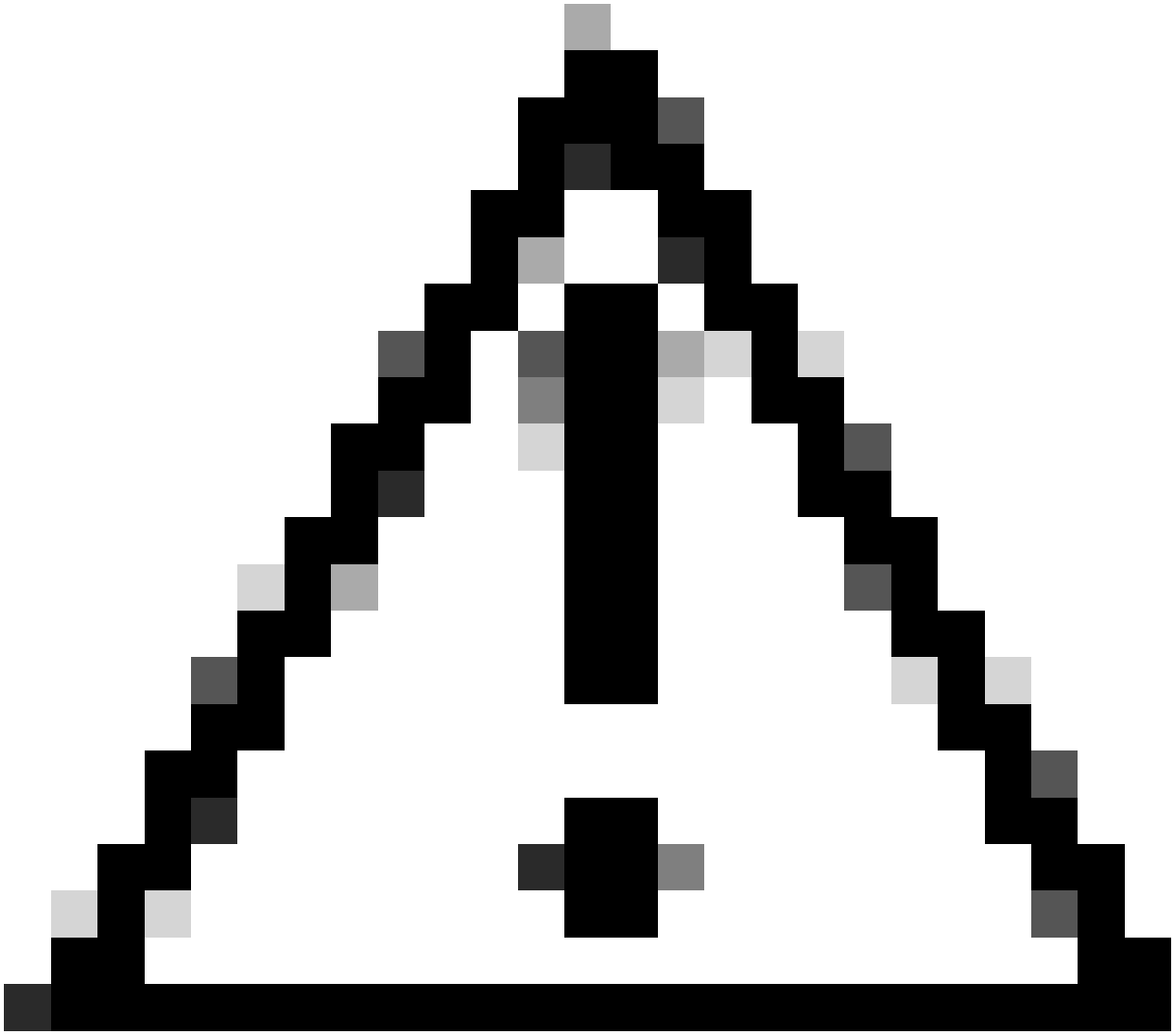
RSA鍵には、ビット単位で定義されたモジュラス長があります。RSAキーの長さは2048ビットと言われますが、実際にはモジュラス値は22047 ~ 22048であることを意味します。特定のペアの公開キーと秘密キーは同じモジュラスを共有するため、定義上は同じ長さになります。

トラストポイント証明書は自己署名証明書であり、他のユーザや他のユーザの信頼に依存しないため、トラストポイントという名前が付けられます。

Cisco IOS Public Key Infrastructure(PKI)は、IP Security(IPSec)、Secure Shell(SSH)、Secure Socket Layer(SSL)などのセキュリティプロトコルをサポートする証明書管理を提供します。

SSH RSAキーは、SSHプロトコルによってSD-WAN ManagerとSD-WAN Edgeデバイス間の通信を確立するために使用されるため、Cisco Catalyst SD-WANでは重要です。これは、SD-WAN Managerでは、SSH経由で動作するNetconfプロトコルを使用してデバイスを管理、設定、および監視するためです。

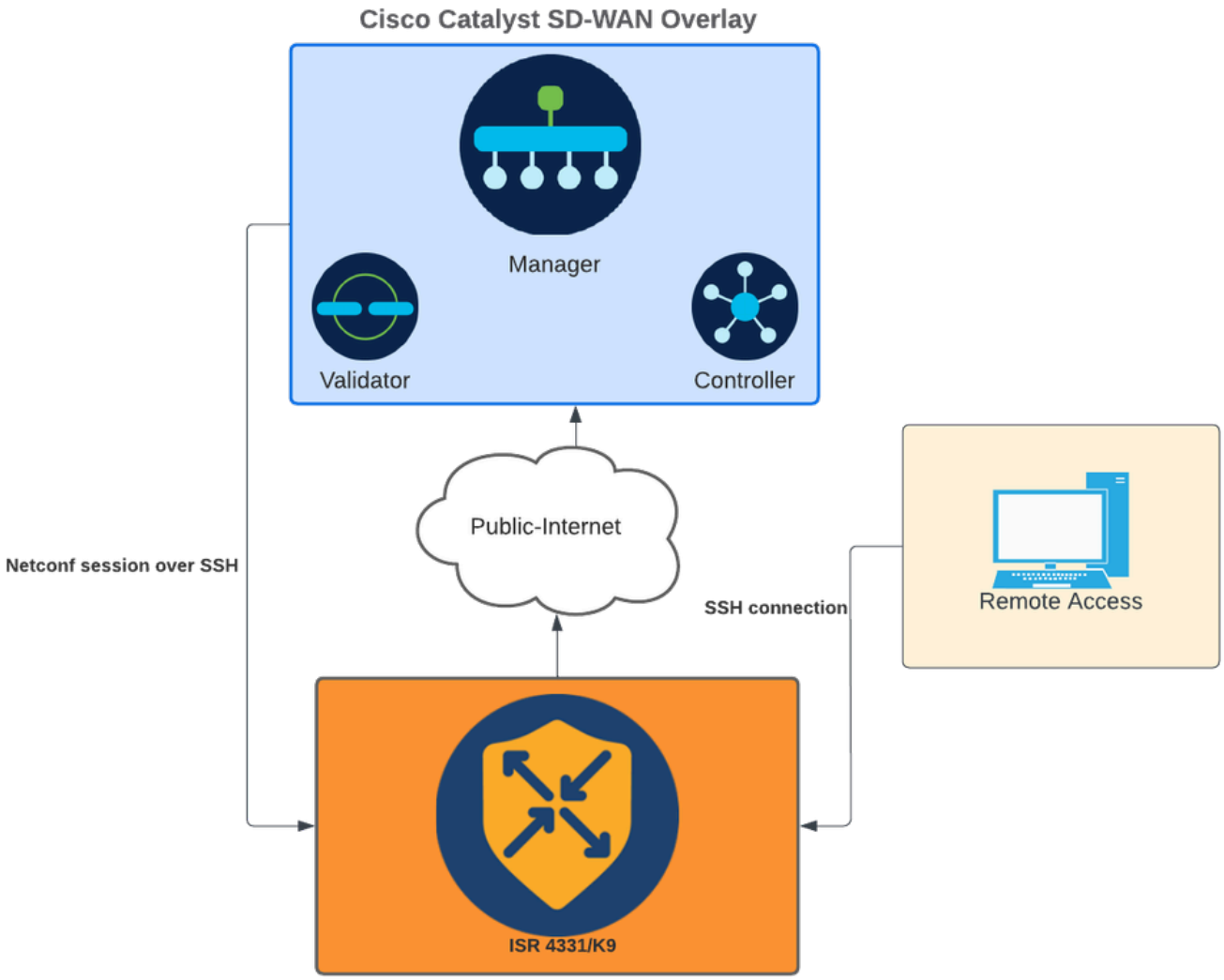
このため、キーは常に同期および更新される必要があります。コンプライアンスと監査により、セキュリティのためにキーの長さを変更する必要がある場合は、SD-WAN Manager(SWM)とSD-WAN Edgeデバイス間の切断を避けるために、このドキュメントで説明されているプロセスを実行して、キーのサイズを変更し、証明書で正しく同期させる必要があります。



注意：デバイスへのアクセスが失われないようにするため、このプロセスのすべての手順を実行してください。デバイスが実稼働環境にある場合は、メンテナンス時間帯にデバイスを実行し、デバイスにコンソールからアクセスできるようにすることをお勧めします。

設定

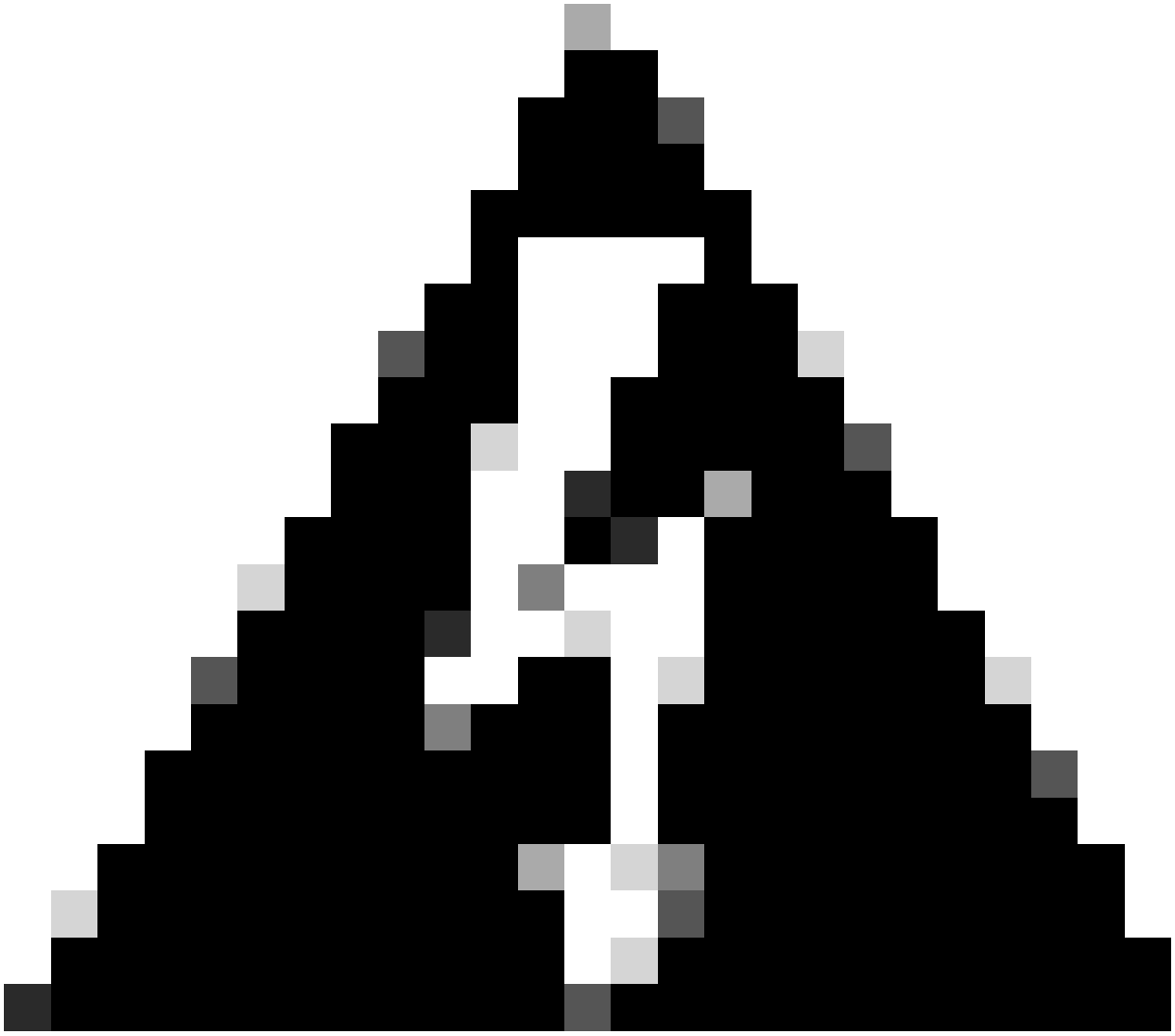
ネットワーク図



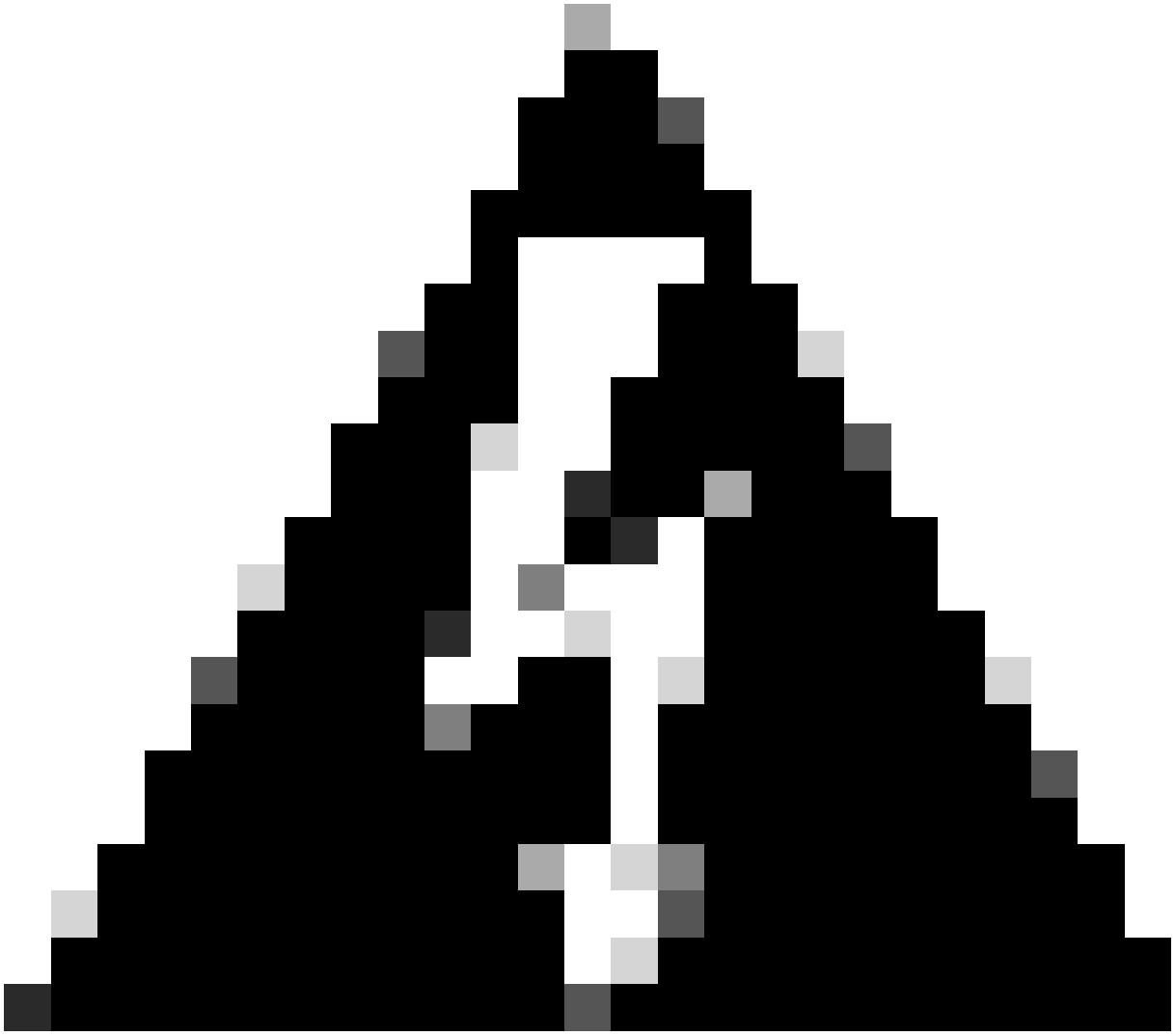
ネットワーク図

コンフィギュレーション

WANエッジデバイスのRSAキーは、コマンドラインインターフェイス(CLI)を使用してのみ変更できます。CLIアドオン機能テンプレートを使用してキーを更新することはできません。



警告：プロセスが終了するまでSD-WAN Manager SSH Toolは使用できないため、コンソールを使用してこのプロセスを実行することをお勧めします。



警告：このプロセスではデバイスを再起動する必要があります。デバイスが実稼働環境にある場合は、メンテナンス時間帯にデバイスを実行し、デバイスにコンソールからアクセスできるようにすることをお勧めします。コンソールアクセスがない場合は、一時的に別のリモートアクセスプロトコルをtelnetとして設定します。

次の設定例では、RSA 2048を削除し、RSA 4096キーを使用する方法を示しています。

1：現在のSSHキー名を取得します。

<#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：現在のトラストポイント自己署名証明書を取得します。

<#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

両方の値の名前が一致している必要があります。

3 -現在のキーを削除します。

<#root>

```
Device#
```

crypto key zeroize rsa

4:古いキーが正常に削除されたことを検証します。

<#root>

Device#

show ip ssh

5:新しいキーを生成します。

<#root>

Device#

crypto key generate rsa modulus 4096 label

```
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
```

このプロセスの完了には2 ~ 5分かかります。

6:生成された新しいキーを検証します。

<#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/SX3oQKN6z0WvOaFAkMcaZNzQ6JgP+7xjuX143
YS7YGm0PwIPgs8N2LWvmdLXQ/PqsQ0GGsdxo2+2Y/idAFm8O8mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796c0B1YLtK3eFO0H1AFmFy5ec8Own7ik0
JjKtwEozImFMjHZfUEUjFuhPJELBO6yYEipPWMRaZYFfTRbNJm8/7SOJGlFkgFVW5nITTIgISoMV8EJv
bLl8cVgATDb1OckeDb7uU6PDXm3zonmZC0yqHtF1OAOJxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5et0O49UH0y0bMgPuWrJpwtk1iYA3+t6N/QdlC5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKyvkcqXi6nDfAKb8o+Z8/43xbvWlDIKAuj1rbdyqPAJB4llTZJkOHk8zRP5gZ8u4jTjNKQHb
vNa3ieg4RLEDOx4lqCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOul4YDJYr
D+jnyoZr4PnfWAgk4M9U89deWSlIRPMIXYd35YmLvD6OeQ5EQALNiNPUEkpdPKs4orYysEV0pRoY+HQ
```

ここで、新しいキーが生成されます。ただし、古いキーが削除された時点で、Netconfセッションで使用されている自己署名証明書もトラストポイントから削除されます。

<#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
```

新しい4096キーが生成されても、キーは自己署名証明書では自動的に更新されません。そのため、キーを更新するには追加の手順を実行する必要があります。

注：キーが生成されただけで、証明書で更新されていない場合、SD-WAN ManagerではNetconfセッションが失われ、その結果、デバイスに対するすべての管理アクティビティ（テンプレート、設定など）が中断される可能性があります。

証明書の生成とキーの割り当てには、次の2つの方法があります。

1:デバイスをリロードする。

<#root>

Device#

reload

2:HTTP secure-serverを再起動します。このオプションは、デバイスがCLIモードの場合にのみ使用できます。

<#root>

Device (config)#

no ip http secure-server

Device (config)#

commit

Device (config)#

ip http secure-server

Device (config)#

commit

確認

リロード後、新しいキーが生成され、証明書が同じ名前のトラストポイントにあることを検証します。

<#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/SX3oQKN6zOWvOaFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm8O8mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796c0B1YLtK3eFOOH1AFmFy5ec8Own7ikO
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
```

SD-WAN Managerがデバイスルータに設定変更を適用できることを確認します。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。