PKI 内での RSA キーの展開

この章では、Public Key Infrastructure（PKI; 公開キー インフラストラクチャ）内で Rivest、Shamir、Adelman（RSA）キーを設定および展開する方法について説明します。ルータの証明書を取得する前に、RSA キー ペア（公開キーと秘密キー）が要求されます。つまり、エンド ホストは RSA キーのペアを生成し、認証局（CA）と公開キーを交換して証明書を取得し、PKI に登録する必要があります。

機能情報の入手

ご使用のソフトウェア リリースでは、この章で説明されるすべての機能がサポートされているとは限りません。最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。この章に記載されている機能の詳細、および各機能がサポートされているリリースのリストについては、「PKI 内の RSA キーに関する機能情報」（P.23）を参照してください。

Cisco Feature Navigator を使用すると、プラットフォーム、および Cisco ソフトウェア イメージの各サポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

この章の構成

PKI での RSA キーの設定に関する前提条件

- PKI の RSA キーを設定および展開する前に、「*Cisco IOS PKI Overview: Understanding and Planning a PKI*」の内容を理解している必要があります。
- Cisco IOS Release 12.3(7)T の時点で、コマンドの先頭に付けられていた「crypto ca」は、すべて「crypto pki」に変更されました。ルータは引き続き crypto ca コマンドを受け入れますが、すべての出力は crypto pki として読み替えられます。

RSA キーの設定に関する情報

RSA キーの概要

RSA キー ペアは、公開キーと秘密キーで構成されます。PKI を設定する場合、証明書登録要求に公開キーを含める必要があります。証明書が付与された後、ピアが公開キーを使用して、ルータに送信されるデータを暗号化できるように、公開キーが証明書に組み込まれます。秘密キーはルータに保持され、ピアによって送信されたデータの復号化と、ピアとネゴシエーションするときの、トランザクションのデジタル署名に使用されます。

RSA キー ペアには、キーのモジュラス値が含まれています。モジュラス値に応じて、RSA キーのサイズが決まります。モジュラス値が大きいほど、RSA キーの安全性が高まります。ただし、モジュラス値が大きくなると、キーの生成にかかる時間が長くなり、キーのサイズが大きくなると暗号化処理および復号化処理にかかる時間が長くなります。

（注） Cisco IOS Release 12.4(11)T の時点では、最大 4096 ビットまでのピアの*公開* RSA キーのモジュラス値が自動的にサポートされます。

秘密 RSA キーの最大モジュラス値は 4096 ビットです。したがって、ルータが生成またはインポートできる RSA 秘密キーの最大サイズは、4096 ビットです。ただし、RFC 2409 では、RSA 暗号化の秘密キーのサイズを 2048 ビット以下に制限しています。

CA の推奨モジュラス値は 2048 ビット、クライアントの推奨モジュラス値は 1024 ビットです。

用途 RSA キーと汎用目的 RSA キー

RSA キー ペアには用途キーと汎用目的キーの 2 つのタイプがあり、これらは相互に排他的です。RSA キー ペアを生成するとき（**crypto key generate rsa** コマンドを使用）、用途キーまたは汎用目的キーを選択するためのプロンプトが表示されます。

用途 RSA キー

用途キーは 2 組の RSA キー ペアで構成されます。このうち 1 組の RSA キー ペアは暗号化用に、もう 1 組の RSA キー ペアは署名用にそれぞれ生成され、使用されます。用途キーを使用すると、各キーは不必要に暴露されなくなります（用途キーを使用しない場合、1 つのキーが両方の認証方法に使用されるため、そのキーが暴露される危険性が高くなります）。

汎用目的 RSA キー

汎用目的キーは、1 つの RSA キー ペアだけで構成され、このキー ペアは暗号化と署名の両方に使用されます。汎用目的のキー ペアは、用途キー ペアよりも頻繁に使用されます。

ルータに複数の RSA キーを保管する理由

複数の RSA キー ペアを設定することで、Cisco IOS ソフトウェアは、対応する CA ごとに異なるキー ペアを維持できます。このようにして、このソフトウェアは、同じ CA で複数のキー ペアおよび証明書を維持できます。したがって、Cisco IOS ソフトウェアは、キーの長さ、キーのライフタイム、汎用目的キーまたは用途キーなど、他の CA で指定される要件を損なうことなく、各 CA のポリシー要件に合致します。

名前付きのキー ペア（**label** *key-label* オプションを使用して指定する）を使用して、複数の RSA キー ペアを用意すると、Cisco IOS ソフトウェアがアイデンティティの証明書ごとに異なるキー ペアを維持できるようになります。

エクスポート可能な RSA キーのメリット

注意 エクスポート可能な RSA キーを使用すると、キーが暴露される危険性があるため、エクスポート可能な RSA キーは、使用前に慎重に評価する必要があります。

既存の RSA キーはすべてエクスポート*不能*です。新しいキーは、デフォルトでエクスポート不能として生成されます。既存のエクスポート不能のキーは、エクスポート可能なキーに変換できません。

Cisco IOS Release 12.2(15)T では、ユーザは、ルータの秘密 RSA キー ペアをスタンバイ ルータと共有できます。したがって、ネットワーキング デバイス間でセキュリティ クレデンシャルを転送できます。キー ペアを 2 台のルータ間で共有すると、一方のルータが、もう一方のルータの機能を迅速かつトランスペアレントに引き継ぐことができます。メイン ルータが故障した場合、スタンバイ ルータがネットワークに投入され、キーの再生、CA への再登録、または手動でのキーの再配布を行うことなく、メイン ルータを置き換えます。

また、セキュア シェル（SSH）を使用するすべての管理ステーションを 1 つの公開 RSA キーで設定できるように、RSA キー ペアをエクスポートおよびインポートすると、ユーザは同じ RSA キー ペアを複数のルータに配置することもできます。

PEM 形式ファイルでエクスポート可能な RSA キー

Privacy-Enhanced Mail（PEM; プライバシーエンハンスト メール）形式ファイルを使用した RSA キーのインポートまたはエクスポートは、Cisco IOS ソフトウェア リリース 12.3(4)T 以降を実行するお客様および、Secure Socket Layer（SSL; セキュア ソケット レイヤ）またはセキュア シェル（SSH）アプリケーションを使用して、RSA キー ペアを手動で生成し、キーを PKI アプリケーションに再インポートするお客様に役立ちます。PEM 形式のファイルを使用すると、新しいキーを生成しなくても、既存の RSA キー ペアを Cisco IOS ルータで直接使用できます。

RSA キーのインポートおよびエクスポート時のパスフレーズ保護

エクスポートする PKCS12 ファイルまたは PEM ファイルを暗号化するには、パスフレーズを含める必要があります。また、PKCS12 または PEM ファイルをインポートするときは、同じパスフレーズを入力して復号化する必要があります。PKCS12 または PEM ファイルをエクスポート、削除、またはインポートする際にこれらのファイルを暗号化すると、ファイルの伝送あるいは外部デバイスへの保管中に、ファイルを不正なアクセスおよび使用から保護します。

パスフレーズには、8 文字以上の任意のフレーズを指定できます。パスフレーズにはスペースおよび句読点を含めることができますが、Cisco IOS パーサに特殊な意味を持つ疑問符（?）は除きます。

エクスポート可能な RSA キー ペアをエクスポート不能な RSA キー ペアに変換する方法

パスフレーズ保護により、外部の PKCS12 または PEM ファイルが不正なアクセスおよび使用から保護されます。RSA キー ペアがエクスポートされないようにするには、RSA キー ペアに「エクスポート不能」のラベルを付ける必要があります。エクスポート可能な RSA キー ペアをエクスポート不能なキー ペアに変換するには、キー ペアをエクスポートしてから、「exportable」キーワードを指定せずに再インポートする必要があります。

PKI 内で RSA キーを設定および展開する方法

RSA キー ペアの生成

RSA キー ペアを手動で生成するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **exit**
5. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*] **例：** `Router(config)# crypto key generate rsa general-keys modulus 360`	（任意）証明書サーバの RSA キー ペアを生成します。 • **storage** キーワードを使用すると、キーの保管場所を指定できます。 • *key-label* 引数を指定することによってラベル名を指定する場合、**crypto pki server** *cs-label* コマンドによって証明書サーバに使用するラベルと同じ名前を使用する必要があります。*key-label* 引数を指定していない場合、ルータの Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）であるデフォルト値が使用されます。 **no shutdown** コマンドを発行する前に、CA 証明書が生成されるまで待ってからエクスポート可能な RSA キー ペアを手動で生成する場合、**crypto ca export pkcs12** コマンドを使用して、証明書サーバ証明書および秘密キーを含む PKCS12 ファイルをエクスポートできます。 • デフォルトでは、CA キーのモジュラス サイズは 1024 ビットです。推奨される CA キーのモジュラスは 2048 ビットです。CA キーのモジュラス サイズの範囲は 350 ～ 4096 ビットです。 • **on** キーワードは、指定した装置上で RSA キー ペアが作成されることを指定します。この装置には Universal Serial Bus（USB; ユニバーサルシリアルバス）トークン、ローカル ディスク、および NVRAM などがあります。装置の名前の後にはコロン（:）を付けます。 **（注）** USB トークン上で作成されるキーは、2048 ビット以下である必要があります。
ステップ 4	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）ルータの RSA 公開キーを表示します。 このステップでは、RSA キー ペアが正常に生成されたことを確認できます。

次の作業

正常に RSA キー ペアを生成したら、この章のいずれかの追加作業に進み、RSA キー ペアに対して追加の RSA キー ペアを生成する、RSA キー ペアのエクスポートおよびインポートを実行する、または追加のセキュリティ パラメータ（秘密キーの暗号化またはロックなど）を設定します。

RSA キー ペアとトラストポイントの証明書の管理

複数の RSA キー ペアを生成および保管し、トラストポイントにキー ペアを関連付け、トラストポイントからルータの証明書を取得するようにルータを設定するには、次の作業を実行します。

トラストポイント（Certificate Authority（CA; 認証局）としても知られる）は、証明書要求を管理し、参加ネットワーク デバイスに証明書を発行します。これらのサービスは、参加デバイスを一元的に管理します。またこれらのサービスによって受信者は、明示的に信頼してアイデンティティを確認し、デジタル証明書を作成できます。PKI の動作を開始する前に、CA は独自の公開キー ペアを生成し、自己署名 CA 証明書を作成します。その後、CA は、証明書要求に署名し、PKI に対してピア登録を開始できます。

前提条件

RSA キー ペアを、「RSA キー ペアの生成」の手順どおりに生成している必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
5. **enrollment selfsigned**（任意）
6. **subject-alt-name** *name*（任意）
7. **exit**
8. **cypto pki enroll** *name*
9. **exit**
10. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki trustpoint** *name* **例：** `Router(config)# crypto pki trustpoint TESTCA`	トラストポイントを作成し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **例：** `Router(ca-trustpoint)# rsakeypair fancy-keys`	（任意）*key-label* 引数には、登録時に生成された RSA キーペアの名前を指定し（まだ存在しない場合、または **auto-enroll regenerate** コマンドが設定されている場合）、トラストポイント証明書と一緒に使用します。デフォルトでは、Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）キーを使用します。 • （任意）*key-size* 引数には、RSA キー ペアのサイズを指定します。 • （任意）*encryption-key-size* 引数には 2 番めのキーのサイズを指定します。2 番めのキーは、個別の暗号化、署名キー、および証明書を要求する場合に使用されます。
ステップ 5	**enrollment selfsigned** **例：** `Router(ca-trustpoint)# enrollment selfsigned`	（任意）トラストポイントの自己署名登録を指定します。
ステップ 6	**subject-alt-name** *name* **例：** `Router(ca-trustpoint)# subject-alt-name TESTCA`	（任意）*name* 引数には、トラストポイントの証明書に含まれる X.509 証明書の所有者別名（subjectAltName）フィールドのトラストポイントの名前を指定します。デフォルトでは、証明書に所有者別名フィールドは含まれていません。 **（注）** X.509 証明書のこのフィールドは、RFC 2511 に定義されています。 このオプションは、所有者別名（subjectAltName）フィールドにトラストポイントの名前を含むルータの自己署名トラストポイント証明書を作成する場合に使用します。所有者別名は、トラストポイント ポリシーの自己署名登録に **enrollment selfsigned** コマンドが指定された場合にのみ使用できます。
ステップ 7	**exit** **例：** `Router(ca-trustpoint)# exit`	CA トラストポイント コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的

ステップ 8	**cypto pki enroll** *name*
	
	例：
	`Router(config)# cypto pki enroll TESTCA`
	`% Include the router serial number in the subject name? [yes/no]: no`
	`% Include an IP address in the subject name? [no]:`
	`Generate Self Signed Router Certificate? [yes/no]: yes`
	
	`Router Self Signed Certificate successfully created`
		トラストポイントからのルータの証明書を要求します。
		name 引数にはトラストポイントの名前を指定します。このコマンドを入力したら、プロンプトに応答します。
		（注） **crypto pki trustpoint** コマンドで入力したものと同じトラストポイント名を使用します。

ステップ 9	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 10	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）ルータの RSA 公開キーを表示します。 このステップでは、RSA キー ペアが正常に生成されたことを確認できます。

例

次に、所有者別名（subjectAltName）フィールドにトラストポイントの名前を含むルータの自己署名トラストポイント証明書を作成する方法の例を示します。

```
Router> enable
Router# configure terminal
Router(config)# crypto pki trustpoint TESTCA
Router(ca-trustpoint)# enrollment selfsigned
Router(ca-trustpoint)# subject-alt-name TESTCA
Router(ca-trustpoint)# exit
Router(config)# cypto pki enroll TESTCA
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]:
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created
Router(config)# exit
```

次の証明書が作成されます。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: CN=TESTCA/unstructuredName=r1.cisco.com
        Validity
            Not Before: Mar 22 20:26:20 2010 GMT
            Not After : Jan  1 00:00:00 2020 GMT
        Subject: CN=TESTCA/unstructuredName=r1.cisco.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
```

```
                Modulus (512 bit):
                    00:8d:71:2e:3b:eb:a2:e2:f3:44:d9:bc:a9:85:88:
                    f4:a9:bd:c9:7f:f0:69:f5:e7:75:8f:00:f2:8e:3e:
                    2f:ca:5e:c5:08:43:95:8c:a2:6a:ae:ce:a0:ae:82:
                    61:61:ff:4e:8c:8f:89:d1:56:d8:35:34:b7:95:93:
                    1a:72:03:71:fb
                Exponent: 65537 (0x10001)
            X509v3 extensions:
                X509v3 Basic Constraints: critical
                CA:TRUE
                X509v3 Subject Alternative Name:
                DNS:TESTCA
                X509v3 Authority Key Identifier:
                keyid:F9:A4:95:87:5F:A4:CA:7D:65:FA:BE:38:20:55:18:F9:4C:6C:D5:F3

                X509v3 Subject Key Identifier:
                F9:A4:95:87:5F:A4:CA:7D:65:FA:BE:38:20:55:18:F9:4C:6C:D5:F3
        Signature Algorithm: md5WithRSAEncryption
            6d:92:e7:a8:a5:1a:5a:ef:13:58:02:1b:79:17:93:41:37:c9:
            2d:9f:1a:a3:f5:3a:73:05:cd:d1:02:84:43:7e:e0:84:07:46:
            55:f9:45:59:51:ba:25:48:6f:d8:e1:0d:35:44:07:5c:16:17:
            35:45:99:e2:80:6e:53:e5:35:76
-----BEGIN CERTIFICATE-----
MIIBszCCAV2gAwIBAgIBAjANBgkqhkiG9w0BAQQFADAuMQ8wDQYDVQQDEwZURVNU
Q0ExGzAZBgkqhkiG9w0BCQIWDHIxLmNpc2NvLmNvbTAeFw0xMDAzMjIyMDI2MjBa
Fw0yMDAxMDEwMDAwMDBaMC4xDzANBgNVBAMTBlRFU1RDQTEbMBkGCSqGSIb3DQEJ
AhYMcjEuY2lzY28uY29tMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAI1xLjvrouLz
RNm8qYWI9Km9yX/wafXndY8A8o4+L8pexQhDlYyiaq7OoK6CYWH/ToyPidFW2DU0
t5WTGnIDcfsCAwEAAaNmMGQwDwYDVR0TAQH/BAUwAwEB/zARBgNVHREECjAIggZU
RVNUQ0EwHwYDVR0jBBgwFoAU+aSVh1+kyn1l+r44IFUY+Uxs1fMwHQYDVR0OBBYE
FPmklYdfpMp9Zfq+OCBVGPlMbNXzMA0GCSqGSIb3DQEBBAUAA0EAbZLnqKUaWu8T
WAIbeReTQTfJLZ8ao/U6cwXN0QKEQ37ghAdGVflFWVG6JUhv2OENNUQHXBYXNUWZ
4oBuU+U1dg==
-----END CERTIFICATE-----
```

RSA キーのエクスポートおよびインポート

ここでは、RSA キーのエクスポートおよびインポートに使用できる次の作業について説明します。エクスポート可能な RSA キーを使用すると、メイン ルータが故障した場合に、使用ファイルが PKCS12 ファイルか PEM ファイルかにかかわらず、新しい RSA キーを生成しなくても、Cisco IOS ルータの既存の RSA キーを使用できます。

- 「PKCS12 ファイルの RSA キーのエクスポートおよびインポート」（P.9）
- 「PEM 形式ファイルの RSA キーのエクスポートおよびインポート」（P.11）

PKCS12 ファイルの RSA キーのエクスポートおよびインポート

RSA キー ペアをエクスポートおよびインポートすることにより、ユーザは、セキュリティ クレデンシャルをデバイス間で転送できます。キー ペアを 2 台のデバイス間で共有すると、一方のデバイスが、もう一方のデバイスの機能を迅速かつトランスペアレントに引き継ぐことができます。

PKCS12 ファイルの RSA キーのエクスポートおよびインポートに関する前提条件

RSA キー ペアを「RSA キー ペアの生成」の作業に従って生成し、そのキー ペアに「エクスポート可能」のマークを付ける必要があります。

PKCS12 ファイルの RSA キーのエクスポートおよびインポートに関する制約事項

- システムを Cisco IOS Release 12.2(15)T 以降にアップグレードするまでは、ルータ上に存在する RSA キーをエクスポートできません。Cisco IOS ソフトウェアのアップグレード後、新しい RSA キーを生成し、このキーに「エクスポート可能」のラベルを付ける必要があります。
- サードパーティ製のアプリケーションで生成された PKCS12 ファイルをインポートする場合、PKCS12 ファイルには CA 証明書が含まれている必要があります。
- RSA キー ペアをすでにエクスポートし、ターゲット ルータにインポートした後で RSA キー ペアを再インポートする場合、RSA キー ペアをインポートするときに、**exportable** キーワードを指定する必要があります。
- ルータがインポートできる RSA キーの最大サイズは、2048 ビットです。

手順の概要

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export** *trustpointname* **pkcs12** *destination-url passphrase*
5. **crypto pki import** *trustpointname* **pkcs12** *source-url passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**crypto pki trustpoint** *name* **例：** `Router(config)# crypto pki trustpoint my-ca`	RSA キー ペアに関連付けるトラストポイント名を作成し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 2	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **例：** `Router(ca-trustpoint)# rsakeypair my-keys`	トラストポイントに使用するキー ペアを指定します。
ステップ 3	**exit** **例：** `Router(ca-trustpoint)# exit`	CA トラストポイント コンフィギュレーション モードを終了します。
ステップ 4	crypto pki export *trustpointname* pkcs12 *destination-url passphrase* **例：** `Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD`	トラストポイント名を使用して RSA キーをエクスポートします。 **（注）** 次のファイル システム タイプのいずれかを使用してトラストポイントをエクスポートできます。フラッシュ、ヌル、FTP、NVRAM、Remote File Copying（RCP）、SCP、システム、TFTP、Webflash、Xmodem、または Ymodem
ステップ 5	**crypto pki import** *trustpointname* **pkcs12** *source-url passphrase* **例：** `Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD`	ターゲット ルータに RSA キーをインポートします。
ステップ 6	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 7	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）ルータの RSA 公開キーを表示します。

PEM 形式ファイルの RSA キーのエクスポートおよびインポート

PEM ファイルの RSA キー ペアをエクスポートまたはインポートするには、次の作業を実行します。

PEM 形式ファイルの RSA キーのエクスポートおよびインポートに関する前提条件

RSA キー ペアを「RSA キー ペアの生成」の作業に従って生成し、そのキー ペアに「エクスポート可能」のマークを付ける必要があります。

PEM 形式ファイルの RSA キーのエクスポートおよびインポートに関する制約事項

- システムを Cisco IOS Release 12.3 (4)T 以降のリリースにアップグレードする前に、エクスポート可能のフラグを付けずに生成された RSA キーは、エクスポートおよびインポートできません。Cisco IOS ソフトウェアをアップグレードしたら、新しい RSA キーを生成する必要があります。
- ルータがインポートできる RSA キーの最大サイズは、2048 ビットです。

手順の概要

1. **crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**]
2. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
3. **crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*
4. **exit**
5. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	`crypto key generate rsa {usage-keys | general-keys} label key-label [exportable]`
	
	例：
	`Router(config)# crypto key generate rsa general-keys label mykey exportable`
		RSA キー ペアを生成します。
		PEM ファイルを使用するには、RSA キー ペアはエクスポート可能のラベルが付いている必要があります。

ステップ 2	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **例：** `Router(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	生成された RSA キー ペアをエクスポートします。 **ヒント** PEM ファイルは、必ず安全な場所に保管してください。たとえば、別のバックアップ ルータに保管することもできます。

ステップ 3	`crypto key import rsa key-label pem [usage-keys] {terminal | url url} [exportable] passphrase`
	
	例：
	`Router(config)# crypto key import rsa mycs2 pem url nvram: PASSWORD`
		生成された RSA キー ペアをインポートします。
		（注） キーを CA からエクスポート可能にしない場合は、そのキーをエクスポート不能のキー ペアとしてエクスポートしてから、CA に再度インポートしてください。このキーは削除できなくなります。

ステップ 4	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）ルータの RSA 公開キーを表示します。

ルータの秘密キーの暗号化およびロック

デジタル署名は、あるデバイスを別のデバイスに対して認証するために使用されます。デジタル署名を使用するには、プライベート情報（秘密キー）を、署名を提示しているデバイスに保管する必要があります。保管されたプライベート情報は、秘密キーを含むハードウェア装置を乗っ取ろうとする攻撃者に役立つことがあります。たとえば、攻撃者は、乗っ取ったルータを使用し、ルータに保管されている RSA 秘密キーを使用して、別のサイトへのセキュアな接続を開始する可能性があります。

（注） RSA キーはパスワードの復元操作中に失われます。パスワードを喪失した場合、パスワードの復元操作を実行すると、RSA キーは削除されます（この機能により、攻撃者がパスワードの復元を実行してキーを使用するのを防止します）。

攻撃者から秘密 RSA キーを保護するために、ユーザは、パスフレーズを使用して NVRAM に保管された秘密キーを暗号化できます。侵入を試みる攻撃者によってルータが乗っ取られた場合、ユーザは、秘密キーを「ロック」することもできます。これにより、稼動中ルータからの新しい接続の試行がブロックされ、ルータ内のキーが保護されます。

NVRAM に保存された秘密キーを暗号化しロックするには、次の作業を実行します。

前提条件

秘密キーを暗号化またはロックする前に、次の作業を実行する必要があります。

- RSA キー ペアを「RSA キー ペアの生成」の手順どおりに生成します。
- 必要に応じて、各ルータを認証し、CA サーバに登録できます。

（注） CA の登録中は、RSA キーのロックを解除する必要があります。ルータの秘密キーは認証時に使用されないため、CA でルータを認証している間、この秘密キーをロックできます。

秘密キーの暗号化およびロックに関する制約事項

下位互換性に関する制約事項

Cisco IOS Release 12.3(7)T よりも前のイメージは、暗号キーをサポートしません。暗号キーがルータによってすべて喪失されないように、Cisco IOS Release 12.3(7)T 以前のイメージを起動する前に、暗号化されていないキーだけが NVRAM に書き込まれていることを確認してください。

Cisco IOS Release 12.3(7)T 以前のイメージをダウンロードする必要がある場合は、ダウンロードされたイメージによって設定が上書きされないように、キーを復号化し、ただちに設定を保存してください。

アプリケーションとの相互作用

ルータの起動後、キーを手動で（**crypto key unlock rsa** コマンドを使用して）アンロックするまで、暗号キーは有効になりません。暗号化されているキー ペアによっては、この機能により、IP セキュリティ（IPsec）、SSH、SSL などのアプリケーションに悪影響が及ぶ可能性があります。つまり必要なキー ペアがアンロックされるまで、セキュア チャネル経由でのルータ管理ができない場合があります。

手順の概要

1. **crypto key encrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*
2. **exit**

3. **show crypto key mypubkey rsa**
4. **crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase*
5. **show crypto key mypubkey rsa**
6. **crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase*
7. **configure terminal**
8. **crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	`crypto key encrypt [write] rsa [name key-name] passphrase passphrase`
	
	例：
	`Router(config)# crypto key encrypt write rsa name pki.example.com passphrase password`
		RSA キーを暗号化します。
		このコマンドが発行されると、ルータはキーを引き続き使用でき、キーはアンロックされたままになります。
		（注） **write** キーワードを発行しない場合、設定を手動で NVRAM に書き込む必要があります。この作業を行わないと、次にルータをリロードしときに暗号化キーが消去されます。

ステップ 2	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 3	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）秘密キーが暗号化（保護）され、アンロックされていることを確認できます。 **（注）** このコマンドを使用して、キーの暗号化後、Internet Key Exchange（IKE; インターネット キー エクスチェンジ）および SSH などのアプリケーションが適切に機能していることを確認することもできます。
ステップ 4	**crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase* **例：** `Router# crypto key lock rsa name pki.example.com passphrase password`	（任意）暗号化された秘密キーを稼動中のルータ上でロックします。 **（注）** キーをロックした後は、そのキーを使用してピア デバイスにルータを認証できません。この動作により、ロックされているキーを使用する IPSec または SSL 接続はすべてディセーブルになります。 ロックされたキーに基づいて作成された既存の IPSec トンネルは閉じられます。 すべての RSA キーをロックすると、SSH は自動的にディセーブルになります。
ステップ 5	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）秘密キーが保護され、ロックされていることを確認できます。 このコマンドの出力では、IKE、SSH、SSL などのアプリケーションによって試行された接続の失敗も表示されます。

	コマンドまたはアクション	目的
ステップ 6	**crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase* **例：** `Router# crypto key unlock rsa name pki.example.com passphrase password`	（任意）秘密キーをアンロックします。 **（注）** このコマンドを発行すると、IKE トンネルを引き続き確立できます。
ステップ 7	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 8	**crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase* **例：** `Router(config)# crypto key decrypt write rsa name pki.example.com passphrase password`	（任意）暗号化されたキーを削除し、暗号化されていないキーだけを残します。 **（注）** **write** キーワードを使用すると、暗号化されていないキーはただちに NVRAM に保存されます。**write** キーワードを発行しない場合、設定を手動で NVRAM に書き込む必要があります。この作業を行わないと、次にルータをリロードしときにキーが暗号化したままになります。

RSA キー ペア設定の削除

次のいずれかの理由により、RSA キー ペアの削除が必要になる場合があります。

- 手動での PKI 操作およびメンテナンスの間に、古い RSA キーを削除して、新しいキーと交換できます。
- 既存の CA を置き換えた場合、新しい CA では、新たにキーを生成する必要があります。たとえば、必要なキーのサイズが組織によって異なることがあるため、古い 1024 ビット キーを削除し、新しい 2048 ビット キーを生成することが必要になる場合があります。
- IKEv1 および IKEv2 での署名確認の問題をデバッグできるように、ピア ルータの公開キーを削除できます。デフォルトでは、キーはトラストポイントに関連付けられた Certificate Revocation List（CRL; 証明書失効リスト）のライフタイムによってキャッシュされます。

すべての RSA キーまたはルータによって生成された指定の RSA キー ペアを削除するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key zeroize rsa** [*key-pair-label*]
4. **crypto key zeroize pubkey-chain** [*index*]
5. **exit**
6. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 3	`crypto key zeroize rsa [key-pair-label]`
	
	例：
	`Router(config)# crypto key zeroize rsa fancy-keys`
		ルータから RSA キー ペアを削除します。
		• *key-pair-label* 引数を指定していない場合、ルータによって生成された RSA キーはすべて削除されます。

ステップ 4	**crypto key zeroize pubkey-chain** [*index*] **例：** `Router(config)# crypto key zeroize pubkey-chain`	キャッシュからリモート ピアの公開キーを削除します。 （任意）特定の公開キーのインデックス エントリを削除するには、*index* 引数を使用します。インデックス エントリが指定されていない場合、すべてのエントリが削除されます。インデックス エントリに指定できる値の範囲は 1 ～ 65535 です。
ステップ 5	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**show crypto key mypubkey rsa** **例：** `Router# show crypto key mypubkey rsa`	（任意）ルータの RSA 公開キーを表示します。 このステップでは、RSA キー ペアが正常に生成されたことを確認できます。

RSA キー ペア展開での設定例

RSA キーの生成および指定例

次の例は、RSA キー ペア「exampleCAkeys」を生成し、指定する方法を示すサンプルのトラストポイント設定です。

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

RSA キーのエクスポートおよびインポート：例

PKCS12 ファイルの RSA キーのエクスポートおよびインポート：例

次の例では、RSA キー ペア「mynewkp」がルータ A で生成され、トラストポイント名「mynewtp」が作成されて、この RSA キー ペアに関連付けられています。このトラストポイントは、ルータ B にインポートできるように、TFTP サーバにエクスポートされます。ユーザは、ルータ B にトラストポイント「mynewtp」をインポートして、ルータ B に RSA キー ペア「mynewkp」をインポートしました。

ルータ A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair mykeys
 exit

crypto pki export mytp pkcs12 flash:myexport companyname
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
Feb 18 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

ルータ B

```
crypto pki import mynewtp pkcs12 flash:myexport companyname
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.

!
Feb 18 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

PEM ファイルの RSA キーの生成、エクスポート、インポート、および検証例

次の例では、キーを生成してエクスポートし、戻す（インポートする）方法、および RSA キー ペア「mycs」のステータスを確認する方法を示します。

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mycs exportable
The name for the keys will be: mycs

Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
```

```
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto key export rsa mycs pem url nvram:3des PASSWORD

% Key name:mycs
Usage:General Purpose Key
Exporting public key...
Destination filename [mycs.pub]?
Writing file to nvram:mycs.pub
Exporting private key...
Destination filename [mycs.prv]?
Writing file to nvram:mycs.prv
!
! Import the key as a different name.
!
Router(config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD

% Importing public key or certificate PEM file...
Source filename [mycs.pub]?
Reading file from nvram:mycs.pub
% Importing private key PEM file...
Source filename [mycs.prv]?
Reading file from nvram:mycs.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2003
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2003
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
```

PEM ファイルからのルータ RSA キー ペアおよび証明書のエクスポート例

次の例では、トラストポイント「mycs」に関連付けられているルータの RSA キー ペア「aaa」と PEM ファイル形式の証明書を生成し、これらをエクスポートする方法を示します。また、この例では、他の SSL および SSH アプリケーションで使用される PEM 形式ファイルも示します。このファイルには、Base 64 暗号化データの前後の PEM 境界が含まれています。

```
Router(config)# crypto key generate rsa general-keys label aaa exportable
```

```
The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs
Router(ca-trustpoint)# enrollment url http://mycs
Router(ca-trustpoint)# rsakeypair aaa
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.

Password:
Re-enter password:

% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be:host.example.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157

00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

Router(config)# crypto ca export aaa pem terminal 3des password
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----

% Key name:aaa
Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,ED6B210B626BC81A

Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbAAUpGk7VnPCT87
<snip>
kLCOtxzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----

% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfigAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
```

```
<snip>
6xlBaIsuMxnHmr89KkKkYlU6
-----END CERTIFICATE-----
```

PEM ファイルからのルータ RSA キー ペアおよび証明書のインポート：例

次の例では、TFTP を使用して、PEM ファイルから RSA キー ペアと証明書をトラストポイント「ggg」にインポートする方法を示します。

```
Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password
% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via Ethernet0):!
[OK - 1082 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via Ethernet0):!
[OK - 573 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via Ethernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#
```

ルータの秘密キーの暗号化およびロック：例

- 「暗号キーの設定および検証例」（P.20）
- 「ロックされたキーの設定および確認例」（P.21）

暗号キーの設定および検証例

次の例では、RSA キー「pki-123.example.com」を暗号化する方法を示します。その後、**show crypto key mypubkey rsa** コマンドを発行して、RSA キーが暗号化（保護）され、アンロックされていることを確認します。

```
Router(config)# crypto key encrypt rsa name pki-123.example.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa

% Key pair was generated at:00:15:32 GMT Jun 25 2003
Key name:pki-123.example.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***
Key is not exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFEE340 0EC1E62B D40E7DCC
23C4D09E
03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001
```

```
% Key pair was generated at:00:15:33 GMT Jun 25 2003
Key name:pki-123.example.com.server
Usage:Encryption Key
Key is exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00D3491E 2A21D383
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FFF5757
3A41CE04 FDCB40A4 B9C68B4F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001
Router#
```

ロックされたキーの設定および確認例

次の例では、RSA キー「pki-123.example.com」をロックする方法を示します。その後、**show crypto key mypubkey rsa** コマンドを発行して、RSA キーが保護（暗号化）され、アンロックされていることを確認します。

```
Router# crypto key lock rsa name pki-123.example.com passphrase password
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki-123.example.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

関連情報

RSA キー ペアを生成したら、トラストポイントを設定する必要があります。すでにトラストポイントを設定している場合は、ルータを認証し、PKI に登録する必要があります。登録に関する情報については、「PKI の証明書登録の設定」を参照してください。

その他の参考資料

関連資料

内容	参照先
PKI の概要（RSA キー、証明書登録、および CA を含む）	「*Cisco IOS PKI Overview: Understanding and Planning a PKI*」
PKI コマンド：完全なコマンドの構文、コマンド モード、デフォルト、使用上の注意事項、例	『*Cisco IOS Security Command Reference*』

規格

規格	タイトル
なし	—

MIB

MIB	MIB リンク
なし	選択したプラットフォーム、Cisco IOS リリース、および機能セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 2409	『*The Internet Key Exchange (IKE)*』
RFC 2511	『Internet X.509 Certificate Request Message Format』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。 以下を含むさまざまな作業にこの Web サイトが役立ちます。 ・テクニカル サポートを受ける ・ソフトウェアをダウンロードする ・セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける ・ツールおよびリソースへアクセスする - Product Alert の受信登録 - Field Notice の受信登録 - Bug Toolkit を使用した既知の問題の検索 ・Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する ・トレーニング リソースへアクセスする ・TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

PKI 内の RSA キーに関する機能情報

表 1 に、この機能のリリース履歴を示します。

ここに記載されていないこのテクノロジーの機能情報については、「Implementing and Managing PKI Features Roadmap」を参照してください。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェア イメージのサポート情報を検索できます。Cisco Feature Navigator を使用すると、ソフトウェア イメージがサポートする特定のソフトウェア リリース、機能セット、またはプラットフォームを確認できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

（注） 表 1 には、一連のソフトウェア リリースのうち、特定の機能が初めて導入されたソフトウェア リリースだけが記載されています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 1　PKI 内の RSA キーに関する機能情報

機能名	ソフトウェア リリース	機能設定情報
Cisco IOS 4096 ビット公開キーのサポート	12.4(12)T	この機能により、Cisco IOS 4096 ビット ピア公開キーがサポートされます。 この機能に関する詳細については、次の項を参照してください。 • 「RSA キーの概要」
RSA キーのエクスポートおよびインポート	12.2(15)T	この機能では、RSA キーをエクスポートし、インポートすることにより、デバイス間でセキュリティ クレデンシャルを転送できます。キー ペアを 2 台のデバイス間で共有すると、一方のデバイスが、もう一方のデバイスの機能を迅速かつトランスペアレントに引き継ぐことができます。 この機能に関する詳細については、次の各項を参照してください。 • 「エクスポート可能な RSA キーのメリット」 • 「PKCS12 ファイルの RSA キーのエクスポートおよびインポート」 この機能により、**crypto ca export pkcs12**、**crypto ca import pkcs12**、**crypto key generate rsa**（**IKE**）コマンドが導入または変更されました。

表 1　PKI 内の RSA キーに関する機能情報 （続き）

機能名	ソフトウェア リリース	機能設定情報
RSA キー ペアおよび PEM 形式証明書のインポート	12.3(4)T	この機能を使用すると、PEM 形式ファイルを使用して、RSA キー ペアをインポートまたはエクスポートできます。PEM 形式のファイルを使用すると、新しいキーを生成しなくても、既存の RSA キー ペアを Cisco IOS ルータで直接使用できます。 この機能に関する詳細については、次の各項を参照してください。 • 「エクスポート可能な RSA キーのメリット」 • 「PEM 形式ファイルの RSA キーのエクスポートおよびインポート」 この機能により、次のコマンドが導入されました。**crypto ca export pem**、**crypto ca import pem**、**crypto key export pem**、**crypto key import pem**
複数の RSA キー ペアのサポート	12.2(8)T	この機能では、複数の RSA キー ペアを保持するようにルータを設定できます。したがって、Cisco IOS ソフトウェアはアイデンティティ証明書ごとに異なるキー ペアを維持できます。 この機能に関する詳細については、次の各項を参照してください。 • 「ルータに複数の RSA キーを保管する理由」 • 「RSA キー ペアとトラストポイントの証明書の管理」 この機能により、**crypto key generate rsa**、**crypto key zeroize rsa**、**rsakeypair** コマンドが導入または変更されました。
秘密キー保管の保護	12.3(7)T	この機能により、ユーザは、Cisco IOS ルータで使用される RSA 秘密キーを暗号化およびロックできます。これにより、秘密キーの不正使用を防止できます。 この機能に関する詳細については、次の項を参照してください。 • 「ルータの秘密キーの暗号化およびロック」 この機能により、次のコマンドが導入または変更されました。**crypto key decrypt rsa**、**crypto key encrypt rsa**、**crypto key lock rsa**、**crypto key unlock rsa**、**show crypto key mypubkey rsa**

表 1　PKI 内の RSA キーに関する機能情報 （続き）

機能名	ソフトウェア リリース	機能設定情報
ソフトウェア暗号エンジンサポートでの RSA 4096 ビット キー生成	15.1(1)T	**crypto key generate rsa** コマンドの **modulus** キーワードの値の範囲は、360 ～ 2048 ビットから 360 ～ 4096 ビットに拡張されました。
IOS PKI のパフォーマンス モニタリングと最適化	15.1(3)T	IOS のパフォーマンス モニタリングと最適化機能は、PKI サブシステム内のパフォーマンスの特徴を示し、PKI のパフォーマンス関連の問題のデバッグや分析を行う方法を提供します。この機能については、『IOS Performance Monitoring and Optimization』で詳しく説明しています。 また、この機能には、このマニュアルでも参照可能な次の拡張機能も含まれています。 • 所有者別名（subjectAltName）フィールドにトラストポイントの名前を含むルータに、自己署名トラストポイント証明書を作成できます。 • IKE バージョン 1 および IKE バージョン 2 の署名確認の問題をデバッグしたり、この処理を実行した結果を受けてピア ルータのパフォーマンスを最適化するために、ピア ルータの公開キーを削除できます。 これらの機能については、次の項で参照できます。 • 「RSA キー ペアの生成」（P.4） • 「RSA キー ペア設定の削除」（P.15） 次のコマンドがこの機能で導入または変更されました。**crypto key zeroize pubkey-chain**、**subject-alt-name**