公開キー インフラストラクチャ構成ガイド

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

第 1 章

最初にお読みください

Cisco IOS XE 16 に関する重要な情報

Cisco IOS XE リリース 3.7.0E（Catalyst スイッチング用）および Cisco IOS XE リリース 3.17S（アクセスおよびエッジルーティング用）の 2 つのリリースは、コンバージドリリースの 1 つのバージョン - Cisco IOS XE 16 - に統合されました。この 1 つのリリースでスイッチングおよびルーティング ポートフォリオのアクセスおよびエッジ製品を幅広くカバーしています。

（注） 技術構成ガイドの機能情報の表に、機能の導入時期を記載しています。他のプラットフォームがその機能をサポートした時期については、記載があるものも、ないものもあります。特定の機能が、使用しているプラットフォームでサポートされているかどうかを判断するには、製品のランディング ページに掲載された技術構成ガイドを参照してください。技術構成ガイドが製品のランディング ページに表示されると、その機能が該当のプラットフォームでサポートされているかどうかが示されます。

第 2 章

Cisco IOS XE PKI の概要 PKI の理解と計画

Cisco IOS XE 公開キー インフラストラクチャ（PKI）には、IP Security（IPSec）、セキュア シェル（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機能があります。

このマニュアルでは、PKI を理解、計画、実装するために必要な概念を確認、説明します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

Cisco IOS XE PKI の情報

Cisco IOS XE PKI とは

PKI は以下のエンティティで構成されています。

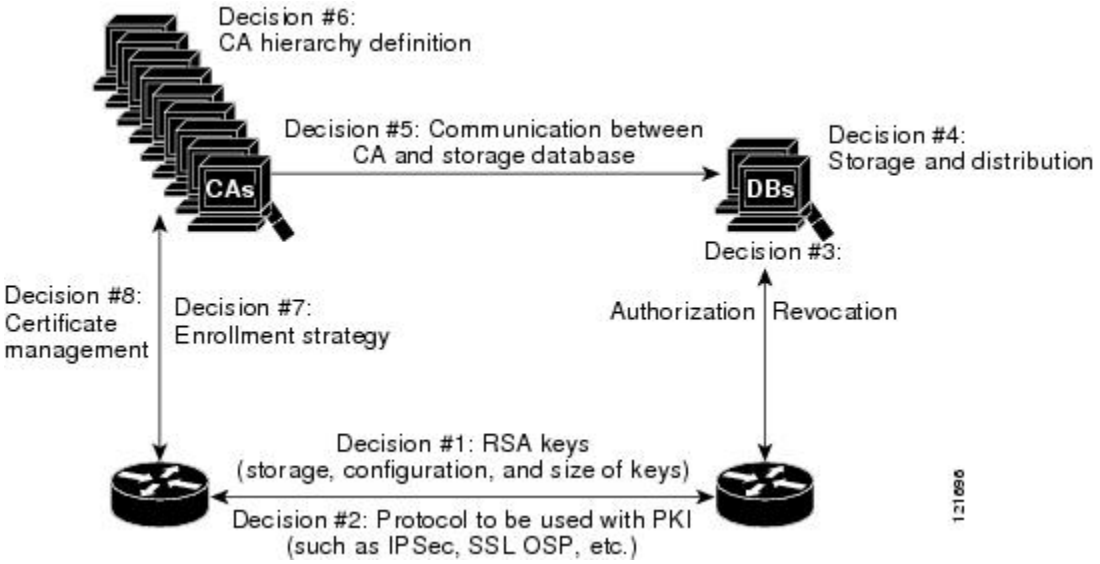
- セキュアなネットワークで通信する複数のピア
- 証明書を発行および維持する認証局（CA）を最低 1 つ
- デジタル証明書（証明書の有効期間、ピアの ID 情報、セキュアな通信に使用する暗号キー、CA 発行のシグニチャなどで構成）
- 登録要求を処理し CA の負荷を軽減する登録局（RA）（任意）
- 証明書失効リスト（CRL）を配信するメカニズム（Lightweight Directory Access Protocol（LDAP）、HTTP など）

PKI を使用すると、セキュアなデータ ネットワークで暗号化情報と ID 情報を配信、管理、失効するためのスケーラブルでセキュアなメカニズムを実現できます。セキュアな通信に関係するエンティティ（人物またはデバイス）はすべて、あるプロセスを経て PKI に登録されます。そのプロセスでは、エンティティが RSA（Rivest、Shamir、Adelman）キーのペア（秘密キーが 1 つ、公開キーが 1 つ）を生成し、信頼されているエンティティ（CA またはトラストポイントともいいます）でキーの ID を確認します。

各エンティティが PKI に登録されると、PKI のすべてのピア（エンド ホストともいいます）は、CA が発行したデジタル証明書を付与されます。セキュアな通信セッションをネゴシエーションする必要があるときは、ピアはデジタル証明書を交換します。ピアは証明書内の情報を基に他のピアの ID を確認し、証明書内の公開キーを使って、暗号化されたセッションを確立します。

PKI はさまざまな方法で計画、設定できますが、次の図に、PKI を構成する主なコンポーネントと、PKI で実行される各選択の順番を示します。図をアプローチとして推奨していますが、別の方法で PKI を設定してもかまいません。

図 *1*：*PKI* の設定方法の決定



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。

CA とは

CA（トラストポイントともいいます）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。証明書要求の管理や証明書発行などのサービスにより、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。

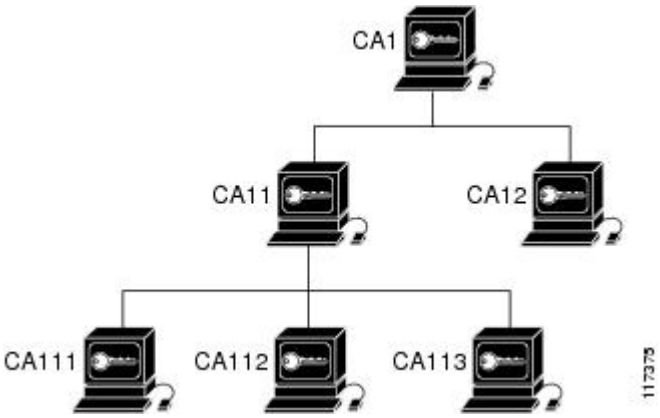
CA は、サードパーティの CA ベンダーが提供する CA を使用するか、内部の CA、つまり Cisco IOS 証明書サーバを使用します。

階層型 PKI：複数の CA

PKI は、複数の CA をサポートするために階層型フレームワーク内に設定できます。階層の最上位にはルート CA があり、自己署名証明書を保持しています。階層全体の信頼性は、ルート CA の RSA キー ペアから導出されます。階層構造内の下位 CA は、ルート CA または別の下位 CA に登録できます。どちらの方法で登録するかによって、CA の複数階層の設定方法が決まります。階層型 PKI 内では、登録されているすべてのピアが信頼できるルート CA 証明書または共通の下位 CA を共有している場合、証明書を相互に検証できます。

次の表 は、3 段の階層の CA 間の登録関係を示したものです。

図 *2*：*3* 段の *CA* 階層のサンプル トポロジ



各 CA が 1 つのトラストポイントに対応します。たとえば、CA11 および CA12 は従属 CA で、CA1 が発行した CA 証明書を保持しています。CA111、CA112、CA113 も従属 CA ですが、その CA 証明書を発行したのは CA11 です。

複数 CA を使用する場合

複数 CA を使用することにより、柔軟性および信頼性が向上します。たとえば、ルート CA を本社オフィスに配置し、下位 CA をブランチ オフィスに配置できます。また、CA ごとに異なる許可ポリシーを実行できるため、階層構造内の、ある CA では各証明書要求を手動で許可する必要があるように、別の CA では証明書要求を自動的に許可するように設定できます。

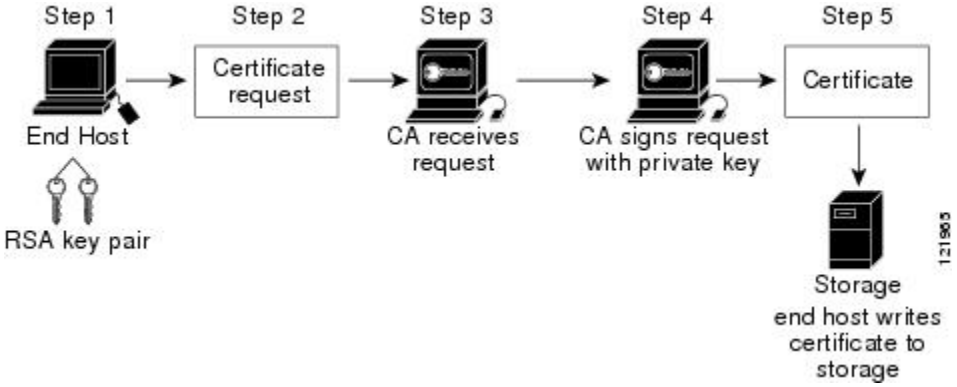
少なくとも 2 階層の CA が推奨されるシナリオは、次のとおりです。

- 多数の証明書が失効し、再発行される大規模かつ非常にアクティブなネットワーク。複数の階層を使用することにより、CA は CRL のサイズを制御しやすくなります。
- オンライン登録方式を使用するときに、ルート CA をオフラインのままにできる場合（従属 CA の証明書の発行を除く）。このシナリオでは、ルート CA のセキュリティが向上します。

証明書の登録：登録の動作

証明書の登録は、CA から証明書を取得するプロセスです。PKI に加わるエンド ホストは、それぞれ証明書を取得する必要があります。証明書の登録は、証明書を要求しているエンド ホストと CA との間で行われます。次の表および手順によって、証明書の登録プロセスを説明します。

図 *3*：証明書の登録プロセス



1. エンド ホストが RSA キーのペアを生成します。
2. エンド ホストが証明書要求を生成し、CA（または使用可能な場合は RA）に送ります。
3. CA が証明書登録要求を受け取ります。ネットワークの設定によって、次のいずれかになります。
   1. 要求の承認に手動による操作が必要。
   2. CA に証明書を自動で要求するようにエンド ホストが設定されている。これにより、登録要求が CA サーバに送信されたときのオペレータによる手動操作は不要になります。

（注） CA に証明書を自動で要求するようにエンドホストを設定するには、別の認証メカニズムが必要になります。

1. 要求が承認されると、CA は自分の秘密キーを使って要求に署名し、処理の終わった証明書をエンド ホストに戻します。
2. エンド ホストは、証明書を NVRAM などの保管領域に書き込みます。

Secure Device Provisioning による証明書登録

Secure Device Provisioning（SDP）は、Cisco IOS XE クライアントと Cisco IOS 証明書サーバなど、2 つのエンド デバイス間で PKI を簡単に配置できる、Web ベースの証明書登録インターフェイスです。

SDP（Trusted Transitive Introduction（TTI）とも呼ばれている）は、新しいネットワーク デバイスと VPN 間といった 2 つのエンド エンティティ間の双方向導入を実現する通信プロトコルです。SDP では次の 3 つのエンティティが関係します。

- イントロデューサ：ペティショナをレジストラに紹介する、相互に信頼できるデバイス。イントロデューサは、システム管理者などのデバイス ユーザの場合があります。
- ペティショナ：セキュアなドメインに参加した新しいデバイス。
- レジストラ：申請者を承認する証明書サーバなどのサーバ。

SDP は Web ブラウザを使い、ようこそ、紹介、完了の 3 つの段階で実装します。各段階は、Web ページを通してユーザに表示されます。

証明書の失効：失効する理由

各ピアが正常に PKI に登録されると、ピアは互いにセキュアな接続を行うためのネゴシエーションを開始できます。そのためにピアは確認に自分の証明書を提示し、失効のチェックを受けます。ピアは、通信相手のピアの証明書が、認証済みの CA によって発行された証明書であることを確認すると、CRL サーバまたは OCSP（Online Certificate Status Protocol）サーバをチェックし、証明書を発行した CA によって証明書が失効になっていないことを確認します。証明書には通常、証明書分散ポイント（CDP）が URL 形式で含まれています。Cisco IOS ソフトウェアはこの CDP を使用して、CRL の場所の特定と取得を行います。CDP サーバが応答しないと Cisco IOS ソフトウェアはエラーを生成し、ピアの証明書が拒否される場合があります。

PKI の計画

PKI の計画では、PKI の計画, （8 ページ） のそれぞれの PKI コンポーネントの要件と予定の用途を評価する必要があります。ユーザ（またはネットワーク管理者）の方で十分に PKI を計画してから、PKI の設定を始めること推奨します。

PKI の計画では検討すべきアプローチがいくつかありますが、このマニュアルでは、ピアツーピアの通信から始めて、PKI の計画, （8 ページ） のような設定に進みます。ただし、ユーザまたはネットワーク管理者が PKI の計画を選択するときは、特定の決定が PKI の他の決定に影響することを理解しておいてください。たとえば、登録および展開をどのようにするかによって、CA の階層の計画が変わってくる場合があります。このため、PKI 内の各コンポーネントがどのように機能するか、また、特定のコンポーネントのオプションが、計画プロセスで行った決定によってどのように変わるかを理解することが重要です。

次の作業

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。

PKI の理解と計画に関する追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
PKI およびセキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』
USB トークンによる RSA 処理：初期の自動登録用の USB トークンにおける RSA キーの使用	「*PKI* の証明書登録の設定」
USB トークンによる RSA 処理：USB トークンを使用するメリット	「*PKI* クレデンシャルの保存」
証明書サーバクライアント証明書の登録、自動登録、および自動ロールオーバー	「*PKI* の証明書登録の設定」
USB トークンの設定および USB トークンへのロギング	「*PKI* クレデンシャルの保存」
Web を使用した証明書登録	「*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*」
PEM 形式ファイル内の RSA キー	「*PKI* 内での *RSA* キーの展開」
証明書失効メカニズムの選択	「*PKI* での証明書の許可および失効の設定」
推奨暗号化アルゴリズム	『Next Generation Encryption』

MIB

MIB	MIB のリンク
• PKI MIB	選択したプラットフォーム、Cisco IOS ソフトウェア リリース、およびフィーチャ 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

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

用語集

CDP：証明書分散ポイント（CDP）。デジタル証明書内のフィールドで、証明書の CRL の取り出し方法を記述した情報が含まれています。最も一般的な CDP としては HTTP や LDAP の URL があります。CDP には、他の種類の URL または LDAP のディレクトリ指定が含まれている場合もあります。それぞれの CDP には、URL またはディレクトリの指定が 1 つ含まれています。

証明書：ユーザ名またはデバイス名を公開キーにバインドする電子ドキュメント。証明書は、一般的にデジタル署名を確認するために使用されます。

CRL：Certificate Revocation List（証明書失効リスト）。失効した証明書のリストが含まれる電子ドキュメントです。CRL は、証明書を発行した CA によって作成され、デジタル署名されます。CRL には、証明書の発行日と失効日が含まれています。現行の CRL が失効すると、新しい CRL が発行されます。

CA：Certification Authority（認証局）証明書要求の管理と、関係する IPSec ネットワーク デバイスへの証明書の発行を担当しているサービス。このサービスは、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。

ピア証明書：ピアが提示する証明書のことで、ピアの公開キーが含まれており、トラストポイント CA が署名します。

PKI：Public Key Infrastructure（公開キー インフラストラクチャ）。セキュアに設定された通信に使用されているネットワーク コンポーネントの暗号キーと ID 情報を管理するシステムです。

RA：登録局（RA）。CA のプロキシとして機能するサーバで、CA がオフラインのときでも CA の機能を継続できます。RA は CA サーバ上に設定するのが通常ですが、別アプリケーションとして、稼働のための別デバイスを必要とする場合もあります。

RSA キー：公開キー暗号化システムで、Ron Rivest（ロナルド・リベスト）、Adi Shamir（アディ・シャミア）、Leonard Adleman（レオナルド・エーデルマン）の 3 人によって開発されました。ルータの証明書を取得するには、RSA キーのペア（公開キーと秘密キー）が必要です。

第 3 章

PKI 内での RSA キーの展開

この章では、公開キーインフラストラクチャ（PKI）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。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

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。

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。

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。

RSA キー ペアとトラストポイントとの連携方法

トラストポイント（認証局（CA）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。

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。

では、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

プライバシー エンハンスト メール（PEM）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、セ

キュア ソケット レイヤ（SSL）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、「nonexportable」と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可能なキー ペアに変換するには、キー ペアをエクスポートし、「exportable」というキーワードを指定しないで再び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. **configureterminal**
3. **cryptokeygeneratersa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label***key-label*] [**exportable**] [**modulus***modulus-size*] [**storage***devicename:*] [**on***devicename:*]
4. **exit**
5. **showcryptokeymypubkeyrsa**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptokeygeneratersa** [**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 2048`	（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、**cryptopkiserver***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、ルータの完全修飾ドメイン名（FQDN）であるデフォルト値が使用されます。 **noshutdown** 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、**cryptocaexportpkcs12** 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 360 ～ 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。この装置にはユニバーサルシリアルバス（USB）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。

	コマンドまたはアクション	目的
ステップ 5	**showcryptokeymypubkeyrsa** 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。

はじめる前に

「RSA キー ペアの生成」の作業どおりに RSA キー ペアを生成しておく必要があります。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint***name*
4. **rsakeypair***key-label* [*key-size* [*encryption-key-size*]]
5. **enrollmentselfsigned**
6. **subject-alt-name***name*
7. **exit**
8. **cyptopkienroll***name*
9. **exit**
10. **showcryptokeymypubkeyrsa**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptopkitrustpoint***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-enrollregenerate** 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、完全修飾ドメイン名（FQDN）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。推奨されるキー サイズは 2048 ビットです。 • （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	**enrollmentselfsigned** 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。

	コマンドまたはアクション	目的
		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 **enrollmentselfsigned** 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。

ステップ 8	**cyptopkienroll***name*
	
	例：
	
	`Router(config)# cypto pki enroll TESTCA`
	
	例：
	
	`% Include the router serial number in the subject name? [yes/no]: no`
	
	例：
	
	`% Include an IP address in the subject name? [no]:`
	
	例：
	
	`Generate Self Signed Router Certificate? [yes/no]: yes`
	
	例：
	
	`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） **cryptopkitrustpoint** 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。

	コマンドまたはアクション	目的
ステップ 10	**showcryptokeymypubkeyrsa** 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)#hash sha256
Router(ca-trustpoint)#rsakeypair testca-rsa-key 2048
Router(ca-trustpoint)#exit
Router(config)#crypto pki enroll TESTCA
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created

Router(config)#
Router(config)#exit
Router#
```

次の証明書が作成されます。

```
Router#show crypto pki certificate verbose Router Self-Signed Certificate
   Status: Available
   Version: 3
   Certificate Serial Number (hex): 01
   Certificate Usage: General Purpose
   Issuer:
     hostname=Router.cisco.com
   Subject:
     Name: Router.cisco.com
     hostname=Router.cisco.com
   Validity Date:
     start date: 11:41:50 EST Aug 13 2012
     end   date: 19:00:00 EST Dec 31 2019
   Subject Key Info:
     Public Key Algorithm: rsaEncryption
     RSA Public Key: (2048 bit)
   Signature Algorithm: SHA256 with RSA Encryption
   Fingerprint MD5: CA92D937 593BF19A 5B7F8466 F554D631
   Fingerprint SHA1: 57A9D411 2DDFAC81 68260F2F C6C8D7CF 4833F3E9
   X509v3 extensions:
     X509v3 Subject Key ID: 44340F76 A6B8DC37 80724650 0672875F 741D518C
     X509v3 Basic Constraints:
         CA: TRUE
     X509v3 Authority Key ID: 44340F76 A6B8DC37 80724650 0672875F 741D518C
     Authority Info Access:
   Associated Trustpoints: TESTCA

-----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

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。

はじめる前に

「RSA キー ペアの生成」で指定した作業のとおりに RSA キー ペアを生成して「exportable」とマークを付ける必要があります。

（注）

- 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「exportable」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. **cryptopkitrustpoint***name*
2. **rsakeypair***key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export***trustpointname***pkcs12***destination-url***password***password-phrase*
5. **crypto pki import***trustpointname***pkcs12***source-url***password***password-phrase*
6. **exit**
7. **showcryptokeymypubkeyrsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**cryptopkitrustpoint***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***password***password-phrase* 例： `Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123`	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。 • *trustpointname* 引数は、ユーザがエクスポート予定の証明書を発行するトラストポイントの名前を入力します。PKCS12 ファイルをエクスポートする場合、トラストポイント名はRSAキー名です。 • *destination-url* 引数は、ユーザが RSA キー ペアをインポートする PKCS12 ファイルのファイル システム ロケーションを入力します。詳細については、「crypto pki export pkcs12

	コマンドまたはアクション	目的
		password」コマンドページを参照してください。 • *password -phrase* 引数は、エクスポート用にPKCS12ファイルを暗号化するのに入力する必要があります。
ステップ 5	**crypto pki import***trustpointname***pkcs12***source-url***password***password-phrase* 例： `Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123`	ターゲット ルータに RSA キーをインポートします。 • *trustpointname* 引数は、ユーザがエクスポートまたはインポート予定の証明書を発行するトラストポイントの名前を入力します。インポートすると、トラストポイントが RSA キー名になります。 • *source-url* 引数は、ユーザが RSA キーペアをエクスポートする PKCS12 ファイルのファイル システム ロケーションを指定します。詳細については、「crypto pki import pkcs12 password」コマンド ページを参照してください。 • *password -phrase* は、RSA キーがインポートされる場合、暗号化を元に戻すために入力する必要があります。
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	**showcryptokeymypubkeyrsa** 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。

はじめる前に

「RSA キー ペアの生成」で指定した作業のとおりに RSA キー ペアを生成して「exportable」とマークを付ける必要があります。

（注）
- 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。

（注）セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイト ペーパーを参照してください。

手順の概要

1. **crypto key generate rsa** {**usage-keys** | **general-keys**} **label***key-label* [**exportable**]
2. **crypto pki export***trustpoint***pem** {**terminal** | **url***destination-url*} {**3des** | **des**} **password***password-phrase*
3. **crypto pki import***trustpoint***pem** [**check** | **exportable** | *usage-keys*] {**terminal** | **url***source-url*} **password***password-phrase*
4. **exit**
5. **showcryptokeymypubkeyrsa**

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 pki export***trustpoint***pem** {**terminal** | **url***destination-url*} {**3des** | **des**} **password***password-phrase*	PEM形式ファイルのトラストポイントと関連付けられた証明書およびRSAキーをエクスポートします。 • エクスポートした証明書および RSA キー ペアに関連付けられた *trustpoint* 名を入力します。トラストポイント名は、**crypto pki**

	コマンドまたはアクション	目的
	例： `Router(config)# crypto pki export mycs pem url nvram: 3des password mypassword123`	**trustpoint** コマンドを使用して指定された名前と一致する必要があります。 • **terminal** キーワードを使用し、コンソール端末に PEM 形式で表示される証明書および RSA キー ペアを指定します。 • **url** キーワードおよび *destination -url* 引数を使用し、ルータが証明書および RSA キー ペアをエクスポートするファイル システムの URL を指定します。 • （任意）**3des** キーワードは、Triple Data Encryption Standard（3DES）暗号化アルゴリズムを使用してトランスポイントをエクスポートします。 • （任意）**des** キーワードは、DES 暗号化アルゴリズムを使用してトランスポイントをエクスポートします。 • *password-phrase* 引数を使用し、インポート用の PEM ファイルの暗号化に使用する暗号化パスワード フレーズを指定します。 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 pki import***trustpoint***pem** [**check** | **exportable** | *usage-keys*] {**terminal** | **url***source-url*} **password***password-phrase* 例： `Router(config)# crypto pki import mycs2 pem url nvram: password mypassword123`	PEM 形式ファイルからにトラストポイントに証明書および RSA キーをインポートします。 • インポートした証明書および RSA キーペアに関連付けられた *trustpoint* 名を入力します。トラストポイント名は、**crypto pki trustpoint** コマンドを使用して指定された名前と一致する必要があります。 • （任意）**check** キーワードを使用し、古い証明書を許可しないように指定します。 • （任意）**exportable** キーワードを使用し、インポートした RSA キー ペアをルータなどの別のシスコ デバイスに再びエクスポートできるように指定します。 • （オプション）*usage-keys* 引数を使用し、1 つの汎用目的キー ペアの代わりに、2 つの RSA 特殊用途キー ペア（暗号化ペア 1 つとシグニチャ ペア 1 つ）がインポートされるように指定します。 • *source-url* 引数を使用し、ルータが証明書および RSA キー ペアをインポートするファイル システムの URL を指定します。 • *password-phrase* 引数を使用し、インポート用の PEM ファイルの暗号化に使用する暗号化パスワード フレーズを指定します。

	コマンドまたはアクション	目的
		（注） パスワード 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。 （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	**showcryptokeymypubkeyrsa** 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、稼働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。

（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。

はじめる前に

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, （16 ページ）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） **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. **showcryptokeymypubkeyrsa**
4. **cryptokeylockrsaname***key-name*] **passphrase** *passphrase*
5. **showcryptokeymypubkeyrsa**
6. **cryptokeyunlockrsa** [**name** *key-name*] **passphrase** *passphrase*
7. **configureterminal**
8. **cryptokeydecrypt** [**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するときに暗号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	**showcryptokeymypubkeyrsa** 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、インターネット キー交換（IKE）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	**cryptokeylockrsaname***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働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	**showcryptokeymypubkeyrsa** 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。
ステップ 6	**cryptokeyunlockrsa** [**name** *key-name*] **passphrase** *passphrase* 例： `Router# crypto key unlock rsa name`	（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。

	コマンドまたはアクション	目的
	`pki.example.com passphrase password`	
ステップ 7	**configureterminal** 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	**cryptokeydecrypt** [**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た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証明書失効リスト（CRL）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. **configureterminal**
3. crypto key zeroize rsa [key-pair-label]
4. **cryptokeyzeroizepubkey-chain** [*index*]
5. **exit**
6. **showcryptokeymypubkeyrsa**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptokeyzeroizepubkey-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。

	コマンドまたはアクション	目的
ステップ 6	**showcryptokeymypubkeyrsa** 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。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。ユーザがルータ B にトランスポイント「mynewtp」をインポートすると、ルータ B に RSA キー ペア「mynewkp」がインポートされます。

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]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair mykeys
 exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
```

```
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

ルータ B

```
crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.
!
July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

PEM 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「mytp」の生成、エクスポート、インポートを示し、そのステータスを確認します。

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mytp exportable

The name for the keys will be: mytp
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]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto pki export mytp pem url nvram:mytp 3des password mypassword123

% Key name:mytp
Usage:General Purpose Key
Exporting public key...
Destination filename [mytp.pub]?
Writing file to nvram:mytp.pub
Exporting private key...
Destination filename [mytp.prv]?
Writing file to nvram:mytp.prv
!
! Import the key as a different name.
!
Router(config)# crypto pki import mytp2 pem url nvram:mytp2 password mypassword123

% Importing public key or certificate PEM file...
Source filename [mytp2.pub]?
Reading file from nvram:mytp2.pub
% Importing private key PEM file...
Source filename [mytp2.prv]?
Reading file from nvram:mytp2.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 2011
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 2011
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」と関連付けられた PEM ファイルに RSA キーペア「aaa」とルータの証明書を生成およびエクスポートする方法について示します。また、この例では、Base64 符号化データの前後の PEM 境界を含む PEM 形式ファイルも示します。このファイルは他の SSL と SSH アプリケーションで使用されます。

```
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

次の例に、RSA キー「pki-123.example.com」を暗号化する方法について示します。そのため、**show crypto key mypubkey rsa** コマンドを発行して、RSA キーが暗号化（保護）またはロック解除されているかを確認します。

```
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

次の例に、キー「pki-123.example.com」をロックする方法について示します。そのため、**show crypto key mypubkey rsa** コマンドを発行して、キーが保護（暗号化）またはロックされているかを確認します。

```
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
```

次の作業

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

関連項目	マニュアル タイトル
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*』
推奨暗号化アルゴリズム	『Next Generation Encryption』

MIB

MIB	MIB のリンク
なし	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

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	機能の設定情報
Cisco IOS 4096 ビット公開キーのサポート	Cisco IOS XE Release 2.4	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。

機能名	ソフトウェア リリース	機能の設定情報
RSA キーのエクスポートおよびインポート	Cisco IOS XE Release 2.1	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。 次のコマンドがこの機能で導入または変更されました。 **cryptocaexportpkcs12**、**cryptocaimportpkcs12**、**cryptokeygeneratersa(IKE)**
RSA キー ペアおよび PEM 形式証明書のインポート	Cisco IOS XE Release 2.1	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。 次のコマンドがこの機能で導入されました。 **cryptocaexportpem**、**cryptocaimportpem**、**cryptokeyexportpem**、**cryptokeyimportpem**

機能名	ソフトウェア リリース	機能の設定情報
複数の RSA キー ペアのサポート	Cisco IOS XE Release 2.1	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。 次のコマンドがこの機能で導入または変更されました。 **cryptokeygeneratersa**、**cryptokeyzeroizersa**、**rsakeypair**
秘密キー保管の保護	Cisco IOS XE Release 2.1	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。 次のコマンドがこの機能で導入または変更されました。 **cryptokeydecryptrsa**、**cryptokeyencryptrsa**、**cryptokeylockrsa**、**cryptokeyunlockrsa**、**showcryptokeymypubkeyrsa**

第 4 章

PKI での証明書の許可および失効の設定

この章では、公開キーインフラストラクチャ（PKI）で証明書の許可および失効を設定する方法について説明します。証明書サーバへのハイ アベイラビリティのサポートに関する情報も挙げています。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイト ペーパーを参照してください。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

証明書の許可および失効に関する前提条件

PKI ストラテジの計画

ヒント　実際の証明書の展開を開始する前に、全体の PKI ストラテジを計画することを強く推奨します。

ユーザまたはネットワーク管理者が次の作業を完了した後に、許可および失効が発生します。

- 認証局（CA）の設定。
- ピア デバイスの CA への登録。
- ピアツーピア通信に使用される（IP セキュリティ（IPsec）またはセキュア ソケット レイヤ（SSL）などの）プロトコルの確認および設定。

許可および失効に固有の情報をピア デバイス証明書に含めなければならない場合があるため、ピア デバイスを登録する前に、設定する許可および失効ストラテジを決定する必要があります。

「crypto ca」から「crypto pki」への CLI の変更

Cisco IOS Release 12.3(7)T では、「crypto ca」で始まるすべてのコマンドが、「crypto pki」から始まるように変更されました。ルータは引き続き crypto ca コマンドを受信しますが、出力はすべて crypto pki に読み替えられます。

ハイ アベイラビリティ

ハイ アベイラビリティのため、IPsec 保護された Stream Control Transmission Protocol（SCTP）はアクティブ ルータとスタンバイ ルータの両方で設定する必要があります。同期を機能させるには、SCTP を設定した後に、証明書サーバの冗長性モードを ACTIVE/STANDBY に設定する必要があります。

証明書の許可および失効に関する制約事項

- シャーシ内での Stateful Switchover（SSO）冗長性の PKI High Availability（HA）サポートは、現在 Cisco IOS Release 12.2 S ソフトウェアを実行するすべてのスイッチ上でサポートされていません。詳細については、Cisco Bug CSCtb59872 を参照してください。
- Cisco IOS リリースに応じて、Lightweight Directory Access Protocol（LDAP）がサポートされます。

証明書の許可および失効に関する情報

PKI の許可

PKI 認証では、許可を行いません。多くの場合、一元的に管理されるソリューションが必要ですが、現在の許可用のソリューションは、設定対象のルータに固有です。

それによって証明書を特定の作業に対して許可し、その他の作業に対しては許可しない、と定義できる標準的なメカニズムはありません。アプリケーションが証明書ベースの許可情報を認識する場合、この許可情報を証明書自体に取り込めます。このソリューションでは、許可情報をリアルタイムで更新するための簡単なメカニズムを提供していないため、証明書に組み込まれた固有の許可情報を認識するように各アプリケーションに強制します。

証明書ベースのACLメカニズムがトラストポイント認証の一部として設定される場合、該当アプリケーションは、この許可情報を判別する役割を担うことはなく、どのアプリケーションに対して証明書を許可するのか指定できません。ルータ上の証明書ベースのACLは、大きくなりすぎて管理できないことがあります。また、外部サーバから証明書ベースのACL指示を取得する方が有利です（認証用に証明書ベースの ACL を使用する場合の詳細は、「許可または失効用に証明書ベースの ACL を使用する場合，（48 ページ）」を参照してください）。

許可の問題にリアルタイムで対処する現在のソリューションでは、新しいプロトコルの指定や新しいサーバの構築（それとともに管理およびデータ配布などの関連作業）が必要になります。

証明書ステータスのための PKI と AAA サーバの統合

PKIを認証、許可、アカウンティング（AAA）サーバと統合することにより、既存のAAAインフラストラクチャを活用する代替オンライン証明書ステータ スソリューションを実現します。証明書を適切な許可レベルで AAA データベースに一覧表示できます。PKI-AAA を明示的にサポートしないコンポーネントでは、デフォルトラベルの「all」を指定すると、AAA サーバからの許可が可能になります。また、AAA データベースのラベルが「none」の場合、指定された証明書が有効でないことを示します（アプリケーション ラベルが欠如していることと同じですが、「none」は完全性および明確性のために含まれます）。アプリケーション コンポーネントが PKI-AAA をサポートしている場合、コンポーネントを直接指定できる場合があります。たとえば、アプリケーション コンポーネントを「ipsec」、「ssl」、または「osp」に指定できます（ipsec = IP セキュリティ、ssl = セキュア ソケット レイヤ、および osp = Open Settlement Protocol）。

（注） 現在、アプリケーション ラベルの指定をサポートするアプリケーション コンポーネントはありません。

- AAA サーバにアクセスしたときに、時間遅延が生じる場合があります。AAA サーバを利用できない場合、許可は失敗します。

RADIUS または TACACS+：AAA サーバ プロトコルの選択

AAA サーバは、RADIUS または TACACS+ プロトコルと連動するように設定できます。PKI 統合用に AAA サーバを設定する場合、許可に必要な RADIUS または TACACS 属性を設定する必要があります。

RADIUS プロトコルが使われている場合は、AAA サーバのユーザ名に設定するパスワードを「cisco」に設定する必要があります。証明書の検証が認証を行い、AAA データベースは許可の目的だけに使用されているので、このパスワードは受け入れ可能です。TACACS プロトコルを使用する場合、TACACS では認証が不要な許可をサポートする（認証にパスワードを使用）ので、AAA サーバのユーザ名に対して設定されるパスワードとは無関係です。

さらに、TACACS を使用する場合は、AAA サーバに PKI サービスを追加する必要があります。カスタム属性「cert-application=all」が、PKI サービスの特定のユーザまたはユーザ グループに追加され、特定のユーザ名が許可されます。

PKI と AAA サーバ統合用の属性値ペア

次の表 に、AAA サーバと PKI との統合を設定する場合に使用される属性値（AV）ペアを示します（表に示す値は、可能な値であることに注意してください）。AV ペアはクライアント設定と一致する必要があります。AV ペアが一致しない場合、ピア証明書は許可されません。

（注） 場合によっては、ユーザは、他のすべてのユーザの AV ペアとは異なる AV ペアを持つことができます。その場合、ユーザごとに一意のユーザ名が必要になります。（**authorizationusername** コマンド内に）**すべての**パラメータを設定すると、証明書の所有者名全体を許可ユーザ名として使用するように指定できます。

表 *2*：一致する必要がある *AV* ペア

AV ペア	値
cisco-avpair=pki:cert-application=all	有効な値は、[all] および [none] です。
cisco-avpair=pki:cert-trustpoint=msca	この値は、Cisco IOS コマンドライン インターフェイス（CLI）設定のトラストポイント ラベルです。 （注） cert-trustpoint AV ペアの指定は、通常任意です。このペアが指定されている場合、Cisco IOS ルータ クエリーは、一致するラベルを持つ証明書トラストポイントから受信する必要があり、認証された証明書は、指定された証明書シリアル番号を持っている必要があります。

AV ペア	値
cisco-avpair=pki:cert-serial=16318DB7000100001671	この値は証明書のシリアル番号です。 （注） cert-serial AV ペアの指定は、通常任意です。このペアが指定されている場合、Cisco IOS ルータ クエリーは、一致するラベルを持つ証明書トラストポイントから受信する必要があり、認証された証明書は、指定された証明書シリアル番号を持っている必要があります。
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	cert-lifetime-end AV ペアは、証明書で指示された期間を越えた証明書のライフタイムを人為的に延長する場合に使用できます。cert-lifetime-end AV ペアを使用する場合は、cert-trustpoint および cert-serial AV ペアも指定する必要があります。この値は、時/分/月/日/年の形式と一致する必要があります。 （注） 月を表す最初の 3 文字（Jan、Feb、Mar、Apr、May、Jun、Jul、Aug、Sep、Oct、Nov、Dec）だけが使用されます。月を表す文字として 4 文字以上入力すると、残りの文字は無視されます（たとえば、Janxxxx）。

CRL または OCSP サーバ：証明書失効メカニズムの選択

証明書が適切に署名された証明書として有効になった後、証明書失効方法を実行して、証明書が発行元 CA によって無効にされていないことを確認します。Cisco IOS ソフトウェアは、2 つの失効メカニズムとして証明書失効リスト（CRL）と Online Certificate Status Protocol（OCSP）をサポートします。Cisco IOS ソフトウェアも、証明書のチェックために AAA 統合をサポートしますが、これには追加の許可機能が含まれます。PKI と AAA 証明書の許可とステータス確認に関する詳細については、「証明書ステータスのための PKI と AAA サーバの統合」を参照してください。

次の項では、各失効メカニズムの機能方法について説明します。

CRL とは

CRL とは、失効した証明書のリストです。CRL は、証明書を発行した CA によって作成され、デジタル署名されます。CRL には、各証明書の発行日と失効日が含まれています。

CA は、新しい CRL を定期的に、あるいは CA が責任を負う証明書が失効したときに公開します。デフォルトでは、現在キャッシュされている CRL が失効すると、新しい CRL がダウンロードさ

れます。管理者は、CRL がルータのメモリにキャッシュされる時間を設定したり、CRL キャッシングを完全にディセーブルにしたりできます。CRL キャッシング設定は、トラストポイントに関連付けられたすべての CRL に適用されます。

CRL が失効すると、ルータはキャッシュから CRL を削除します。証明書が検証用に表示されると、新しいCRLがダウンロードされます。ただし、検証中の証明書を記載した新しいバージョンの CRL がサーバ上にあるにもかかわらず、ルータがキャッシュ内の CRL を使用し続ける場合、ルータは証明書が失効したことを認識しません。証明書は拒否されるはずのものでも、失効チェックに合格します。

CA は、証明書を発行すると、証明書にその CRL 配布ポイント（CDP）を含めることができます。Cisco IOS クライアント デバイスは、CDP を使用して適切な CRL を見つけ、ロードします。Cisco IOS クライアントは複数の CDP をサポートしますが、Cisco IOS CA は現在 1 つの CDP しかサポートしません。ただし、サードパーティ ベンダー製の CA には、証明書ごとに複数の CDP または異なる CDP をサポートするものがあります。CDP が証明書に指定されていない場合、クライアント デバイスは、デフォルトの Simple Certificate Enrollment Protocol（SCEP）方式を使用して CRL を取得します（CDP の場所は、**cdp-url** コマンドを使用して指定できます）。

CRL を実装する際は、次の設計上の注意事項を考慮する必要があります。

- CRL ライフタイムとセキュリティ アソシエーション（SA）およびインターネット キー交換（IKE）ライフタイム
- CRL ライフタイムにより、CA が CRL の更新を発行する時間間隔が決まります（デフォルト CRL ライフタイム値は 168 時間（1 週間）です。これは、**lifetimecrl** コマンドで変更できます）。
- CDP のこの方式により、CRL の取得方法が決まり、この方式として、HTTP、Lightweight Directory Access Protocol（LDAP）、SCEP、または TFTP を選択できます。最も一般的に使用されている方式は、HTTP、TFTP、および LDAP です。Cisco IOS ソフトウェアでは、SCEP にデフォルト設定されていますが、CRL を使用して大容量のインストールを実行する場合、HTTP CDP を推奨します。HTTP では高いスケーラビリティを実現できるからです。
- CDP のこの場所は、CRL の取得先を決定します。たとえば、サーバおよび CRL の取得先となるファイル パスを指定できます。

失効チェック中にすべての CDP を照会

CDP サーバが要求に返答しない場合、Cisco IOS ソフトウェアはエラーを報告し、その結果、ピアの証明書が拒否されることがあります。証明書に複数の CDP がある場合、証明書が拒否されないようにするために、Cisco IOS ソフトウェアは、証明書に表示されている順序で CDP を使用しようと試みます。ルータは、それぞれの CDP URL またはディレクトリ指定を使用して CRL を取得しようと試みます。ある CDP を使用してエラーが発生すると、次の CDP を使用して試行します。

（注） Cisco IOS Release 12.3(7)T 以前のリリースでは、証明書に 2 つ以上の CDP が含まれていても、Cisco IOS ソフトウェアは、CRL の取得を 1 回だけ試行します。

ヒント　Cisco IOS ソフトウェアは、指示された CDP のいずれかから CRL を取得するためにあらゆる試行を行いますが、CDP応答の遅延によりアプリケーションのタイムアウトを避けるために、HTTP CDP サーバを高速の冗長 HTTP サーバと併用することを推奨します。

OCSP とは

OCSP は、証明書の有効性を判別するために使用されるオンラインのメカニズムであり、失効メカニズムとして次のような柔軟性を備えています。

- OCSP では、証明書ステータスをリアルタイムでチェックできます。
- OCSP を使用すると、ネットワーク管理者は、中央 OCSP サーバを指定でき、これにより、ネットワーク内のすべてのデバイスにサービスを提供できます。
- また、OCSP により、ネットワーク管理者は、クライアント証明書ごと、またはクライアント証明書のグループごとに複数の OCSP サーバを柔軟に指定できます。
- OCSP サーバの検証は通常、ルート CA 証明書または有効な下位 CA 証明書に基づいて実行されますが、外部の CA 証明書または自己署名証明書を使用できるように設定することもできます。外部の CA 証明書または自己署名証明書を使用すると、代替の PKI 階層から OCSP サーバ証明書を発行し、有効にできます。

ネットワーク管理者は、さまざまな CA サーバから CRL を収集し、更新するように OCSP サーバを設定できます。ネットワーク内のデバイスは、OCSP サーバに依存して、ピアごとに CRL を取得してキャッシュすることなく証明書ステータスをチェックできます。ピアは、証明書の失効ステータスをチェックする必要がある場合、OCSP 要求に関して疑わしい証明書のシリアル番号およびオプションの固有識別情報（ナンス）を含む OCSP サーバにクエリーを送信します。OCSP サーバは、CRL のコピーを保持して、CA がその証明書を無効として記載しているかどうか判別します。次に、サーバは、ナンスを含むピアに応答します。応答のナンスが OCSP サーバからピアによって送信された元のナンスと一致しない場合、応答は無効と見なされ、証明書の検証が失敗します。OCSP サーバとピア間の対話での帯域幅の消費量は、ほとんどの場合、CRL ダウンロードより少なくなります。

OCSP サーバが CRL を使用する場合は、CRL 時間の制約事項が適用されます。つまり、追加の証明書失効情報を含む CRL によって新しい CRL が発行されていても、まだ有効な CRL が OCSP サーバで使用されることがあります。CRL 情報を定期的にダウンロードするデバイスが少なくなっているため、CRL ライフタイム値を小さくするか、CRL をキャッシュしないように OCSP サーバを設定できます。詳細は、OCSP サーバのマニュアルを参照してください。

OCSP サーバを使用する場合

PKI に次のいずれかの特性がある場合、CRL よりも OCSP の方が適している場合があります。

- リアルタイムの証明書失効ステータスが必要。CRL が定期的にしか更新されず、必ずしも最新の CRL がクライアント デバイスでキャッシュされていない場合があります。たとえば、

最新のCRLがまだクライアントにキャッシュされておらず、また、新たに無効にされた証明書がチェック中の場合は、無効にされた証明書が失効チェックに合格します。

- 無効にされた大量の証明書または複数のCRLがあります。大きなCRLをキャッシュすると、Cisco IOSメモリの大部分が消費されてしまい、他のプロセスに使用できるリソースが減少することがあります。
- CRL が頻繁に失効するため、CDP は大量の CRL を処理します。

（注） Cisco IOS Release 12.4(9)T 以降では、管理者は、CRL キャッシングを完全にディセーブルにするか、キャッシュされた CRL のトラストポイントごとに最大ライフタイムを設定することによって、CRL キャッシングを設定できます。

許可または失効用に証明書ベースの ACL を使用する場合

証明書には、指定された処理の実行をデバイスまたはユーザが許可されているかどうかの判別に使用されるフィールドがいくつか含まれています。

証明書ベースACLはデバイス上に設定されるため、大量のACLを十分にスケーリングしません。ただし、証明書ベースのACLでは、特定のデバイスの動作を非常に細かく制御できます。また、証明書ベースACLは追加機能で活用され、失効、許可、またはトラストポイントなどのPKIコンポーネントを使用するタイミングを判別するのを助けます。証明書ベースACLは全般的なメカニズムを提供しており、このメカニズムによりユーザは、許可または追加処理に対して有効になっている特定の証明書または証明書のグループを選択できます。

証明書ベース ACL では、証明書内の 1 つ以上のフィールドおよび指定された各フィールドで許可される値を指定します。証明書内でチェックする必要があるフィールドと、それらのフィールドで認められる値または認められない値を指定できます。

フィールドと値との比較には、6 つの論理テスト（Equal（等しい）、Not equal（等しくない）、Contains（含む）、Less than（未満）、Does not contain（含まない）、Greater than or equal（以上））を使用できます。1 つの証明書ベース ACL で複数のフィールドを指定した場合、その ACL と一致するには、ACL 内のすべてのフィールド条件に合致しなければなりません。同じ ACL 内で、同じフィールドを複数回指定できます。複数の ACL を指定できます。一致するものが見つかるか、または ACL の処理がすべて完了するまで、各 ACL が順に処理されます。

証明書ベース ACL を使用した失効チェックの無視

証明書ベースACLを設定して、有効なピアの失効チェックおよび失効した証明書を無視するようルータに指示できます。したがって、指定基準を満たす証明書は、証明書の有効期間にかかわらず受け入れることができます。また、証明書が指定基準を満たしている場合は失効チェックを実行する必要がなくなります。AAAサーバとの通信が証明書で保護される場合にも、証明書ベースACL を使用して失効チェックを無視できます。

失効リストの無視

トラストポイントが特定の証明書を除いてCRLを適用できるようにするには、**skip revocation-check** キーワードを指定して **match certificate** コマンドを入力します。このような適用は、スポークツースポークの直接接続も可能なハブアンドスポーク設定に最も便利です。純粋なハブアンドスポーク設定では、すべてのスポークはハブだけに接続するので、CRL チェックはハブ上だけで済みます。スポークが別のスポークと直接通信する場合、ネイバー ピア証明書に対して、各スポーク上でCRL を要求する代わりに、**skip revocation-check** キーワードを指定して **match certificate** コマンドを使用できます。

失効した証明書の無視

失効した証明書を無視するようにルータを設定するには、**allow expired-certificate** キーワードを指定して **match certificate** コマンドを入力します。このコマンドには、次のような目的があります。

- このコマンドは、ピアの証明書が失効した場合にピアが新しい証明書を取得するまで、失効した証明書を「許可する」ために使用できます。
- ルータ クロックがまだ正しい時間に設定されていない場合、クロックが設定されるまで、ピアの証明書はまだ有効ではないものとして表示されます。このコマンドは、ルータ クロックが未設定であっても、ピアの証明書を許可する場合に使用できます。

（注） ネットワーク タイム プロトコル（NTP）が IPSec 接続だけで（通常、ハブアンドスポーク設定のハブによって）利用可能な場合は、ルータ クロックを絶対に設定できません。ハブの証明書がまだ有効でないため、ハブへのトンネルを「アップ」状態にできません。

- 「失効」とは、失効している証明書またはまだ有効ではない証明書の総称です。証明書には、開始時刻と終了時刻が指定されます。ACL を目的とした、失効証明書は、ルータの現在時刻が証明書で指定された開始および終了時刻の範囲外の証明書です。

証明書の AAA チェックのスキップ

AAA サーバとの通信が証明書で保護され、証明書の AAA チェックをスキップする場合は、**skip authorization-check** キーワードを指定して **match certificate** コマンドを使用します。たとえば、すべての AAA トラフィックがバーチャル プライベート ネットワーク（VPN）トンネルを通過するように設定され、このトンネルが証明書で保護されている場合は、**skip authorization-check** キーワードを指定して **match certificate** コマンドを使用すると、証明書チェックをスキップしてトンネルを確立できます。

AAA サーバとの PKI 統合が設定されると、**match certificate** コマンドと **skip authorization-check** キーワードを設定する必要があります。

（注） AAA サーバが IPSec 接続によってのみ使用可能な場合は、IPSec 接続が確立されるまで AAA サーバとは通信できません。AAA サーバの証明書がまだ有効でないため、IPSec 接続を「アップ」状態にできません。

PKI 証明書チェーンの検証

証明書チェーンにより、ピア証明書からルート CA 証明書までの、一連の信頼できる証明書を確立します。階層型 PKI 内では、登録されているすべてのピアが信頼できるルート CA 証明書または共通の下位 CA を共有している場合、証明書を相互に検証できます。各 CA が 1 つのトラストポイントに対応します。

証明書チェーンをピアから受信すると、最初の信頼できる証明書またはトラストポイントに到達するまで、証明書チェーン パスのデフォルト処理が続けられます。Cisco IOS Release 12.4(6)T 以降のリリースでは、管理者は、下位 CA 証明書を含むすべての証明書における証明書チェーンの処理レベルを設定できます。

証明書チェーンの処理レベルを設定すると、信頼できる証明書の再認証、信頼できる証明書チェーンの延長、および欠落のある証明書チェーンの補完が可能になります。

信頼できる証明書の再認証

このデフォルト動作でルータは、チェーンを検証する前に、ピアによって送信された証明書チェーンから任意の信頼できる証明書を削除します。管理者は証明書チェーン パス処理を設定して、チェーン検証の前にすでに信頼されている CA 証明書をルータが削除しないようにできます。そのため、チェーン内のすべての証明書は現在のセッションに対して再度認証されます。

信頼できる証明書チェーンの延長

このデフォルト動作でルータは、ピアによって送信された証明書チェーンに欠落している証明書がある場合、その信頼できる証明書を使用して証明書チェーンを延長します。ルータが検証するのは、ピアによって送信されたチェーンの証明書だけです。管理者は証明書チェーン パス処理を設定して、ピアの証明書チェーンの証明書およびルータの信頼できる証明書を、指定したポイントに対して有効にできます。

証明書チェーンの欠落の補完

管理者は証明書チェーン処理を設定して、設定済みの Cisco IOS トラストポイント階層に欠落がある場合、ピアによって送信された証明書を使用して証明書のセットを有効にできます。

（注） 親検証を要求するようにトラストポイントが設定され、ピアが完全な証明書チェーンを提示しない場合、欠落を補完できないため証明書チェーンは拒否され、無効になります。

（注） 親検証を要求するようにトラストポイントが設定されていて、設定済みの親トラストポイントがない場合は、設定エラーです。発生する証明書チェーンの欠落を補完できず、下位 CA 証明書を有効にできません。この証明書チェーンは無効です。

ハイ アベイラビリティのサポート

証明書サーバへのハイ アベイラビリティのサポートは、次の方法で実現します。

- 取り消しコマンドのスタンバイ証明書サーバとの同期
- 証明書の新規発行時のシリアル番号コマンドの送信

スタンバイ証明書サーバがアクティブになると、証明書と CRL を発行する手段の準備が完了します。

ハイ アベイラビリティのサポートをさらに高めるには、スタンバイとの次の同期を行います。

- 証明書サーバ設定
- 保留中の要求
- コマンドの許可と拒否
- 設定の同期がサポートされないボックスツーボックスのハイ アベイラビリティのためには、基本設定の同期メカニズムが冗長性機能上で動作します。
- トラストポイント設定同期のサポート

PKI に対して証明書の許可および失効を設定する方法

AAA サーバとの PKI 統合の設定

ピアによって提出された証明書から AAA ユーザ名を生成し、証明書内で AAA データベース ユーザ名の作成に使用するフィールドを指定するには、次の作業を実行します。

（注） **authorizationusername** コマンドで所有者名として **all** キーワードを使用する際に、次の制約事項を考慮する必要があります。

- 一部の AAA サーバでは、ユーザ名の長さが制限されます（たとえば、64 文字まで）。その結果、証明書の全体の所有者名は、サーバの制約条件より長くできません。
- 一部の AAA サーバでは、ユーザ名に使用できる文字セットが制限されます（たとえば、スペース（）および等号（=）を使用できない場合があります）。このような文字セットの制限がある AAA サーバでは、**all** キーワードを使用できません。
- トラストポイント設定の **subject-name** コマンドは、必ずしも最終の AAA 所有者名とはかぎりません。証明書要求に完全修飾ドメイン名（FQDN）、シリアル番号、またはルータの IP アドレスが含まれている場合は、発行された証明書の所有者名フィールドにもこれらのコンポーネントが含まれます。コンポーネントをオフにするには、**fqdn**、**serial-number**、および **ip-address** の各コマンドに **none** キーワードを使用します。
- CA サーバが証明書を発行すると、CA サーバは、要求した所有者名フィールドを変更することがあります。たとえば、一部のベンダーの CA サーバが要求した所有者名の相対識別名（RDN）を CN、OU、O、L、ST、および C に切り替えます。ただし、別の CA サーバは、設定した LDAP ディレクトリ ルート（O=cisco.com など）を要求した所有者名の最後に追加する場合があります。
- 証明書の表示用に選択するツールによっては、所有者名の RDN の印刷順序が異なることがあります。Cisco IOS ソフトウェアでは、重要度が最低の RDN を先頭に表示しますが、Open Source Secure Socket Layer（OpenSSL）などの、他のソフトウェアでは、重要度が最高の RDN を先頭に表示します。したがって、完全な識別名（DN）（所有者名）を持つ AAA サーバを対応するユーザ名として設定する場合は、Cisco IOS ソフトウェア スタイル（つまり、重要度が最低の RDN を先頭に表示）が使用されていることを確認してください。

または

radius-serverhost*hostname* [**key***string*]

手順の概要

1. **enable**
2. **configureterminal**
3. **aaanew-model**
4. **aaaauthorizationnetwork***listname* [*method*]
5. **cryptopkitrustpoint***name*
6. **enrollment** [**mode**] [**retry period***minutes*] [**retry count***number*] **url***url* [**pem**]
7. revocation-check method
8. **exit**
9. **authorizationusernamesubjectname***subjectname*
10. **authorizationlist***listname*
11. **tacacs-serverhost** hostname [**key** string]

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**aaanew-model** 例： `Router(config)# aaa new-model`	AAA アクセス コントロール モデルをイネーブルにします。
ステップ 4	**aaaauthorizationnetwork***listname* [*method*] 例： `Router (config)# aaa authorization network maxaaa group tacacs+`	ネットワークへのユーザ アクセスを制限するパラメータを設定します。 • *method*：**groupradius**、**grouptacacs+**、または **groupgroup-name** を指定できます。
ステップ 5	**cryptopkitrustpoint***name* 例： `Route (config)# crypto pki trustpoint msca`	トラストポイントおよび設定された名前を宣言して、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。

	コマンドまたはアクション	目的

ステップ 6	**enrollment** [**mode**] [**retry period***minutes*] [**retry count***number*] **url***url* [**pem**]
	
	例：
	
	`Router (ca-trustpoint)# enrollment url http://caserver.myexample.com`
	または
	
	`Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80`
		CA の次の登録パラメータを指定します。
		• （任意）CA システムが登録局（RA）を提供する場合、**mode** キーワードとして登録局（RA）モードを指定します。デフォルトでは、RA モードはディセーブルです。
		• （任意）**retry period** キーワードおよび *minutes* 引数は、CA に別の証明書要求を送信するまでルータが待機する期間を分単位で指定します。有効値は 1 ～ 60 です。デフォルトは 1 です。
		• （任意）**retry count** キーワードおよび *number* 引数は、直前の要求に対する応答をルータが受信しない場合、ルータが証明書要求を再送信する回数を指定します。有効な値は、1 ～ 100 です。デフォルトは 10 です。
		• *url* 引数は、ルータが証明書要求を送信する CA の URL です。
		（注） Cisco IOS Release 15.2(1)T を導入すると、IPv6 アドレスを **http:** 登録方式に追加できます。たとえば、http://[ipv6-address]:80 です。URL 内の IPv6 アドレスは括弧で囲む必要があります。使用できるその他の登録方式に関する詳細については、enrollment url (ca-trustpoint) コマンド ページを参照してください。
		• （任意）**pem** キーワードは、証明書要求に Privacy Enhanced Mail（PEM）の境界を追加します。

ステップ 7	revocation-check method 例： `Router (ca-trustpoint)# revocation-check crl`	（任意）証明書の失効ステータスをチェックします。
ステップ 8	**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、グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 9	**authorizationusernamesubjectname***subjectname* 例： `Router (config)# authorization username subjectname serialnumber`	AAA ユーザ名の構築に使用する異なる証明書フィールドのパラメータを設定します。 *subjectname* 引数には、次のいずれかを指定できます。 • **all**：証明書の識別名（所有者名）全体 • **commonname**：証明書の通常名 • **country**：証明書の国 • **email**：証明書の E メール • **ipaddress**：証明書の IP アドレス • **locality**：証明書の地域 • **organization**：証明書の組織 • **organizationalunit**：証明書の組織単位 • **postalcode**：証明書の郵便番号 • **serialnumber**：証明書のシリアル番号 • **state**：証明書の州フィールド • **streetaddress**：証明書の所在地 • **title**：証明書のタイトル • **unstructuredname**：証明書の非公式名
ステップ 10	**authorizationlist***listname* 例： `Route (config)# authorization list maxaaa`	AAA 認可リストを指定します。

ステップ 11	**tacacs-serverhost** hostname [**key** string]
	
	例：
	
	`Router(config)# tacacs-server host 192.0.2.2 key a_secret_key`
	
	例：
	
	`radius-server host hostname [key string]`
		TACACS+ ホストを指定します。
		
		または
		
		RADIUS ホストを指定します。

	コマンドまたはアクション	目的
	例： `Router(config)# radius-server host 192.0.2.1 key another_secret_key`	

トラブルシューティングのヒント

CA とルータ間のインタラクションのトレース（メッセージ タイプ）に関するデバッグ メッセージを表示するには、**debug crypto pki transactions** コマンドを使用します（サンプル出力を参照してください。ここでは、AAA サーバ交換との成功した PKI 統合、および AAA サーバ交換との失敗した PKI 統合を示します）。

成功した交換

```
Router# debug crypto pki transactions
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
```

「CRYPTO_PKI_AAA」と表示されている各行は、AAA認可チェックの状態を示します。各AAA AV ペアが示され、認可チェックの結果が表示されます。

```
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aaalist,
PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

失敗した交換

```
Router# debug crypto pki transactions
Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = “all”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"= “CA1”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = “233D”)
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

上記の失敗した交換では、証明書が失効しています。

PKI 証明書ステータス チェックの失効メカニズムの設定

証明書失効メカニズム（CRL または OCSP）として CRL を設定し、PKI の証明書のステータスをチェックするには、次の作業を実行します。

revocation-check コマンド

revocation-check コマンドを使用し、ピアの証明書が無効にされていないことを確認するための方式（OCSP、CRL、または失効チェックのスキップ）を少なくとも 1 つ指定します。複数の方式を指定する場合、方式を適用する順序は、このコマンドで指定した順序になります。

ルータに適用可能な CRL がなく、いずれの CRL も取得できない場合、あるいは OCSP サーバがエラーを返す場合、設定に **none** キーワードを含めないかぎり、ルータはピアの証明書を拒否します。**none** キーワードを設定した場合、失効チェックは実行されず、証明書は常に受け入れられます。

OCSP サーバとのナンスおよびピア通信

OCSP を使用すると、OCSP サーバとのピア通信時に、OCSP 要求に関するナンス（固有識別情報）がデフォルトで送信されます。ナンスを使用することにより、ピアと OCSP サーバ間にセキュアで信頼性の高い通信チャネルが確立されます。

OCSP サーバがナンスをサポートしていない場合は、ナンスの送信をディセーブルにできます。詳細は、OCSP サーバのマニュアルを参照してください。

はじめる前に

- クライアント証明書を発行する前に、サーバで適切な設定（CDP の設定など）を行う必要があります。
- OCSP サーバから CA サーバの失効ステータスを返すように設定するときは、CA サーバが発行した OCSP 応答署名証明書を OCSP サーバに設定する必要があります。署名証明書が正しいフォーマットであることを確認してください。署名証明書のフォーマットが正しくない場合、ルータは、OCSP 応答を受理しません。詳細については、OCSP のマニュアルを参照してください。

（注）

- OCSP は、HTTP を使用してメッセージを転送するので、OCSP サーバにアクセスする際に遅延が発生する場合があります。
- OCSP サーバが、失効ステータスのチェックを通常の CRL 処理に依存している場合、CRL の遅延は OCSP にも適用されます。

>

手順の概要

1. **enable**
2. **configureterminal**
3. crypto pki trustpoint *name*
4. **ocspurl***url*
5. **revocation-check***method1* [*method2method3*]]
6. **ocspdisable-nonce**
7. **exit**
8. **exit**
9. **showcryptopkicertificates**
10. **showcryptopkitrustpoints** [**status** | *label* [**status**]]

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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 hazel`	トラストポイントおよび設定された名前を宣言して、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	**ocspurl***url*
	
	例：
	
	`Router(ca-trustpoint)# ocsp url http://ocsp-server`
	または
	
	`Router(ca-trustpoint)# ocsp url http://10.10.10.1:80`
	または
	
	`Router(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80`
		url 引数は、トラストポイントが証明書ステータスをチェックできるように OCSP サーバの URL を指定します。この URL は、証明書の AIA 拡張部に指定されている OCSP サーバの URL（存在する場合）を上書きします。設定したトラストポイントに関連するすべての証明書は、OCSP サーバによって確認されます。使用可能な URL は、ホスト名、IPv4 アドレス、または IPv6 アドレスです。

	コマンドまたはアクション	目的
ステップ 5	**revocation-check***method1* [*method2method3*]] 例： `Router(ca-trustpoint)# revocation-check ocsp none`	証明書の失効ステータスをチェックします。 • **crl**：CRL によって証明書をチェックします。これがデフォルトのオプションです。 • **none**：証明書のチェックを無視します。 • **ocsp**：OCSP サーバによって証明書をチェックします。 2 番目と 3 番目の方法を指定した場合、各方法はその直前の方法でエラーが返された場合（サーバがダウンしている場合など）にだけ使用されます。
ステップ 6	**ocspdisable-nonce** 例： `Router(ca-trustpoint)# ocsp disable-nonce`	（任意）OCSP サーバとピアが通信するときに、ナンス（OCSP 要求に関する固有識別情報）が送信されないように指定します。
ステップ 7	**exit** 例： `Router(ca-trustpoint)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**exit** 例： `Router(config)# exit`	特権 EXEC モードに戻ります。
ステップ 9	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）証明書に関する情報を表示します。
ステップ 10	**showcryptopkitrustpoints** [**status** | *label* [**status**]] 例： `Router# show crypto pki trustpoints`	ルータに設定されているトラストポイントに関する情報を表示します。

証明書の許可および失効の設定

証明書ベースACLの指定、失効チェックまたは失効した証明書の無視、手動によるデフォルトのCDPの場所の上書き、手動によるOCSPサーバ設定の上書き、CRLキャッシングの設定、あるいは証明書シリアル番号に基づくセッションの受理/拒否の設定を行うには、必要に応じて次の作業を実行します。

失効チェックを無視するように証明書ベース ACL を設定

証明書ベースACLを使用して、失効チェックおよび失効証明書を無視するようにルータを設定するには、次の手順を実行します。

- 既存のトラストポイントの識別またはピアの証明書の検証に使用される新しいトラストポイントを作成します。トラストポイントがまだ認証されていない場合は、認証してください。必要に応じて、ルータをこのトラストポイントに登録できます。**match certificate** コマンドと **skip revocation-check** キーワードを使用する場合は、トラストポイントにオプションの CRL を設定しないでください。
- 証明書自体のCRLをチェックする必要がない証明書の固有の特性と、許可する必要がある失効証明書の固有の特性を判別します。
- 前のステップで確認した特性と一致する証明書マップを定義します。
- 最初の手順で作成または識別したトラストポイントに、**match certificate** コマンドと **skip revocation-check** キーワード、**match certificate コマンド**と **allow expired-certificate** キーワードを追加できます。

（注） 証明書マップは、ピアの公開キーがキャッシュされている場合でも確認されます。たとえば、ピアによって公開キーがキャッシュされており、証明書マップがトラストポイントに追加されて証明書が禁止されると、証明書マップが有効になります。これにより、過去に一度接続され、現在は禁止されている証明書を持つクライアントが再接続することを防ぎます。

証明書内の CDP の手動による上書き

ユーザは、手動で設定したCDPで証明書内のCDPを上書きできます。証明書のCDPの手動による上書きは、特定のサーバが長時間利用できない場合に便利です。元のCDPを含む証明書のすべてを再発行しなくても、証明書のCDPをURLまたはディレクトリ指定に置き換えることができます。

手動による証明書の OCSP サーバ設定の上書き

管理者は、**ocsp url** コマンドを発行して、クライアント証明書のAuthority Information Access（AIA）フィールドに指定されているOCSPサーバの設定値を上書きまたは設定できます。**match certificate**

override ocsp コマンドを使用すると、複数の OCSP サーバをクライアント証明書ごとに、またはクライアント証明書のグループごとに手動で指定できます。失効チェック時にクライアント証明書が証明書マップに正常に照合された場合、**match certificate override ocsp** コマンドを発行すると、クライアント証明書 AIA フィールドまたは **ocsp url** コマンド設定が上書きされます。

（注） 1 つのクライアント証明書には、OCSP サーバを 1 つだけ指定できます。

CRL キャッシュ コントロールの設定

デフォルトでは、現在キャッシュされている CRL が失効すると、新しい CRL がダウンロードされます。管理者は、**crl cache delete-after** コマンドを発行して、CRL がキャッシュに保持される最大時間（分単位）を設定するか、**crl cache none** コマンドを発行して CRL キャッシュをディセーブルにできます。指定できるのは、**crl-cache delete-after** コマンドまたは **crl-cache none** コマンドだけです。トラストポイントに両方のコマンドを入力した場合は、後に実行されたコマンドが有効になり、メッセージが表示されます。

crl-cache none コマンドまたは **crl-cache delete-after** コマンドのいずれを実行しても現在キャッシュされている CRL に影響はありません。**crl-cache none** コマンドを設定した場合、このコマンドを発行すると、ダウンロードされたすべての CRL はキャッシュされません。**crl-cache delete-after** コマンドを設定した場合、このコマンドの発行後に設定されたライフタイムだけがダウンロードされた CRL に影響します。

この機能は、CA が失効日を指定せずに CRL を発行する場合、あるいは失効日が数日後または数週間後に迫っている場合に役立ちます。

証明書のシリアル番号セッション コントロールの設定

証明書検証要求がセッションのトラストポイントによって受け入れられる、または拒否されるように証明書シリアル番号を指定できます。証明書のシリアル番号セッション コントロールによっては、証明書がまだ有効であっても、セッションが拒否される場合があります。証明書のシリアル番号セッション コントロールは、**serial-number** フィールドを持つ証明書マップまたは **cert-serial-not** コマンドを使用する AAA 属性のいずれかを使用して設定できます。

セッション コントロールに証明書マップを使用すると、管理者は、1 つの証明書シリアル番号を指定できます。AAA 属性を使用すると、管理者は、セッション コントロールに証明書シリアル番号を指定できます。

はじめる前に

- 証明書マップをトラストポイントに関連付ける前に、トラストポイントを定義し、認証する必要があります。
- CDP オーバライド機能をイネーブルにする、または **serial-number** コマンドを発行する前に、証明書マップを設定する必要があります。

- PKI と AAA サーバとの統合は、「証明書ステータスのための PKI と AAA サーバの統合」の説明のとおりに AAA 属性を使用して正常に完了する必要があります。

手順の概要

1. **enable**
2. **configureterminal**
3. crypto pki certificate map label sequence-number
4. *field-namematch-criteriamatch-value*
5. **exit**
6. **cryptopkitrustpoint***name*
7. 次のいずれかを実行します。
   - **crl-cachenone**
   - **crl-cachedelete-after***time*
8. **matchcertificate***certificate-map-label* [**allowexpired-certificate** | **skiprevocation-check** | **skipauthorization-check**
9. **matchcertificate***certificate-map-label***overridecdp** {**url** | **directory**} *string*
10. **matchcertificate***certificate-map-label***overrideocsp**[**trustpoint***trustpoint-label*] *sequence-number***url***ocsp-url*
11. **exit**
12. **aaanew-model**
13. **aaaattributelist***list-name*
14. **attributetype** {*name*}{*value*}
15. **exit**
16. **exit**
17. **showcryptopkicertificates**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 3	crypto pki certificate map label sequence-number 例： `Router(config)# crypto pki certificate map Group 10`	証明書において、一致する必要がある値または一致する必要がない値を定義し、CA 証明書マップ 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	*field-namematch-criteriamatch-value* 例： `Router(ca-certificate-map)# subject-name co MyExample`	1 つまたは複数の証明書フィールドと、これらのフィールドの一致基準および照合する値を指定します。 *field-name* には、次のいずれかの名前文字列（大文字と小文字を区別しない）または日付を指定します。 • **alt-subject-name** • **expires-on** • **issuer-name** • **name** • **serial-number** • **subject-name** • **unstructured-subject-name** • **valid-start** （注） 日付フィールドのフォーマットは、dd mm yyyy hh:mm:ss または mmm dd yyyy hh:mm:ss です。 *match-criteria* には、次の論理演算子のいずれかを指定します。 • **co**：含む（名前およびシリアル番号フィールドでのみ有効） • **eq**：等しい（名前、シリアル番号、および日付フィールドで有効） • **ge**：以上（日付フィールドでのみ有効） • **lt**：未満（日付フィールドでのみ有効） • **nc**：含まない（名前およびシリアル番号フィールドでのみ有効）

	コマンドまたはアクション	目的
		• **ne**：等しくない（名前、シリアル番号、および日付フィールドで有効） *match-value* は、match-criteria で割り当てられた論理演算子を使用してテストする名前または日付です。 （注） このコマンドは、証明書ベース ACL を設定する場合にだけ使用し、失効チェックまたは失効した証明書を無視するように証明書ベース ACL を設定する場合には使用しないでください。
ステップ 5	**exit** 例： `Router(ca-certificate-map)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**cryptopkitrustpoint***name* 例： `Router(config)# crypto pki trustpoint Access2`	トラストポイントおよび設定された名前を宣言して、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 7	次のいずれかを実行します。
	
	• **crl-cachenone**
	
	• **crl-cachedelete-after***time*
	
	例：
	
	`Router(ca-trustpoint)# crl-cache none`
	
	例：
	
	`Router(ca-trustpoint)# crl-cache delete-after 20`
		（任意）トラストポイントに関連付けられたすべての CRL の CRL キャッシングを完全にディセーブルにします。
		
		crl-cachenone コマンドを実行しても、現在キャッシュされている CRL に影響はありません。このコマンドが設定された後にダウンロードされるすべての CRL は、キャッシュされません。
		
		（任意）トラストポイントに関連付けられたすべての CRL に関して、CRL がキャッシュに保持される最大時間を指定します。
		
		• *time*：CRL が削除されるまでの時間（分単位）。
		
		crl-cachedelete-after コマンドを実行しても、現在キャッシュされている CRL

	コマンドまたはアクション	目的
		に影響はありません。設定されたライフタイムは、このコマンドが設定された後にダウンロードされた CRL だけに影響します。
ステップ 8	**matchcertificate***certificate-map-label* [**allowexpired-certificate** | **skiprevocation-check** | **skipauthorization-check** 例： `Router(ca-trustpoint)# match certificate Group skip revocation-check`	（任意）証明書ベース ACL（**cryptopkicertificatemap** コマンドによって定義されている）をトラストポイントに関連付けます。 • *certificate-map-label*：**cryptopkicertificatemap** コマンドを使用して指定した *label* 引数と一致する必要があります。 • **allowexpired-certificate**：失効した証明書を無視します。 • **skiprevocation-check**：トラストポイントが、特定の証明書を除く CRL を適用できるようにします。 • **skipauthorization-check**：AAA サーバとの PKI 統合を設定すると、証明書の AAA チェックをスキップします。
ステップ 9	**matchcertificate***certificate-map-label***overridecdp** {**url** | **directory**} *string* 例： `Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com`	（任意）URL またはディレクトリが指定された証明書の、既存の CDP エントリを手動で上書きします。 • *certificate-map-label*：ユーザ指定のラベル。事前に定義された **cryptopkicertificatemap** コマンドに指定した *label* 引数と一致する必要があります。 • **url**：証明書の CDP が HTTP または LDAP URL で上書きされるように指定します。 • **directory**：証明書の CDP が LDAP ディレクトリ指定で上書きされるように指定します。 • *string*：URL またはディレクトリ指定。

	コマンドまたはアクション	目的
		（注） 一部のアプリケーションは、すべての CDP が試行される前にタイムアウトすることがあり、エラー メッセージで報告します。エラー メッセージはルータに影響を及ぼしません。また、Cisco IOS ソフトウェアは、すべての CDP が試行されるまで CRL の取得を続行します。
ステップ 10	**matchcertificate***certificate-map-label***overrideocsp**[**trustpoint***trustpoint-label*] *sequence-number***url***ocsp-url* 例： `Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url http://192.0.2.2`	（任意）OCSP サーバをクライアント証明書ごとに、またはクライアント証明書のグループごとに指定し、複数回発行して、追加の OCSP サーバおよびクライアント証明書の設定（代替の PKI 階層を含む）を指定できます。 • *certificate-map-label*：既存の証明書マップ名。 • **trustpoint**：OCSP サーバ証明書を検証するときに使用されるトラストポイント。 • *sequence-number*：**matchcertificateoverrideocsp** コマンド文を検証対象の証明書に適用する順序。照合が最低のシーケンス番号から最高のシーケンス番号に実行されます。同じシーケンス番号で複数のコマンドを発行すると、前の OCSP サーバ オーバライド設定が上書きされます。 • **url**：OCSP サーバの URL。 証明書が設定された証明書マップと一致すると、クライアント証明書の AIA フィールドおよび以前に発行された **ocspurl** コマンド設定値は、指定された OCSP サーバで上書きされます。 マップベースの一致が発生しない場合、引き続き次の 2 つのケースがクライアント証明書に適用されます。

	コマンドまたはアクション	目的
		• OCSP を失効方法として指定すると、AIA フィールド値がクライアント証明書に引き続き適用されます。 • **ocspurl** 設定が存在する場合は、**ocspurl** 設定が引き続きクライアント証明書に適用されます。
ステップ 11	**exit** 例： `Router(ca-trustpoint)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 12	**aaanew-model** 例： `Router(config)# aaa new-model`	（任意）AAA アクセス コントロール モデルをイネーブルにします。
ステップ 13	**aaaattributelist***list-name* 例： `Router(config)# aaa attribute list crl`	（任意）ルータにローカルで AAA 属性リストを定義し、config-attr-list コンフィギュレーション モードを開始します。
ステップ 14	**attributetype** {*name*}{*value*} 例： `Router(config-attr-list)# attribute type cert-serial-not 6C4A`	（任意）ルータの AAA 属性リストにローカルに追加される AAA 属性タイプを定義します。 証明書のシリアル番号セッション コントロールを設定するために、管理者は、*value* フィールドの特定の証明書を、*name* が **cert-serial-not** に設定されているシリアル番号に基づき受け入れるか、拒否するか指定できます。証明書のシリアル番号が属性タイプ設定で指定されたシリアル番号と一致した場合、証明書は拒否されます。 使用可能な AAA 属性タイプのリストを表示するには、**showaaaattributes** コマンドを実行してください。

	コマンドまたはアクション	目的

ステップ 15	**exit**
	
	例：
	`Router(ca-trustpoint)# exit`
	
	例：
	`Router(config-attr-list)# exit`
		グローバル コンフィギュレーション モードに戻ります。

ステップ 16	**exit** 例： `Router(config)# exit`	特権 EXEC モードに戻ります。
ステップ 17	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）CA 証明書が認証されたら、ルータにインストールされた証明書のコンポーネントを表示します。

例

次に、サンプル証明書を示します。OCSP 関連の拡張子は感嘆符を使用して示されます。

```
Certificate:
        Data:
            Version: v3
            Serial Number:0x14
            Signature Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Issuer:CN=CA server,OU=PKI,O=Cisco Systems
            Validity:
                Not Before:Thursday, August 8, 2002 4:38:05 PM PST
                Not After:Tuesday, August 7, 2003 4:38:05 PM PST
            Subject:CN=OCSP server,OU=PKI,O=Cisco Systems
            Subject Public Key Info:
                Algorithm:RSA - 1.2.840.113549.1.1.1
                Public Key:
                    Exponent:65537
                    Public Key Modulus:(2048 bits) :
                        <snip>
            Extensions:
                Identifier:Subject Key Identifier - 2.5.29.14
                    Critical:no
                    Key Identifier:
                        <snip>
                Identifier:Authority Key Identifier - 2.5.29.35
                    Critical:no
                    Key Identifier:
                        <snip>
!                Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
                     Critical:no
                Identifier:Extended Key Usage:- 2.5.29.37
                     Critical:no
```

```
                     Extended Key Usage:
                     OCSPSigning
!
                Identifier:CRL Distribution Points - 2.5.29.31
                    Critical:no
                    Number of Points:1
                    Point 0
                        Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
        Signature:
            Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
            Signature:
            <snip>
```

次の例は、既存のシーケンスの先頭に **match certificate override ocsp** コマンドを追加したときの実行コンフィギュレーション出力の抜粋を示します。

```
match certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
.
.
.
        match certificate map3 override ocsp 5 url http://192.0.2.3/
        match certificate map1 override ocsp 10 url http://192.0.2.1/
        match certificate map2 override ocsp 15 url http://192.0.2.2/
```

次の例は、既存の **match certificate override ocsp** コマンドが置き換えられ、トラストポイントが代替の PKI 階層を使用するように指定された場合の、実行コンフィギュレーション出力の抜粋を示します。

```
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.0.2.3/
        match certificate map1 override ocsp trustpoint tp1 10 url http://192.0.2.1/
        match certificate map4 override ocsp trustpoint tp4 10 url
http://192.0.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.0.2.2/
```

トラブルシューティングのヒント

失効チェックまたは失効した証明書を無視した場合は、慎重に設定を確認する必要があります。証明書マップが、当該の証明書または許可する証明書、あるいはスキップする AAA チェックのいずれかと適切に一致していることを確認してください。管理された環境で、証明書マップを変更して想定どおりに機能していないものを判別します。

証明書チェーンの設定

ピア証明書の証明書チェーン パスに処理レベルを設定するには、次の作業を実行します。

はじめる前に

- デバイスを PKI 階層に登録する必要があります。
- 適切なキー ペアを証明書に関連付ける必要があります。

（注）

- ルート CA に関連付けられたトラストポイントは、次のレベルに対して有効になるように設定できません。

chain-validation コマンドは、ルート CA に関連付けられたトラストポイントに対して **continue** キーワードとともに設定します。エラー メッセージが表示され、チェーン検証はデフォルトの **chain-validation** コマンド設定に戻ります。

手順の概要

1. **enable**
2. **configureterminal**
3. crypto pki trustpoint *name*
4. **chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]]
5. **exit**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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 ca-sub1`	トラストポイントおよび設定された名前を宣言して、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	**chain-validation** [{**stop** | **continue**} [*parent-trustpoint*]] 例： `Router(ca-trustpoint)# chain-validation continue ca-sub1`	証明書チェーンが、すべての証明書（下位 CA 証明書を含む）で処理されるレベルを設定します。 • **stop** キーワードを使用して、証明書がすでに信頼できることを明示します。これがデフォルト設定です。 • **continue** キーワードを使用して、トラストポイントに関連付けられた下位 CA 証明書を有効にする必要があることを明示します。

	コマンドまたはアクション	目的
		• *parent-trustpoint* 引数は、証明書を照合する必要がある親トラストポイント名を指定します。
ステップ 5	**exit** 例： `Router(ca-trustpoint)# exit`	グローバル コンフィギュレーション モードに戻ります。

証明書サーバのハイ アベイラビリティの設定

取り消しコマンドを同期させ、新しい証明書を発行するときにシリアル番号コマンドを送信するように証明書サーバを設定し、アクティブになった場合に証明書と CRL を発行できるように、スタンバイ証明書サーバを準備することができます。

前提条件

証明書サーバのハイ アベイラビリティを確保するには、次の条件を満たす必要があります。

- IPsec 保護された SCTP は、アクティブ ルータとスタンバイ ルータの両方で設定する必要があります。
- 同期を機能させるには、SCTP を設定した後に、証明書サーバの冗長性モードを ACTIVE/STANDBY に設定する必要があります。

ここでは、次の内容について説明します。

証明書サーバの冗長性モードの ACTIVE/STANDBY の設定

この作業は、証明書サーバの冗長性モードを ACTIVE/STANBY に設定することで、同期をイネーブルにするためにアクティブ ルータで実行します。

1. **configureterminal**
2. **redundancyinter-device**
3. **schemestandby***standby-group-name*
4. **exit**
5. **interface***interface-name*
6. **ipaddress***ip-addressmask*
7. **noiproute-cachecef**

8 **noiproute-cache**

9 **standbyip***ip-address*

10 **standbypriority***priority*

11 **standbyname***group-name*

12 **standbydelayminimum** [*min-seconds*] **reload** [*reload-seconds*

13 スタンバイ ルータに対してステップ 1 ～ 12 を繰り返し、アクティブ ルータの IP アドレスとは異なる IP アドレスを使用してインターフェイスを設定します（ステップ 6）。

14 **exit**

15 **exit**

16 **showcryptokeymypubkeyrsa**

手順の概要

1. **configureterminal**
2. **redundancyinter-device**
3. **schemestandby***standby-group-name*
4. **exit**
5. interface *interface-name*
6. **ipaddress***ip-addressmask*
7. **noiproute-cachecef**
8. **noiproute-cache**
9. standby ip *ip-address*
10. standby priority *priority*
11. standby name *group-name*
12. standby delay minimum **[** *min-seconds* **]** reload [*reload-seconds*]
13. スタンバイ ルータに対してステップ 1 ～ 12 を繰り返し、アクティブ ルータのインターフェイスの IP アドレス（ステップ 6）とは異なる IP アドレスを使用して、インターフェイスを設定します。
14. **exit**
15. **exit**
16. **showredundancystates**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 2	**redundancyinter-device** 例： `Router(config)# redundancy inter-device`	冗長性を設定し、デバイス内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	**schemestandby***standby-group-name* 例： `Router(config-red-interdevice)# scheme standby SB`	使用する冗長性スキームを定義します。 • サポートされているスキームは「standby」だけです。 • *standby-group-name*：**standby name** インターフェイス コンフィギュレーション コマンドで指定したスタンバイ名と一致させる必要があります。また、スタンバイ名は両方のルータで同じである必要があります。
ステップ 4	**exit** 例： `Router(config-red-interdevice)# exit`	デバイス内コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。

ステップ 5	interface *interface-name*
	
	例：
	
	`Router(config)`
	`# interface gigabitethernet0/1`
		ルータのインターフェイスタイプを設定し、インターフェイス コンフィギュレーション モードを開始します。

ステップ 6	**ipaddress***ip-addressmask* 例： `Router(config-if) ip address 10.0.0.1 255.255.255.0`	インターフェイスにローカル IP アドレスを設定します。
ステップ 7	**noiproute-cachecef** 例： `Router(config-if)# no ip route cache cef`	インターフェイスでシスコエクスプレスフォワーディングの動作をディセーブルにします。
ステップ 8	**noiproute-cache** 例： `Router(config-if)# no ip route cache`	インターフェイスで高速スイッチングをディセーブルにします。

	コマンドまたはアクション	目的
ステップ 9	standby ip *ip-address* **例：** `Router(config-if)# standby ip 10.0.0.3`	ホットスタンバイ ルータ プロトコル（HSRP）をアクティブにします。 （注） アクティブ ルータおよびスタンバイ ルータに同じアドレスを設定します。
ステップ 10	standby priority *priority* **例：** `Router(config-if)# standby priority 50`	HSRP のプライオリティを 50 に設定します。 指定できるプライオリティの範囲は 1 ～ 255 です。1 は一番低いプライオリティ、255 は一番高いプライオリティを意味します。HSRP グループ内の最高のプライオリティ値が設定されたルータがアクティブ ルータになります。
ステップ 11	standby name *group-name* **例：** `Router(config-if)# standby name SB`	スタンバイ グループの名前を設定します。 • 名前には、使用されている HSRP グループを指定します。HSRP グループ名はそのルータで一意である必要があります。
ステップ 12	standby delay minimum **[** *min-seconds* **]** reload [*reload-seconds*] **例：** `Router(config-if)# standby delay minimum` **30** `reload` *60*	HSRP グループの初期化の遅延を次のように設定します。 • インターフェイスがアップした後に HSRP グループを初期化するまでの遅延の最小値は 30 秒です。 • ルータがリロードされた後の遅延は 60 秒です。
ステップ 13	スタンバイ ルータに対してステップ 1 ～ 12 を繰り返し、アクティブ ルータのインターフェイスの IP アドレス（ステップ 6）とは異なる IP アドレスを使用して、インターフェイスを設定します。	--
ステップ 14	**exit** **例：** `Router(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 15	**exit** **例：** `Router(config)# exit`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 16	**showredundancystates** 例： `Router# show redundancy states`	（任意）冗長性の状態（スタンバイまたはアクティブ）を確認します。

アクティブおよびスタンバイ証明書サーバでの SCTP の設定

この作業は、アクティブおよびスタンバイの両方の証明書サーバで SCTP を設定するためにアクティブ ルータで実行します。

手順の概要

1. **configureterminal**
2. **ipczonedefault**
3. **association***association-ID*
4. **noshutdown**
5. **protocolsctp**
6. **local-port***local-port-number*
7. **local-ip** device-real-ip-address [device-real-ip-address2]
8. **exit**
9. **remote-port***remote-port-number*
10. **remote-ip***peer-real-ip-address*
11. スタンバイ ルータに対してステップ 1 ～ 10 を繰り返し、ステップ 7 とステップ 10 で指定したローカル ピアおよびリモート ピアの IP アドレスを逆にします。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 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 2	**ipczonedefault** 例： `Router(config)# ipc zone default`	デバイス内通信プロトコルである、Inter-Process Communication（IPC）を設定し、IPC ゾーン コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
		このコマンドを使用して、アクティブ ルータとスタンバイ ルータとの間の通信リンクを開始します。
ステップ 3	**association***association-ID* 例： `Router(config-ipczone)# association 1`	2 つのデバイス間におけるアソシエーションを設定し、IPC アソシエーション 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	**noshutdown** 例： `Router(config-ipczone-assoc)# no shutdown`	サーバ アソシエーションがデフォルトの状態（イネーブル）であることを確認します。
ステップ 5	**protocolsctp** 例： `Router(config-ipczone-assoc)# protocol sctp`	SCTP をトランスポート プロトコルとして設定し、SCTP プロトコル コンフィギュレーション モードを開始します。
ステップ 6	**local-port***local-port-number* 例： `Router(config-ipc-protocol-sctp)# local-port 5000`	冗長ピアとの通信に使用されるローカル SCTP ポート番号を定義して、IPC トランスポート SCTP ローカル コンフィギュレーション モードを開始します。 • *local-port-number*：デフォルト値は存在しません。デバイス内の冗長性をイネーブルにするには、この引数によってローカルポートの設定を行う必要があります。有効なポート値：1 ～ 65535。ローカル ポート番号は、ピア ルータ上のリモート ポート番号と同じにする必要があります。
ステップ 7	**local-ip** device-real-ip-address [device-real-ip-address2] 例： `Router(config-ipc-local-sctp)# local-ip 10.0.0.1`	冗長ペアと通信を行うために使用されるローカル IP アドレスを最低 1 つ定義します。 • ローカル IP アドレスは、ピア ルータ上のリモート IP アドレスと一致している必要があります。1 つまたは 2 つの IP アドレスを指定できます。このアドレスはグローバルな VPN ルーティングおよび転送（VRF）のものである必要があります。仮想 IP アドレスは使用できません。
ステップ 8	**exit** 例： `Router(config-ipc-local-sctp)# exit`	IPC トランスポート - SCTP ローカル 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 9	**remote-port***remote-port-number* 例： `Router(config-ipc-protocol-sctp)# remote-port 5000`	冗長ピアとの通信に使用されるリモート SCTP ポート番号を定義して、IPC トランスポート SCTP リモート コンフィギュレーション モードを開始します。 （注） *remote-port-number*：デフォルト値は存在しません。デバイス内の冗長性をイネーブルにするには、この引数によってリモート ポートの設定を行う必要があります。有効なポート値：1 ～ 65535。リモート ポート番号は、ピア ルータ上のローカル ポート番号と同じにする必要があります。
ステップ 10	**remote-ip***peer-real-ip-address* 例： `Router(config-ipc-remote-sctp)# remote-ip 10.0.0.2`	ローカル デバイスとの通信に使用される冗長ピアのリモート IP アドレスを定義します。 すべてのリモート IP アドレスによって同じデバイスが参照される必要があります。 仮想 IP アドレスは使用できません。
ステップ 11	スタンバイ ルータに対してステップ 1 ～ 10 を繰り返し、ステップ 7 とステップ 10 で指定したローカル ピアおよびリモート ピアの IP アドレスを逆にします。	仮想IPアドレス（10.0.0.3）は、両方のルータで同じになります。

アクティブ証明書サーバとスタンバイ証明書サーバの同期

この作業は、アクティブ サーバとスタンバイ サーバを同期にするために実行します。

手順の概要

1. **configure terminal**
2. **crypto key generate rsa general-keys redundancy label***key-labe***modulus***modulus-size*
3. **exit**
4. **show crypto key mypubkey rsa**
5. **configure terminal**
6. **ip http server**
7. **crypto pki server***cs-label*
8. **redundancy**
9. **no shutdown**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**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 2	**crypto key generate rsa general-keys redundancy label***key-labe***modulus***modulus-size* 例： `Router (config)# crypto key generate rsa general-keys redundancy label HA modulus 2048`	証明書サーバの HA という名前の RSA キー ペアを生成します。 （注） **redundancy** キーワードを指定すると、キーはエクスポート不可能であることを意味します。
ステップ 3	**exit** 例： `Router(config)# exit`	特権 EXEC モードに戻ります。
ステップ 4	**show crypto key mypubkey rsa** 例： `Router# show crypto key mypubkey rsa`	冗長性がイネーブルであることを確認します。
ステップ 5	**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 6	**ip http server** 例： `Router(config)# ip http server`	ご使用のシステムの HTTP サーバをイネーブルにします。
ステップ 7	**crypto pki server***cs-label* 例： `Router(config)# crypto pki server HA`	ステップ 2 で生成した RSA キー ペアを証明書サーバのラベルに指定し、証明書サーバのコンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 8	**redundancy** 例： `Router (cs-server)# redundancy`	サーバがスタンバイ サーバと同期されていることを確認します。
ステップ 9	**no shutdown** 例： `Router(cs-server)# no shutdown`	証明書サーバをイネーブルにします。 （注） SCRP トラフィックを使用するルータインターフェイスが保護されていない場合、ハイ アベイラビリティ デバイス間の SCTP トラフィックが IPsec を使用して保護されていることを確認します。

証明書の許可および失効の設定例

PKI AAA 認可の設定および検証例

ここでは、PKI AAA 認可の設定例を示します。

ルータの設定例

次の **showrunning-config** コマンド出力は、AAA サーバ機能との PKI 統合を使用して、VPN 接続を許可するように設定されたルータの動作設定を示します。

```
Router# show running-config
Building configuration...
!
version 12.3
!
hostname router7200router7200
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
 enrollment url http://192.0.2.33:80
```

```
 serial-number
 crl optional
 rsakeypair STOREVPN 2048
 auto-enroll
 authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
 certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
  31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
  55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
  312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
  30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
  7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
  FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
  16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
  030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
  341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
  12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
  08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
  15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
  EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
  31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
  55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
  589223AB 99A7DC14 04F74EF2 AAEEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
  54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
  E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
  22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
  30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
  F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
  BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
  0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
  12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
  3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
 encr aes
 group 14
!
!
crypto ipsec transform-set ISC_TS_1 esp-aes esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
 set security-association lifetime kilobytes 530000000
 set security-association lifetime seconds 14400
 set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
 description MGRE Interface provisioned by ISC
 bandwidth 10000
 ip address 192.0.2.172 255.255.255.0
 no ip redirects
 ip mtu 1408
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 500
 ip nhrp server-only
```

```
 no ip split-horizon eigrp 101
 tunnel source FastEthernet2/1
 tunnel mode gre multipoint
 tunnel key 101
 tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
 ip address 192.0.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 ip address 192.0.2.2 255.255.255.0
 duplex auto
 speed auto
!
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end
```

成功した PKI AAA 認可のデバッグ例

次の **show debugging** コマンド出力は、AAA サーバ機能との PKI 統合を使用して、成功した許可を示します。

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
Crypto PKI Trans debugging is on
Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD5.example.com)
May 28 19:36:12.813: TPLUS: Using server 192.0.2.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is
 up: new adjacency
Router#
Router# show crypto isakmp sa
```

```
dst             src             state          conn-id slot
192.0.2.22     192.0.2.102   QM_IDLE            84    0
```

失敗した PKI AAA 認可のデバッグ例

次の **show debugging** コマンド出力は、ルータが、VPN を使用しての接続を許可されていないことを示します。このメッセージは、このような状況で表示される典型的なメッセージです。

この例においてピア ユーザ名は、Cisco Secure ACS の VPN_Router_Disabled と呼ばれる Cisco Secure ACS グループに移動することにより、許可されていないものとして設定されました。ルータ（router7200.example.com）は、任意のピアに VPN 接続を確立する前に、Cisco Secure ACS AAA サーバに確認するように設定されています。

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing
May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD5.example.com)
May 28 19:48:31.533: TPLUS: Using server 192.0.2.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
 <all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD5.example.com)
May 28 19:48:41.505: TPLUS: Using server 198.168.244.55
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
```

```
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
Router#
Router# show crypto iskmp sa
dst             src             state          conn-id slot
192.0.2.2        192.0.2.102   MM_KEY_EXCH         95    0
```

失効メカニズムの設定例

ここでは、PKI の失効メカニズムを指定する際に使用できる設定例を示します。

OCSP サーバの設定例

次の例では、証明書の AIA 拡張部で指定された OCSP サーバを使用するようにルータを設定する方法を示します。

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

CRL および OCSP サーバの指定例

次の例では、CRL を CDP からダウンロードするようにルータを設定する方法を示します。CRL を利用できない場合は、証明書の AIA 拡張部で指定される OCSP サーバが使用されます。両方のオプションが失敗した場合、証明書の検証も失敗します。

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

OCSP サーバの設定例

以下に、HTTP URL 「http://myocspserver:81」にある OCSP サーバを使用するようにルータを設定する例を示します。このサーバがダウンしている場合は、失効チェックは行われません。

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

OCSP サーバとの通信でのナンスのディセーブル例

次の例は、OCSP 要求に関するナンス（固有識別情報）が、OCSP サーバとの通信でディセーブルになっている場合の通信を示します。

```
Router(config)# crypto pki trustpoint mytp
```

```
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
Router(ca-trustpoint)# ocsp disable-nonce
```

セントラル サイトにあるハブ ルータを証明書失効チェック用に設定する例

次の例では、複数のブランチ オフィスにセントラル サイトへの接続を提供しているセントラル サイトにあるハブ ルータを示します。

ブランチ オフィスも追加の IPSec トンネルを使用して、ブランチ オフィス間で直接相互に通信できます。

CA は、セントラル サイトにある HTTP サーバの CRL を公開します。セントラル サイトは、各ピアと IPSec トンネルを設定する場合、そのピアの CRL をチェックします。

次の例では、IPSec 設定を示しません。PKI 関連の設定だけを示します。

ホーム オフィスのハブ設定

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

セントラル サイトのハブ ルータ

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Central VPN Gateway
    cn=Central VPN Gateway
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Sep 26 2004
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: VPN-GW
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
```

```
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: VPN-GW
```

ブランチ オフィス ルータのトラストポイント

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none

ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
```

証明書マップがブランチ オフィス ルータに入力されます。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

セントラル サイトのハブ ルータ上で発行された **show certificate** コマンドの出力では、証明書が以下によって発行されたことを示しています。

```
cn=Central Certificate Authority
o=Home Office Inc
```

この 2 行は、行を区切るためのカンマ（,）を使用して 1 行に結合され、元の 2 行が最初の一致基準として追加されています。

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be shown on one line with the line above it.
```

セントラル サイト ルータの証明書の所有者名についても、同じように組み合わされています（「Name:」で始まる行は、所有者名の一部ではなく、証明書マップ基準を作成する際に無視する必要があることに注意してください）。これが証明書マップで使用される所有者名です。

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

これで、以前に設定された証明書マップがトラストポイントに追加されます。

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

設定がチェックされます（大部分の設定は示されていません）。

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
```

```
 revocation-check crl
 match certificate central-site skip revocation-check
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
```

今後のピアの証明書との照合のために、発行者名の行と所有者名の行が矛盾しないように再フォーマットされていることに注意してください。

ブランチ オフィスが AAA をチェックする場合は、トラストポイントには次のような行があります。

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
```

証明書マップが上記のように定義されると、次のコマンドがトラストポイントに追加され、セントラル サイト ハブの AAA チェックがスキップされます。

```
match certificate central-site skip authorization-check
```

両方のケースにおいてブランチ サイト ルータは、CRL のチェックまたは AAA サーバと通信するために、セントラル サイトに IPSec トンネルを確立する必要があります。ただし、**match certificate** コマンドと **central-site skip authorization-check**（引数とキーワード）を使用しないと、ブランチ オフィスが CRL または AAA サーバを確認するまで、トンネルを確立することはできません。（**match certificate** コマンドと **central-site skip authorization-check** 引数およびキーワードを使用しないかぎり、トンネルは確立されません）。

ブランチ サイトにあるルータの証明書が失効していて、その証明書を更新するためにセントラル サイトにトンネルを確立する必要がある場合、セントラル サイトで **match certificate** コマンドと **allow expired-certificate** キーワードを使用します。

セントラル サイト ルータのトラストポイント

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

ブランチ 1 サイト ルータのトラストポイント

```
Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
```

```
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Oct 3 2003
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office
```

証明書マップがセントラル サイト ルータに入力されます。

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office
 Inc
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Brahcn 1 Site,o=home office inc
```

証明書マップがトラストポイントに追加されます。

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) #exit
```

設定がチェックされます（設定の大部分は示されていません）。

```
Router# write term
!many lines left out
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

match certificate コマンド、**branch1 allow expired-certificate**（引数とキーワード）および証明書マップは、ブランチ ルータが新しい証明書を取得した後すぐに削除する必要があります。

証明書の許可および失効の設定例

この項では、CRL キャッシュ コントロールの設定または証明書のシリアル番号セッション コントロールを指定する場合に使用する設定例を示します。

CRL キャッシュ コントロールの設定

次の例では、CA1 トラストポイントに関連付けられたすべての CRL の CRL キャッシングをディセーブルにする方法を示します。

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache none
```

上記の例の設定を実行した直後は、まだ現在の CRL がキャッシュされています。

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

現在の CRL が失効すると、次の更新時に新しい CRL がルータにダウンロードされます。**crl-cache none** コマンドが有効になり、トラストポイントの CRL はすべてキャッシュされなくなります。また、キャッシュはディセーブルになります。**show crypto pki crls** コマンドを実行して、CRL がキャッシュされていないことを確認できます。キャッシュされている CRL がないため、出力は表示されません。

次の例では、CA1 トラストポイントに関連付けられたすべての CRL に 2 分の最大ライフタイムを設定する方法を示します。

```
crypto pki trustpoint CA1
 enrollment url http://CA1:80
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 crl-cache delete-after 2
```

CRL の最大ライフタイムを設定するために上記例の設定を実行した直後でも、依然現在の CRL がキャッシュされます。

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 18:57:42 GMT Nov 26 2005
    NextUpdate: 22:57:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
      ldap://ldap.example.com/CN=name Cert Manager,O=example.com
When the current CRL expires, a new CRL is downloaded to the router at the next update and
 the crl-cache delete-after
command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after
a maximum lifetime of 2 minutes.
You can verify that the CRL will be cached for 2 minutes by executing the show crypto pki
crls
 command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.
```

Router# **show crypto pki crls**

```
CRL Issuer Name:
    cn=name Cert Manager,ou=pki,o=example.com,c=US
    LastUpdate: 22:57:42 GMT Nov 26 2005
```

```
    NextUpdate: 22:59:42 GMT Nov 26 2005
    Retrieved from CRL Distribution Point:
ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

証明書のシリアル番号セッション コントロールの設定

次の例では、CA1 トラストポイントの証明書マップを使用した証明書のシリアル番号セッション コントロールの設定を示します。

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 chain-validation stop
 crl query ldap://ldap_server
 revocation-check crl
 match certificate crl
!
crypto pki certificate map crl 10
 serial-number co 279d
```

（注） *match-criteria* 値が **co**（含む）ではなく **eq**（等しい）に設定されている場合、シリアル番号はスペースを含めて、証明書マップのシリアル番号に正確に一致する必要があります。

次の例では、AAA 属性を使用した証明書のシリアル番号セッション コントロールの設定を示します。この場合、証明書にシリアル番号「4ACA」がなければ、有効な証明書はすべて受け入れられます。

```
crypto pki trustpoint CA1
 enrollment url http://CA1
 ip-address FastEthernet0/0
 crl query ldap://ldap_CA1
 revocation-check crl
 aaa new-model
!
aaa attribute list crl
attribute-type aaa-cert-serial-not 4ACA
```

サーバ ログは、シリアル番号「4ACA」を持つ証明書が拒否されたことを示しています。証明書の拒否は、感嘆符で表示されます。

```
.
.
.
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: validation path has 1 certs
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: Certificate validated without revocation check
Dec 3 04:24:39.135: CRYPTO_PKI: Selected AAA username: 'PKIAAA'
Dec 3 04:24:39.135: CRYPTO_PKI: Anticipate checking AAA list:'CRL'
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: checking AAA authorization (CRL, PKIAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x4)
Dec 3 04:24:39.135: AAA/BIND(00000021): Bind i/f
Dec 3 04:24:39.135: AAA/AUTHOR (0x21): Pick method list 'CRL'
.
.
.
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
```

```
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: cert-serial doesn't match ("4ACA" != "4ACA")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: post-authorization chain validation status (0x7)
!
Dec 3 04:24:39.175: CRYPTO_PKI: AAA authorization for list 'CRL', and user 'PKIAAA' failed.
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain
validation result was: CRYPTO_PKI_CERT_NOT_AUTHORIZED
!
Dec 3 04:24:39.175: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.0.2.43 is bad:
 certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer
at 192.0.2.43
.
.
.
```

証明書チェーン検証の設定例

この項では、デバイス証明書の証明書チェーン処理レベルを指定する場合に使用する設定例を示します。

ピアからルート CA への証明書チェーン検証の設定

次の設定例では、ピア、SubCA11、SubCA1、および RootCA のすべての証明書が検証されます。

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
 revocation-check none
 rsakeypair SubCA11
```

ピアから下位 CA への証明書チェーン検証の設定

次の設定例では、ピア証明書および SubCA1 証明書が有効にされます。

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA1
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA1
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue SubCA1
```

```
 revocation-check none
 rsakeypair SubCA11
```

証明書チェーンの欠落確認の設定

次の設定例では、SubCA1 が、設定済みの Cisco IOS 階層にはないが、提出された証明書チェーンでピアによって提示されたと想定しています。

ピアが、提出された証明書チェーンで SubCA1 証明書を提示した場合、ピア、SubCA11、および SubCA1 の各証明書が有効になります。

ピアが、提出された証明書チェーンで SubCA1 証明書を提示しない場合、チェーンの検証は失敗します。

```
crypto pki trustpoint RootCA
 enrollment terminal
 chain-validation stop
 revocation-check none
 rsakeypair RootCA
crypto pki trustpoint SubCA11
 enrollment terminal
 chain-validation continue RootCA
 revocation-check none
 rsakeypair SubCA11
```

証明書サーバのハイ アベイラビリティの設定例

次の例では、SCTP の設定、アクティブおよびスタンバイ証明書サーバの冗長性の設定、およびこれらのサーバ間の同期のアクティブ化を示します。

アクティブ ルータ

```
ipc zone default
 association 1
 no shutdown
 protocol sctp
  local-port 5000
   local-ip 10.0.0.1
   exit
  remote-port 5000
   remote-ip 10.0.0.2
```

スタンバイ ルータ

```
ipc zone default
 association 1
 no shutdown
 protocol sctp
  local-port 5000
   local-ip 10.0.0.2
   exit
  remote-port 5000
   remote-ip 10.0.0.1
```

アクティブ ルータ

```
redundancy inter-device
```

```
 scheme standby SB
interface GigabitEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 no ip route-cache cef
 no ip route-cache

 standby 0 ip 10.0.0.3
 standby 0 priority 50
 standby 0 name SB
 standby delay min 30 reload 60
```

スタンバイ ルータ

```
redundancy inter-device
 scheme standby SB
interface GigabitEthernet0/1
 ip address 10.0.0.2 255.255.255.0
 no ip route-cache cef
 no ip route-cache

 standby 0 ip 10.0.0.3
 standby 0 priority 50
 standby 0 name SB
 standby delay min 30 reload 60
```

アクティブ ルータ

```
crypto pki server mycertsaver
crypto pki server mycertsaver redundancy
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
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*』
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」モジュール
RSA キーの生成および展開	「PKI 内での RSA キーの展開」モジュール
証明書登録：サポートされる方法、登録プロファイル、設定作業	「PKI の証明書登録の設定」モジュール
Cisco IOS 証明書サーバの概要および設定作業	「PKI 展開での Cisco IOS 証明書サーバの設定および管理」モジュール
推奨暗号化アルゴリズム	『Next Generation Encryption』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

証明書の許可および失効に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *3*：*PKI* 証明書の許可および失効に関する機能情報

機能名	リリース	機能情報
認証失効リストのキャッシュコントロール拡張機能	Cisco IOS XE Release 2.4	この機能を使用すると、ユーザは CRL キャッシングをディセーブルにしたり、ルータのメモリに CRL がキャッシュされる最大ライフタイムを指定したりできます。この機能は、証明書のシリアル番号セッションコントロールを設定するための機能も提供します。 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。**crl-cachedelete-after**、**crl-cachenone**、**cryptopkicertificatemap**

機能名	リリース	機能情報
Certificate-Complete チェーンの検証	Cisco IOS XE Release 2.4	この機能を使用すると、すべての証明書（下位 CA 証明書を含む）で証明書チェーンが処理されるレベルを設定できます。 この機能により、次のコマンドが導入されました。 **chain-validation**
OCSP：代替階層からのサーバ認証	Cisco IOS XE Release 2.4	この機能は、複数 OCSP サーバをクライアント証明書ごとに、またはクライアント証明書のグループごとに指定できる柔軟性を備えています。また、この機能を使用すると、外部の CA 証明書または自己署名証明書に基づいて OCSP サーバを検証できます。 この機能により、**matchcertificateoverrideocsp** コマンドが導入されました。
オプションの OCSP ナンス	Cisco IOS XE Release 2.1	この機能では、OCSP 通信時にナンス（OCSP 要求に関する固有識別情報）を送信するように設定できます。

機能名	リリース	機能情報
証明書のセキュリティ属性ベースのアクセス コントロール	Cisco IOS XE Release 2.1	IPsec プロトコルでは、CA の相互運用性により、Cisco IOS デバイスと CA が通信を行い、Cisco IOS デバイスは、CA からデジタル証明書を取得し、使用できるようになります。証明書には、指定された処理の実行をデバイスまたはユーザが許可されているかどうかの判別に使用されるフィールドがいくつか含まれています。この機能により、ACL の指定が可能な証明書にフィールドを追加し、証明書ベース ACL を作成できます。 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。**cryptopkicertificatemap**、**cryptopkitrustpointmatchcertificate**
Online Certificate Status Protocol（OCSP）	Cisco IOS XE Release 2.1	この機能により、CRL の代わりに OCSP をイネーブルにして、証明書のステータスをチェックできます。証明書のステータスを定期的に提供するだけの CRL とは異なり、OCSP では証明書ステータスに関する情報をタイムリーに利用できます。 この機能により、**ocspurl** および **revocation-check** コマンドが導入されました。
所有者名全体を使用した PKI AAA 認可	Cisco IOS XE Release 2.1	この機能により、ユーザは、所有者名全体を一意の AAA ユーザ名として使用し、証明書から AAA サーバを照会できます。 この機能により、**authorizationusername** コマンドが変更されました。

機能名	リリース	機能情報
AAA サーバとの PKI 統合	Cisco IOS XE Release 2.1	この機能では、ピアによって提出された証明書から AAA ユーザ名を生成することにより、許可に関するスケーラビリティが向上します。AAA サーバは、内部コンポーネントでの証明書の使用を許可するか決定するよう尋ねられます。許可は、コンポーネントで指定されたラベルによって示され、このラベルはユーザの AV ペアに存在している必要があります。 この機能により、**authorizationlist** および **authorizationusername** コマンドが導入されました。
PKI：証明書失効チェック時の複数のサーバ照会	Cisco IOS XE Release 2.1	Cisco IOS ソフトウェアではこの機能により、特定のサーバが利用できない場合に操作を続行できるように CRL の取得を複数回試行できます。また、証明書の CDP を、手動で設定した CDP で上書きすることもできます。証明書の CDP の手動による上書きは、特定のサーバが長時間利用できない場合に便利です。元の CDP を含む証明書のすべてを再発行しなくても、証明書の CDP を URL またはディレクトリ指定に置き換えることができます。 この機能により、**matchcertificateoverridecdp** コマンドが導入されました。

機能名	リリース	機能情報
証明書 ACL を使用して失効チェックおよび失効した証明書の無視	Cisco IOS XE Release 2.1	この機能により、指定基準を満たす証明書は、証明書の有効期間にかかわらず受け入れることができます。また、証明書が指定基準を満たしている場合は失効チェックを実行する必要がなくなります。証明書 ACL は、証明書を受け入れるために満たす必要がある基準を指定する場合や、失効チェックを回避する場合に使用されます。さらに、AAA 通信が証明書によって保護されている場合、この機能は無視される証明書に対して AAA チェックを実行します。 この機能により、**matchcertificate** コマンドが変更されました。
トラストポイントごとのクエリー モードの定義	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 シリーズ ルータで導入されました。
PKI ハイ アベイラビリティ	Cisco IOS XE Release 3.2S	次のコマンドが導入または変更されました。**cryptopkiserver**、**cryptopkiserverstart**、**cryptopkiserverstop**、**cryptopkitrustpoint**、**cryptokeygeneratersa**、**cryptokeyimportpem**、**cryptokeymoversa**、**showcryptokeymypubkeyrsa**。

第 5 章

PKI の証明書登録の設定

この章では、証明書登録に利用可能なさまざまな方式および参加する PKI ピアの各セットアップ方法について説明します。証明書登録は、認証局（CA）から証明書を取得するプロセスであり、証明書を要求するエンドホストと CA の間で発生します。公開キーインフラストラクチャ（PKI）に参加する各ピアは、CA に登録する必要があります。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

PKI 証明書登録の前提条件

証明書登録用にピアを設定する前に、次のものを準備、あるいは次の作業を実行することが必要です。

- 登録用に生成された Rivest、Shamir、Adelman（RSA）キー ペアおよび登録する PKI。
- 認証された CA。
- 「Cisco IOS PKI Overview: Understanding and Planning a PKI」の内容を理解していること。
- 自動登録と証明書ロールオーバーなどの PKI サービスが正しく動作するように、デバイスの NTP を有効にします。

（注） Cisco IOS Release 12.3(7)T では、「**cryptoca**」で始まるすべてのコマンドが、「**cryptopki**」から始まるように変更されました。ルータは引き続き **cryptoca** コマンドを受信しますが、出力はすべて **cryptopki** と表示されます。

PKI の証明書登録に関する情報

CA とは

CA は他の通信相手が使用できるデジタル証明書を発行するエンティティです。これが、信頼できる第三者の例です。CA は多くの PKI スキームの特性です。

CA は証明書要求を管理し、参加ネットワーク装置に証明書を発行します。これらのサービスでは、身元情報を検証してデジタル証明書を作成するために、参加装置のキーを一元的に管理します。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。

Cisco IOS 証明書サーバまたはサードパーティの CA ベンダーが指定する CA を使用できます。

複数の CA のためのフレームワーク

PKI は、複数の CA をサポートするために階層型フレームワーク内に設定できます。階層の最上位にはルート CA があり、自己署名証明書を保持しています。階層全体の信頼性は、ルート CA の RSA キー ペアから導出されます。階層構造内の下位 CA は、ルート CA または別の下位 CA に登録できます。CA の複数の階層が、ルート CA または別の下位 CA で設定されます。階層型 PKI

内では、登録されているすべてのピアが信頼できるルート CA 証明書または共通の下位 CA を共有している場合、証明書を相互に検証できます。

複数 CA を使用する場合

複数 CA を使用することにより、柔軟性および信頼性が向上します。たとえば、ルート CA を本社オフィスに配置し、下位 CA をブランチ オフィスに配置できます。また、CA ごとに異なる許可ポリシーを実行できるため、階層構造内の、ある CA では各証明書要求を手動で許可する必要があるように、別の CA では証明書要求を自動的に許可するように設定できます。

少なくとも 2 階層の CA が推奨されるシナリオは、次のとおりです。

- 多数の証明書が失効し、再発行される大規模かつ非常にアクティブなネットワーク。複数の階層を使用することにより、CA は証明書失効リスト（CRL）のサイズを制御しやすくなります。
- 下位の CA 証明書を発行する場合を除いて、オンラインの登録プロトコルが使用されているときは、ルート CA をオフラインにしておくことができます。このシナリオでは、ルート CA のセキュリティが向上します。

CA の認証

装置に自身の証明書が発行されて証明書登録が発生する前に、CA の証明書が認証される必要があります。CA の認証は通常、ルータで PKI サポートを初期設定するときにだけ実行されます。CA を認証するには、**crypto pki authenticate** コマンドを発行します。これにより、CA の公開キーが組み込まれた CA の自己署名証明書が取得されて CA がルータに対して認証されます。

fingerprint コマンドによる認証

Cisco IOS Release 12.3(12) 以降では、**fingerprint** コマンドを発行して、認証時に CA 証明書のフィンガープリントと照合するフィンガープリントを事前入力できます。

フィンガープリントがトラスト ポイントにあらかじめ入力されていない場合や、認証要求がインタラクティブでない場合は、CA 証明書の認証時に表示されるフィンガープリントを検証する必要があります。認証要求がインタラクティブでない場合、事前入力フィンガープリントがないと、証明書は拒否されます。

（注） 認証要求がコマンドライン インターフェイス（CLI）を使用して行われる場合、その要求はインタラクティブな要求です。認証要求が HTTP または別の管理ツールを使用して行われる場合、その要求はインタラクティブでない要求です。

サポートされる証明書の登録方式

Cisco IOS ソフトウェアは、CA から証明書を取得するために次の方式をサポートしています。

- Simple Certificate Enrollment Protocol（SCEP）：HTTP を使用して CA または登録局（RA）と通信する、シスコが開発した登録プロトコル。SCEP は、要求および証明書の送受信用に最も一般的に使用される方式です。

（注） 自動証明書およびキー ロールオーバー機能を活用するには、ロールオーバーをサポートする CA を実行する必要があります。また、クライアント登録方式として SCEP を使用する必要があります。Cisco IOS CA を実行する場合は、ロールオーバーをサポートするために Cisco IOS Release 12.4(2)T 以降のリリースを実行する必要があります。

- PKCS12：ルータは、外部のサーバから証明書を PKCS12 形式でインポートします。
- IOS ファイル システム（IFS）：ルータは、Cisco IOS ソフトウェアでサポートされるファイル システム（TFTP、FTP、フラッシュ、および NVRAM など）を使用して証明書要求を送信し、発行された証明書を受信します。ユーザの CA が SCEP をサポートしない場合、IFS 証明書登録をイネーブルにできます。

（注） Cisco IOS Release 12.3(4)T 以前のリリースでは、IFS 内で TFTP ファイル システムだけがサポートされます。

- 手動でのカットアンドペースト：ルータはコンソール端末に証明書要求を表示し、ユーザはコンソール端末で発行された証明書を入力できます。ルータと CA の間にネットワーク接続がない場合、ユーザは証明書要求および証明書を手動でカットアンドペーストできます。
- 登録プロファイル：ルータは、HTTP ベースの登録要求を RA モードの証明書サーバ（CS）ではなく、CA サーバに直接送信します。CA サーバが SCEP をサポートしない場合に、登録プロファイルを使用できます。
- トラストポイントの自己署名証明書登録：セキュア HTTP（HTTPS）サーバは、セキュア ソケット レイヤ（SSL）ハンドシェイク時に使用される自己署名証明書を生成し、HTTPS サーバとクライアントの間にセキュアな接続を確立します。自己署名証明書は、ルータのスタートアップ コンフィギュレーション（NVRAM）に保存されます。保存された自己署名証明書は、将来の SSL ハンドシェイクに使用できます。これにより、ルータがリロードされる度に、証明書を受け入れるために必要だったユーザによる介入が不要になります。

（注） 自動登録および自動再登録を活用するには、登録方式として、TFTP または手動でのカットアンドペースト登録を使用しないでください。TFTP およびカットアンドペーストによる手動での登録方式は手動の登録プロセスでは、ユーザによる入力が必要です。

PKI の証明書登録のための Cisco IOS Suite-B サポート

Suite B の要件は、IKE および IPSec で使用するための暗号化アルゴリズムの 4 つのユーザ インターフェイス スイートで構成され、RFC4869に記述されています。各スイートは、暗号化アルゴリズム、デジタル署名アルゴリズム、キー合意アルゴリズム、ハッシュまたはメッセージ ダイジェスト アルゴリズムで構成されています。

Suite-B によって、PKI の証明書登録に次のサポートが追加されます。

- X.509 証明書内の署名操作で、楕円曲線デジタル署名アルゴリズム（ECDSA）（256 ビットおよび 384 ビットの曲線）が使用されます。
- ECDSA の署名を使用した X.509 証明書の確認で PKI がサポートされます。
- ECDSA の署名を使用した証明書要求の生成、および発行された証明書の IOS へのインポートで、PKI がサポートされます。

Cisco IOS での Suite-B サポートに関する詳細については、『Configuring Security for VPNs with IPsec』フィーチャ モジュールを参照してください。

登録局

Cisco IOS 証明書サーバは、RA モードで実行できるように設定できます。RA は、CA から認証および認可責任をオフロードします。RA が SCEP または手動での登録要求を受信すると、管理者はローカル ポリシーごとに要求を拒否または許可できます。要求が許可された場合、その要求は発行元 CA に転送されます。また、自動的に証明書を生成して、証明書を RA に返すように CA を設定できます。クライアントは、許可された証明書を RA から後で取得できます。

自動証明書登録

証明書自動登録を使用すると、CA クライアントは、CA サーバから証明書を自動的に要求できます。この自動ルータ要求では、登録要求が CA サーバに送信された時点で、オペレータによる介入が不要になります。自動登録は、設定済みの、有効なクライアント証明書を持っていないトラストポイント CA の起動時に実行されます。証明書が失効すると、新しい証明書が自動的に要求されます。

（注） 自動登録が設定されると、クライアントは自動的にクライアント証明書を要求します。CA サーバは、独自の許可チェックを実行します。このチェックに証明書を自動的に発行するポリシーが含まれている場合は、すべてのクライアントが自動的に証明書を受信しますが、これはそれほど安全ではありません。そのため、自動証明書登録を追加の認証および許可メカニズム（既存の証明書およびワンタイム パスワードを活用した Secure Device Provisioning（SDP）など）と組み合わせる必要があります。

自動クライアント証明書およびキー ロールオーバー

デフォルトでは、自動証明書登録機能により、クライアントの現在の証明書が失効する前に、CS から新しいクライアント証明書とキーが要求されます。証明書およびキー ロールオーバーにより、新しいキーおよび証明書、ロールオーバー、証明書が利用可能になるまで、現在のキーおよび証明書を保持して証明書が失効する前に証明書更新ロールオーバー要求を行うことができます。指定された時間が経過すると、ロールオーバー証明書およびキーがアクティブになります。失効した証明書およびキーは、ロールオーバー時にただちに削除され、証明書チェーンおよびCRLから削除されます。

自動ロールオーバーのセットアップは 2 段階で行われます。まず CA クライアントが自動的に登録され、クライアントの CA が自動的に登録される必要があります。さらに **auto-rollover** コマンドがイネーブルになる必要があります。CA サーバを自動証明書ロールオーバー用に設定する場合の詳細については、『*Public Key Infrastructure Configuration Guide*』の「Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment」の章にある「Automatic CA Certificate and Key Rollover」の項を参照してください。

任意の renewal percentage パラメータを **auto-enroll** コマンドと一緒に使用すると、証明書の指定されたパーセンテージの有効期間が経過したときに、新しい証明書を要求できます。たとえば、更新パーセンテージが 90 に設定され、証明書の有効期間が 1 年の場合は、古い証明書が失効する 36.5 日前に新しい証明書が要求されます。自動ロールオーバーが発生するには、更新パーセンテージが 100 未満である必要があります。指定するパーセント値は、10 以上でなくてはなりません。CA 証明書の失効が差し迫っているため、有効設定期間よりも短い期間のクライアント証明書を発行する場合、その期間の残り日数に対してロールオーバー証明書が発行されます。最低でも、設定されている有効期間の 10 % と、ロールオーバーが機能するのに十分な時間（絶対最小値：3 分）を見込んでおく必要があります。

ヒント　CA 自動登録がイネーブルになっておらず、現在のクライアント証明書の有効期間が、対応する CA 証明書の有効期間と同じか、それよりも長い場合は、**cryptopkienroll** コマンドを使用して既存のクライアント上で手動でロールオーバーを開始できます。クライアントはロールオーバー プロセスを開始しますが、このプロセスは、サーバが自動ロールオーバーに設定され、利用可能なロールオーバー サーバ証明書を保持している場合にだけ発生します。

（注）　キー ペアが **auto-enrollre-generate** コマンドおよびキーワードによって設定されている場合は、キー ペアも送信されます。新しいキー ペアは、セキュリティ上の問題に対処するために発行することを推奨します。

証明書登録プロファイル

登録プロファイルを使用すると、証明書認証、登録および再登録の各パラメータを指定するよう求められたときにユーザは、これらのパラメータを指定できます。これらのパラメータ値は、プロファイルを構成する 2 つのテンプレートによって参照されます。このうち、1 つのテンプレートには、CA の証明書を取得するために CA サーバに送られる HTTP 要求のパラメータ（証明書認

証としても知られる）が含まれ、もう 1 つのテンプレートには、証明書を登録するために CA に送られる HTTP 要求のパラメータが含まれます。

2 つのテンプレートを設定すると、ユーザは、証明書の認証と登録用に異なる URL または方法を指定できます。たとえば、認証（CA の証明書の取得）を TFTP によって（**authentication url** コマンドを使用して）実行できる一方で、（**enrollment terminal** コマンドを使用して）登録を手動で実行できます。

Cisco IOS Release 12.3(11)T 以前のリリースでは、証明書要求は PKCS10 形式でしか送信できませんでしたが、現在では、プロファイルにパラメータが追加されたことにより、証明書更新要求用に PKCS7 形式を指定できるようになりました。

（注） 1 つの登録プロファイルには、タスクごとに最大 3 つのセクション（証明書の認証、登録および再登録）を指定できます。

PKI の証明書登録を設定する方法

ここでは、次の登録の任意手順について説明します。登録または自動登録を設定する（最初の作業）場合は、手動での証明書登録を設定できません。また、TFTP またはカットアンドペーストによる手動での証明書登録を設定した場合、自動登録、自動再登録、登録プロファイルは設定できず、自動 CA 証明書ロールオーバー機能も利用できません。

証明書登録または自動登録の設定

PKI に参加しているクライアントの証明書登録を設定するには、次の作業を実行します。

はじめる前に

自動証明書登録要求を設定する前に、必要な登録情報がすべて設定されていることを確認する必要があります。

自動クライアント証明書およびキー ロールオーバーをイネーブルにするための前提条件

自動登録を使用するときには、証明書ロールオーバーの CA クライアント サポートが自動的にイネーブルになります。自動 CA 証明書ロールオーバーを正常に実行するには、次の前提条件が適用されます。

- ネットワーク装置はシャドウ PKI をサポートしている必要があります。
- クライアントは Cisco IOS Release 12.4(2)T 以降のリリースを実行している必要があります。
- クライアントの CS は自動ロールオーバーをサポートする必要があります。CA サーバの自動ロールオーバー設定コンフィギュレーションに関する詳細については、『*Public Key Infrastructure Configuration Guide*』の「Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment」の章にある「Automatic CA Certificate and Key Rollover」を参照してください。

自動登録の初期キー生成場所を指定するための前提条件

自動登録の初期キー生成場所を指定するには、Cisco IOS Release 12.4(11)T 以降のリリースを実行する必要があります。

（注）

自動登録の RSA キー ペアに関する制約事項

regenerate コマンドまたは **auto-enroll** コマンドの **regenerate** キーワードを使用して新しいキーペアを生成するように設定したトラストポイントは、他のトラストポイントとキー ペアを共有することはできません。各トラストポイントに独自のキーペアを付与するには、CA トラストポイント コンフィギュレーション モードで **rsakeypair** コマンドを使用します。再生トラストポイント間でのキーペアの共有がサポートされていない場合にキーペアを共有すると、キーと証明書が一致しなくなるため、トラストポイントの一部のサービスが失われます。

自動クライアント証明書およびキー ロールオーバーに関する制約事項

クライアントが自動 CA 証明書ロールオーバーを正常に実行するには、次の制約事項が適用されます。

- SCEP を使用してロールオーバーをサポートする必要があります。SCEP の代わりに証明書管理プロトコルまたはメカニズム（登録プロファイル、手動での登録、または TFTP による登録など）を使用して、PKI に登録する装置では、SCEP で提供されているロールオーバー機能を利用できません。
- シャドウ証明書の生成後に、設定をスタートアップ コンフィギュレーションに保存できない場合、ロールオーバーは発生しません。

>

（注）

セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint***name*
4. **enrollment** [**mode** | **retry period***minutes* | **retry count***number*] **url***url* [**pem**]
5. **eckeypair***label*
6. **subject-name** [*x.500-name*]
7. **vrf***vrf-name*
8. **ip-address** {*ip-address* | *interface* | **none**}
9. serial-number [none]
10. **auto-enroll** [*percent*] [**regenerate**]
11. **usage***method1* [*method2* [*method3*]]
12. **password***string*
13. **rsakeypair***key-labelkey-sizeencryption-key-size*]]
14. **fingerprint***ca-fingerprint*
15. **on***devicename***:**
16. **exit**
17. **cryptopkiauthenticate***name*
18. **exit**
19. **copysystem:running-configvram:startup-config**
20. **showcryptopkicertificates**

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。 • パスワードを入力します（要求された場合）。
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 mytp`	トラストポイントおよび設定された名前を宣言して、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。

	コマンドまたはアクション	目的
ステップ 4	**enrollment** [**mode** | **retry period***minutes* | **retry count***number*] **url***url* [**pem**] 例： `Router(ca-trustpoint)# enrollment url http://cat.example.com`	ルータが証明書要求を送信する CA の URL を指定します。 • **mode**：CA システムが RA を提供する場合は、RA モードを指定します。 • **retryperiod***minutes*：証明書要求を再試行するまでの待機時間を指定します。デフォルトの再試行間隔は 1 分です。 • **retrycount***number*：直前の要求に対する応答をルータが受信しない場合、ルータが証明書要求を再送信する回数を指定します（1 ～ 100 回の範囲で指定できます）。 • **url***url*：ルータが証明書要求を送信するファイル システムの URL。URL 内の IPv6 アドレスは括弧で囲む必要があります。たとえば、http:// [2001:DB8:1:1::1]:80 です。登録方式オプションの詳細については、「enrollment url (ca-trustpoint)」コマンド ページを参照してください。 • **pem**：証明書要求に Privacy Enhanced Mail（PEM）の境界を追加します。 （注） 自動登録をサポートするには、TFTP または手動でのカットアンドペースト以外の登録方式を設定する必要があります。
ステップ 5	**eckeypair***label* 例： `Router(ca-trustpoint)# eckeypair Router_1_Key`	（任意）ECDSA の署名を使用して証明書要求を生成する Elliptic Curve（EC）キーを使用するように、トラストポイントを設定します。*label* 引数は、グローバル コンフィギュレーション モードで **cryptokeygeneratersa** または **cryptokeygenerateeckeysize** コマンドを使用して設定される EC キー ラベルを指定します。詳細については、『Configuring Internet Key Exchange for IPsec VPNs』フィーチャ モジュールを参照してください。 （注） トラストポイントの設定を使用せずに ECDSA の署名を持つ証明書をインポートする場合、ラベルにはデフォルトで FQDN の値が使用されます。
ステップ 6	**subject-name** [*x.500-name*] 例： `Router(ca-trustpoint)# subject-name cat`	（任意）証明書要求で使用される件名を指定します。 • *x.500-name*：この名前が指定されていない場合、完全修飾ドメイン名（FQDN）が使用されます。FQDN はデフォルトの件名です。

	コマンドまたはアクション	目的
ステップ 7	**vrf***vrf-name* 例： `Router(ca-trustpoint)# vrf myvrf`	（任意）登録、証明書失効リスト（CRL）の取得、および Online Certificate Status Protocol（OCSP）のステータに使用される公開キー インフラストラクチャ（PKI）トラストポイントで VRF インスタンスを指定します。
ステップ 8	**ip-address** {*ip-address* | *interface* | **none**} 例： `Router(ca-trustpoint)# ip address 192.168.1.66`	（任意）指定されたインターフェイスの IP アドレスを証明書要求に含めます。 • IPv4 または IPv6 アドレスのいずれかを指定するには、*ip-address* 引数を発行します。 • ルータのインターフェイスを指定するには、*interface* 引数を発行します。 • IP アドレスを含めない場合は、**none** キーワードを発行します。 （注） このコマンドがイネーブルになっている場合、このトラストポイントの登録時に IP アドレスのプロンプトは表示されません。
ステップ 9	serial-number [none] 例： `Router(ca-trustpoint)# serial-number`	（任意）**none** キーワードを発行しない場合は、証明書要求でルータのシリアル番号を指定します。 • 証明書要求にシリアル番号を含めない場合は、**none** キーワードを発行します。
ステップ 10	**auto-enroll** [*percent*] [**regenerate**] 例： `Router(ca-trustpoint)# auto-enroll regenerate`	（任意）自動登録をイネーブルにします。これにより、クライアントは CA から自動的にロールオーバー証明書を要求できます。 • 自動登録イネーブルでない場合、証明書の失効時にクライアントを手動で PKI に再登録する必要があります。 • デフォルトでは、ルータのドメイン ネーム システム（DNS）名だけが証明書に含められます。 • 現行の証明書の有効期間が指定のパーセンテージに達したときに、新しい証明書が要求されるように指定するには、*percent* 引数を使用します。 • 名前付きのキーがすでに存在する場合でも、証明書の新しいキーを生成するには、**regenerate** キーワードを使用します。

	コマンドまたはアクション	目的
		（注） ロールオーバー中のキー ペアがエクスポート可能な場合、新しいキー ペアもエクスポート可能です。次のコメントがトラストポイントコンフィギュレーションに表示され、キー ペアがエクスポート可能かどうかが示されます。「! RSA key pair associated with trustpoint is exportable.」 （注） 新しいキー ペアは、セキュリティ上の問題に対処するために生成することを推奨します。
ステップ 11	**usage***method1* [*method2* [*method3*]] 例： `Router(ca-trustpoint)# usage ssl-client`	（任意）証明書の目的の用途を指定します。 • 指定可能なオプションは **ike**、**ssl-client**、および **ssl-server** です。デフォルトは **ike** です。
ステップ 12	**password***string* 例： `Router(ca-trustpoint)# password string1`	（任意）証明書の失効パスワードを指定します。 • このコマンドがイネーブルになっている場合、このトラストポイントの登録時にパスワードは求められません。 （注） SCEP が使用されている場合、このパスワードを使用して証明書要求を認可できます（多くの場合、ワンタイム パスワードまたは類似のメカニズムによって行われます）。
ステップ 13	**rsakeypair***key-labelkey-sizeencryption-key-size*]] 例： `Router(ca-trustpoint)# rsakeypair key-label 2048 2048`	（任意）証明書に関連付けるキー ペアを指定します。 • *key-label* 付きのキー ペアがまだ存在しない、あるいは **auto-enrollregenerate** コマンドが発行された場合は、登録時にキー ラベル付きのキー ペアが生成されます。 • キーを生成するための *key-size* 引数を指定し、*encryption-key-size* 引数を指定して、個別の暗号化、署名キー、および証明書を要求します。key-size と encryption-key-size は同じサイズでなければなりません。2048 未満の長さを指定することは推奨されません。 （注） このコマンドがイネーブルでない場合に、FQDN キー ペアが使用されます。
ステップ 14	**fingerprint***ca-fingerprint* 例： `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	（任意）認証時に CA 証明書のフィンガープリントと照合するフィンガープリントを指定します。 （注） フィンガープリントが指定されておらず、CA 証明書の認証がインタラクティブな場合、フィンガープリントは検証用に表示されます。

	コマンドまたはアクション	目的
ステップ 15	**on***devicename***:** 例： `Router(ca-trustpoint)# on usbtoken0:`	（任意）自動登録の初期キー生成時に、RSA キーが指定された装置に対して作成されるよう指定します。 • 指定可能な装置には、NVRAM、ローカルディスク、およびユニバーサル シリアル バス（USB）トークンがあります。USB トークンは、ストレージ デバイス以外に、暗号化装置として使用できます。USB トークンを暗号化装置として使用すると、トークンでキー生成、署名、認証などの RSA 操作を実行できます。
ステップ 16	**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、グローバル コンフィギュレーション モードに戻ります。
ステップ 17	**cryptopkiauthenticate***name* 例： `Router(config)# crypto pki authenticate mytp`	CA 証明書を取得して、認証します。 • 証明書フィンガープリントをチェックするよう求められた場合、証明書フィンガープリントをチェックします。 （注） CA 証明書がコンフィギュレーションにすでにロードされている場合、このコマンドはオプションです。
ステップ 18	**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 19	**copysystem:running-confignvram:startup-config**
	
	例：
	
	`Router#`
	`copy system:running-config`
	`nvram:startup-config`
		（任意）実行コンフィギュレーションをNVRAMスタートアップ コンフィギュレーションにコピーします。
		
		（注） 実行コンフィギュレーションが変更されていてもNVRAMに書き込まれていない場合は、自動登録によって NVRAM が更新されません。

ステップ 20	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）ロールオーバー証明書などの、証明書に関する情報を表示します。

手動での証明書登録の設定

手動での証明書登録は、TFTP または手動でのカットアンドペースト方式によって設定できます。これらの方式は両方とも、CA が SCEP をサポートしない場合またはルータと CA 間のネットワーク接続が不可能な場合に使用できます。手動での証明書登録を設定するには、次のいずれかの作業を実行します。

証明書登録要求用の PEM 形式ファイル

証明書要求用の PEM 形式ファイルは、端末またはプロファイルベースの登録を使用して CA サーバから証明書を要求する場合に役立ちます。PEM 形式ファイルを使用すると、ルータで既存の証明書を直接使用できます。

手動での証明書登録に関する制約事項

SCEP の制約事項

SCEP が使用されている場合、URL を切り替えることは推奨しません。つまり、登録 URL が「http://myca」である場合、CA 証明書を取得した後と証明書を登録する前で、登録 URL を変更しないでください。ユーザは、TFTP と手動でのカットアンドペーストを切り替えることができます。

キー再生に関する制約事項

crypto key generate コマンドを使用して、キーを手動で再生しないでください。キーの再生は、**regenerate** キーワードを指定して **crypto pki enroll** コマンドを発行します。

カットアンドペーストによる証明書登録の設定

この作業は、カットアンドペーストによる証明書登録を設定するために実行します。PKI に参加しているピアに対してカットアンドペースト方式による手動での証明書登録を設定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint***name*
4. **enrollmentterminalpem**
5. **fingerprint***ca-fingerprint*
6. **exit**
7. **cryptopkiauthenticate***name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **showcryptopkicertificates**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptopkitrustpoint***name* 例： `Router(config)# crypto pki trustpoint mytp`	トラストポイントおよび設定された名前を宣言して、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	**enrollmentterminalpem** 例： `Router(ca-trustpoint)# enrollment terminal`	カットアンドペーストによる手動での証明書登録方式を指定します。 • 証明書要求は、手動でコピー（または切り取り）できるように、コンソール端末上に表示されます。 • **pem**：PEM 形式の証明書要求をコンソール端末に対して生成するようトラストポイントを設定します。

	コマンドまたはアクション	目的
ステップ 5	**fingerprint***ca-fingerprint* 例： `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	（任意）認証時に CA 証明書のフィンガープリントと照合するフィンガープリントを指定します。 （注） フィンガープリントが指定されていない場合は、フィンガープリントは検証用に表示されます。
ステップ 6	**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、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**cryptopkiauthenticate***name* 例： `Router(config)# crypto pki authenticate mytp`	CA 証明書を取得して、認証します。
ステップ 8	crypto pki enroll name 例： `Router(config)# crypto pki enroll mytp`	証明書要求を生成し、証明書サーバにコピーおよびペーストするために要求を表示します。 • 証明書要求にルータの FQDN および IP アドレスを含めるかどうかなどの登録情報を求められます。コンソール端末に対して証明書要求を表示するかについても選択できます。 • 必要に応じて、Base 64 符号化証明書を PEM ヘッダーを付けて、または付けずに表示します。
ステップ 9	crypto pki import name certificate 例： `Router(config)# crypto pki import mytp certificate`	コンソール端末で証明書を手動でインポートします（貼り付けます）。 • Base 64 符号化証明書はコンソール端末から受け取られ、内部証明書データベースに挿入されます。 （注） 用途キー、署名キー、および暗号キーを使用する場合は、このコマンドを 2 度入力する必要があります。このコマンドが初めて入力されたとき、証明書の 1 つがルータにペーストされます。このコマンドが 2 回目に入力されたとき、もう 1 つの証明書がルータにペーストされます。どちらの証明書が先にペーストされても問題ありません。 （注） 一部の CA は、証明書要求の用途キー情報を無視し、汎用目的の証明書を発行します。ご使用の認証局がこれに該当する場合は、汎用目的の証明書をインポートしてください。ルータは、生成される 2 つのキー ペアのいずれも使用しません。

	コマンドまたはアクション	目的
ステップ 10	**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 11	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）証明書、CA の証明書、および RA 証明書に関する情報を表示します。

TFTP による証明書登録の設定

この作業は、TFTP による証明書登録を設定するために実行します。この作業を実行すると、TFTP サーバを使用して手動で証明書登録を設定できます。

はじめる前に

- TFTP によって証明書登録を設定する場合は、使用する適切な URL がわかっている必要があります。
- ルータは、**crypto pki enroll** コマンドで TFTP サーバにファイルを書き込むことができる必要があります。
- ファイル指定と共に **enrollment** コマンドを使用する場合、ファイルには、バイナリ フォーマットまたは Base 64 符号化の CA 証明書が含まれている必要があります。
- ご使用の CA が証明書要求内のキーの用途情報を無視し、汎用目的の証明書だけを発行するかどうかを知っておく必要があります。

注意 一部の TFTP サーバでは、サーバが書き込み可能になる前に、ファイルがサーバ上に存在している必要があります。ほとんどの TFTP サーバでは、ファイルを上書きできる必要があります。任意のルータまたは他の装置によって証明書要求が書き込まれたり、上書きされることがあるため、この要件によって危険が生じる可能性があります。そのため、証明書要求を許可する前に、まず登録要求フィンガープリントをチェックする必要がある CA 管理者は交換証明書要求を使用しません。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint***name*
4. **enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**]
5. **fingerprint***ca-fingerprint*
6. **exit**
7. **cryptopkiauthenticate***name*
8. crypto pki enroll name
9. crypto pki import name certificate
10. **exit**
11. **showcryptopkicertificates**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptopkitrustpoint***name* 例： `Router(config)# crypto pki trustpoint mytp`	トラストポイントおよび設定された名前を宣言して、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	**enrollment** [**mode**] [**retry period** minutes] [**retry count** number] **url** url [**pem**] 例： `Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification`	登録要求を送信して、CA 証明書とルータ証明書および任意のオプションのパラメータを取得するための登録方式として TFTP を指定します。 （注） TFTP 登録の場合、URL は TFTP URL（tftp://example_tftp_url）として設定する必要があります。 • TFTP URL には、任意のファイル指定ファイル名を使用できます。ファイル指定が含まれていない場合は、FQDN が使用されます。ファイル指定が含まれている場合は、ルー

	コマンドまたはアクション	目的
		タは指定されたファイル名に「.ca」という拡張子を付加します。
ステップ 5	**fingerprint***ca-fingerprint* 例： `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	（任意）CA 管理者からアウトオブバンド方式によって受け取る CA 証明書のフィンガープリントを指定します。 （注） フィンガープリントが指定されていない場合は、フィンガープリントは検証用に表示されます。
ステップ 6	**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、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**cryptopkiauthenticate***name* 例： `Router(config)# crypto pki authenticate mytp`	指定された TFTP サーバから CA 証明書を取得して認証します。
ステップ 8	crypto pki enroll name 例： `Router(config)# crypto pki enroll mytp`	証明書要求を生成し、この要求を TFTP サーバに書き込みます。 • 証明書要求にルータの FQDN および IP アドレスを含めるかどうかなどの登録情報を求められます。コンソール端末に証明書要求を表示するかどうかについて尋ねられます。 • 書き込まれるファイル名には「.req」という拡張子が付加されます。用途キー、署名キー、および暗号キーの場合、2 つの要求が生成されて送信されます。用途キーの要求ファイル名には、拡張子「-sign.req」および「-encr.req」がそれぞれ付加されます。
ステップ 9	crypto pki import name certificate 例： `Router(config)# crypto pki import mytp certificate`	許可された証明書を取得するコンソール端末で、TFTP によって証明書をインポートします。 • ルータは、拡張子が「.req」から「.crt」に変更されたことを除いて、要求の送信に使用した同じファイル名を使用して、許可された証明書を TFTP によって取得しようと試みます。用途キー証明書の場合、拡張子「-sign.crt」および「-encr.crt」が使用されます。 • ルータは、受信したファイルを解析して証明書を検証し、証明書をルータの内部証明書データベースに挿入します。

	コマンドまたはアクション	目的
		（注） 一部の CA は、証明書要求の用途キー情報を無視し、汎用目的の証明書を発行します。ご使用の CA が証明書要求の用途キー情報を無視する場合は、汎用目的の証明書だけをインポートしてください。ルータは、生成される 2 つのキー ペアのいずれも使用しません。
ステップ 10	**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 11	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）証明書、CA の証明書、および RA 証明書に関する情報を表示します。

Trend Micro サーバとセキュアな通信を行うための URL リンクの認証

この作業は、Trend Micro サーバとセキュアに通信できるようにする URL フィルタリングで使用されるリンクを認証するために実行します。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

手順の概要

1. **enable**
2. **clockset***hh*:*mm*:*ssdatemonthyear*
3. **configureterminal**
4. **clocktimezone***zone hours-offset* [*minutes-offset*]
5. **iphttpserver**
6. **hostname***name*
7. **ipdomain-name***name*
8. **cryptokeygeneratersageneral-keysmodulus***modulus-size*
9. **cryptopkitrustpoint***name*
10. **enrollmentterminal**
11. **cryptocaauthenticate***name*
12. Base 64 符号化の CA 証明書が含まれている次のテキスト部分をコピーし、プロンプトにペーストします。
13. **yes** と入力し、この証明書を受け入れます。
14. **serial-number**
15. **revocation-checknone**
16. **end**
17. **trmregister**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**clockset***hh*:*mm*:*ssdatemonthyear* 例： `Router# clock set 23:22:00 22 Dec 2009`	ルータのクロックを設定します。
ステップ 3	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 4	**clocktimezone***zone hours-offset* [*minutes-offset*] 例： `Router(config)# clock timezone PST -08`	時間帯を設定します。 • *zone* 引数は、時間帯の名前です（通常は標準略語）。*hours-offset* 引数は、使用する時間帯が協定世界時（UTC）から異なる時間数です。*minutes-offset* 引数は、使用する時間帯が UTC から異なる分数です。 （注） **clocktimezone** コマンドの *minutes-offset* 引数は、ローカル時間帯の UTC またはグリニッジ標準時（GMT）からの差が 1 時間未満の割合で異なる場合に使用できます。たとえば、アトランティック カナダの一部の地域の時間帯（大西洋標準時（AST））は UTC-3.5 です。この場合、必要なコマンドは **clocktimezoneAST-330** となります。
ステップ 5	**iphttpserver** 例： `Router(config)# ip http server`	HTTP サーバをイネーブルにします。
ステップ 6	**hostname***name* 例： `Router(config)# hostname hostname1`	ルータのホスト名を設定します。
ステップ 7	**ipdomain-name***name* 例： `Router(config)# ip domain-name example.com`	ルータのドメイン名を定義します。
ステップ 8	**cryptokeygeneratersageneral-keysmodulus***modulus-size* 例： `Router(config)# crypto key generate rsa general-keys modulus general`	暗号キーを生成します。 • **general-keys** キーワードは、汎用のキー ペアが生成されることを指定します。これがデフォルトです。 • **modulus** キーワードと *modulus-size* 引数は、キーのモジュラスの IP サイズを指定します。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。RSA キーを生成する場合、モジュラスの長さを入力するように促されます。モジュラスの長さが長いほど安全性が高まりますが、生成と使用にかかる時間も長くなります。2048 未満の長さを指定することは推奨されません。 （注） 生成される汎用キーの名前は、手順 7 で設定したドメイン名に基づきます。たとえば、キーの名前は「example.com」になります。

	コマンドまたはアクション	目的
ステップ 9	**cryptopkitrustpoint***name* 例： `Router(config)# crypto pki trustpoint mytp`	ルータが使用する CA を宣言し、CA トラストポイント コンフィギレーション モードを開始します。 （注） Cisco IOS Release 12.3(8)T で有効です。**cryptocatrustpoin** コマンドは **cryptopkitrustpoint** コマンドに置き換えられ した。
ステップ 10	**enrollmentterminal** 例： `Router(ca-trustpoint)# enrollment terminal`	カットアンドペーストによる手動での証明書登録方式を指定します • 証明書要求は、手動でコピー（または切り取り）できるように コンソール端末上に表示されます。

ステップ 11	**cryptocaauthenticate***name*
	
	例：
	
	`Router(ca-trustpoint)# crypto ca authenticate mytp`
		CA の名前を引数として取得し、これを認証します。
		
		• 次のコマンドの出力が表示されます。
		
		`Enter the base 64 encoded CA certificate.`
		`End with a blank line or the word "quit" on a line by itself.`

	コマンドまたはアクション	目的

ステップ 12	Base 64 符号化の CA 証明書が含まれている次のテキスト部分をコピーし、プロンプトにペーストします。	MIIDIDCCAomgAwIBAgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJV
		UzEQMA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBTZWN1cmUgQ2Vy
		dGlmaWNhdGUgQXV0aG9yaXR5MB4XDTk4MDgyMjE2NDE1MVoXDTE4MDgyMjE2NDE1
		MVowTjELMAkGA1UEBhMCVVMxEDAOBgNVBAoTB0VxdWlmYXgxLTArBgNVBAsTJEVx
		dWlmYXggU2VjdXJlIENlcnRpZmljYXRlIEF1dGhvcml0eTCBnzANBgkqhkiG9w0B
		AQEFAAOBjQAwgYkCgYEAwV2xWGcIYu6gmi0fCG2RFGiYCh7+2gRvE4RiIcPRfM6f
		BeC4AfBONOziipUEZKzxa1NfBbPLZ4C/QgKO/t0BCezhABRP/PvwDN1Dulsr4R+A
		cJkVV5MW8Q+XarfCaCMczE1ZMKxRHjuvK9buY0V7xdlfUNLjUA86iOe/FP3gx7kC
		AwEAAaOCAQkwggEFMHAGA1UdHwRpMGcwZaBjoGGkXzBdMQswCQYDVQQGEwJVUzEQ
		MA4GA1UEChMHRXF1aWZheDEtMCsGA1UECxMkRXF1aWZheCBTZWN1cmUgQ2VydGlm
		aWNhdGUgQXV0aG9yaXR5MQ0wCwYDVQQDEwRDUkwxMBoGA1UdEAQTMBGBDzIwMTgw
		ODIyMTY0MTUxWjALBgNVHQ8EBAMCAQYwHwYDVR0jBBgwFoAUSOZo+SvSspXXR9gj
		IBBPM5iQn9QwHQYDVR0OBBYEFEjmaPkr0rKV10fYIyAQTzOYkJ/UMAwGA1UdEwQF
		MAMBAf8wGgYJKoZIhvZ9B0EABA0wCxsFVjMuMGMDAgbAMA0GCSqGSIb3DQEBBQUA
		A4GBAFjOKer89961zgK5F7WF0bnj4JXMJTENAKaSbn+2kmOeUJXRmm/kEd5jhW6Y
		7qj/WsjTVbJmcVfewCHrPSqnI0kBBIZCe/zuf6IWUrVnZ9NA2zsmWLIodz2uFHdh
		1voqZiegDfqnc1zqcPGUIWVEX/r87yloqaKHee9570+sB3c4
		次のコマンドの出力が表示されます。
		`Certificate has the following attributes:`
		`Fingerprint MD5: 67CB9DC0 13248A82 9BB2171E D11BECD4`
		`Fingerprint SHA1: D23209AD 23D31423 2174E40D 7F9D6213 9786633A`

ステップ 13	**yes** と入力し、この証明書を受け入れます。	`% Do you accept this certificate? [yes/no]: yes`
		次のコマンドの出力が表示されます。
		`Trustpoint CA certificate accepted.`
		`% Certificate successfully imported`

	コマンドまたはアクション	目的
ステップ 14	**serial-number** 例： `hostname1(ca-trustpoint)# serial-number`	ルータのシリアル番号を証明書要求で指定します。
ステップ 15	**revocation-checknone** 例： `hostname1(ca-trustpoint)# revocation-check none` 例：	証明書の確認が無視されることを指定します。
ステップ 16	**end** 例： `hostname1(ca-trustpoint)# end`	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、特権 EXEC モードに戻ります。
ステップ 17	**trmregister** 例： `hostname1# trm register`	Trend Micro サーバ登録プロセスを手動で開始します。

登録用の永続的自己署名証明書の SSL による設定

ここでは、次のタスクについて説明します。

（注） これらの作業は任意です。これは、HTTPS サーバをイネーブルにした場合、このサーバがデフォルト値を使用して自動的に自己署名証明書を生成するからです。

永続的自己署名証明書の概要

SSL プロトコルは、HTTPS サーバとクライアント（Web ブラウザ）の間でセキュアな接続を確立するために使用されます。SSL ハンドシェイクの間、クライアントは、すでに所有している証明書を使用して SSL サーバの証明書が検証可能であると想定します。

Cisco IOS ソフトウェアが HTTP サーバで使用できる証明書を保持していない場合、サーバは、PKI アプリケーション プログラミング インターフェイス（API）を呼び出して自己署名証明書を生成

します。クライアントがこの自己署名証明書を受け取ったにもかかわらず、検証できない場合、ユーザによる介入が必要です。クライアントは、証明書を受け入れるか、あとで使用するために保存するかどうかを尋ねます。証明書を受け入れると、SSL ハンドシェイクは続行されます。

それ以降、同じクライアントとサーバ間の SSL ハンドシェイクでは、同じ証明書が使用されます。ただし、ルータをリロードすると、自己署名証明書は失われます。その場合、HTTPS サーバは新しい自己署名証明書を作成する必要があります。この新しい自己署名証明書は前の証明書と一致しないため、この自己署名証明書を受け入れるかどうか再度確認されます。

ルータがリロードするたびにルータの証明書を受け入れるかどうか確認されると、この確認中に、攻撃者に不正な証明書を使用する機会を与えてしまうことがあります。永続的自己署名証明書では、ルータのスタートアップ コンフィギュレーションに証明書を保存することにより、これらの制約をすべて解消しています。

機能制限

1 つの永続的自己署名証明書には、トラストポイントを 1 つだけ設定できます。

（注） 自己署名証明書の作成後は、ルータの IP ドメイン名またはホスト名を変更しないでください。いずれかの名前を変更すると、自己署名証明書の再生がトリガーされて、設定済みのトラストポイントが上書きされます。WebVPN は、SSL トラストポイント名を WebVPN ゲートウェイ設定に結合します。新しい自己署名証明書がトリガーされると、新しいトラストポイント名が WebVPN 設定と一致しなくなり、WebVPN 接続は失敗します。

トラストポイントの設定および自己署名証明書パラメータの指定

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

トラストポイントを設定し、自己署名証明書パラメータを指定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint**name
4. **enrollmentselfsigned**
5. **subject-name** [*x.500-name*]
6. **rsakeypair***key-label* [key-size [encryption-key-size]]
7. **cryptopkienroll**name
8. **end**
9. **showcryptopkicertificates**[*trustpoint-name*[**verbose**]]
10. **showcryptopkitrustpoints**[**status** | *label* [**status**]]

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**cryptopkitrustpoint**name 例： `Router(config)# crypto pki trustpoint local`	ルータが使用する CA を宣言し、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。 （注） Cisco IOS Release 12.3(8)T で有効です。**cryptocatrustpoint** コマンドは **cryptopkitrustpoint** コマンドに置き換えられました。
ステップ 4	**enrollmentselfsigned** 例： `Router(ca-trustpoint)# enrollment selfsigned`	自己署名登録を指定します。
ステップ 5	**subject-name** [*x.500-name*] 例： `Router(ca-trustpoint)# subject-name`	（任意）証明書要求に使用する要求件名を指定します。 • *x-500-name* 引数を指定しない場合、デフォルト件名である FQDN が使用されます。

	コマンドまたはアクション	目的
ステップ 6	**rsakeypair***key-label* [key-size [encryption-key-size]] 例： `Router(ca-trustpoint)# rsakeypair examplekey 2048`	（任意）証明書に関連付けるキーペアを指定します。 • *key-label* 引数の値がまだ存在しない、あるいは **auto-enrollregenerate** コマンドが発行された場合は、登録時にこの引数の値が生成されます。 • キーを生成するための *key-size* 引数を指定し、*encryption-key-size* 引数を指定して、個別の暗号化、署名キー、および証明書を要求します。key-size と encryption-key-size は同じサイズでなければなりません。2048 未満の長さを指定することは推奨されません。 （注） このコマンドがイネーブルでない場合に、FQDN キー ペアが使用されます。
ステップ 7	**cryptopkienroll**name 例： `Router(ca-trustpoint)# crypto pki enroll local`	永続的自己署名証明書を生成するようルータに指示します。
ステップ 8	**end** 例： `Router(ca-trustpoint)# end`	（任意）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。 • グローバル コンフィギュレーション モードを終了するため、このコマンドをもう一度入力します。
ステップ 9	**showcryptopkicertificates**[*trustpoint-name*[**verbose**]] 例： `Router# show crypto pki certificates local verbose`	証明書、認証局証明書、および任意の登録認局証明書に関する情報を表示します。
ステップ 10	**showcryptopkitrustpoints**[**status** | *label* [**status**]] 例： `Router# show crypto pki trustpoints status`	ルータに設定されているトラストポイントを表示します。

HTTPS サーバのイネーブル化

HTTPS サーバをイネーブルにするには、次の作業を実行します。

はじめる前に

パラメータを指定するには、トラストポイントを作成し、設定する必要があります。デフォルト値を使用するには、すべての既存の自己署名トラストポイントを削除します。自己署名トラストポイントをすべて削除すると、HTTPS サーバがイネーブルになるとただちに、サーバはデフォルト値を使用して永続的自己署名証明書を生成します。

手順の概要

1. **enable**
2. **configureterminal**
3. **iphttpsecure-server**
4. **end**
5. **copysystem:running-confignvram:startup-config**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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	**iphttpsecure-server** 例： `Router(config)# ip http secure-server`	HTTPS Web サーバをイネーブルにします。 （注） キー ペア（Modulus 1024）および自己署名証明書が自動的に生成されます。
ステップ 4	**end** 例： `Router(config)# end`	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	**copysystem:running-confignvram:startup-config** 例： `Router# copy system:running-config nvram: startup-config`	イネーブルになっているモードで自己署名証明書および HTTPS サーバを保存します。

登録または再登録用の証明書登録プロファイルの設定

この作業は、登録または再登録用の証明書登録プロファイルを設定するために実行します。この作業は、サードパーティ ベンダー製 CA にすでに登録されている証明書またはルータを Cisco IOS CA に登録または再登録するための登録プロファイルを設定するのに役立ちます。

登録要求が自動的に許可されるように、サードパーティ ベンダー製 CA に登録されているルータを Cisco IOS 証明書サーバに登録するには、このルータをイネーブルにして、その既存の証明書を使用します。この機能をイネーブルにするには、**enrollmentcredential** コマンドを発行する必要があります。また、手動による証明書登録は設定できません。

はじめる前に

次の作業は、サードパーティ ベンダー製 CA にすでに登録されているクライアント ルータの証明書登録プロファイルを設定する前に、クライアント ルータで実行します。これにより、そのルータを Cisco IOS 証明書サーバに再登録できます。

- サードパーティ ベンダー製 CA をポイントするトラストポイントの定義
- サードパーティ ベンダー製 CA でのクライアント ルータの認証および登録

（注）

- 証明書プロファイルを使用するには、ネットワークに、CA への HTTP インターフェイスが設定されている必要があります。
- 登録プロファイルが指定されている場合、トラストポイント設定に登録 URL が指定されていないことがあります。両方のコマンドがサポートされていても、トラストポイントに使用できるコマンドは一度に 1 つだけです。
- 各 CA で使用される HTTP コマンドには規格がないため、ユーザは使用している CA に適したコマンドを入力する必要があります。

>

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint***name*
4. enrollment profile label
5. **exit**
6. **cryptopkiprofileenrollment***label*
7. 次のいずれかを実行します。
   - **authenticationurl***url*
   - **authenticationterminal**
8. **authenticationcommand**
9. 次のいずれかを実行します。
   - **enrollmenturl***url*
   -
   - **enrollmentterminal**
10. **enrollmentcredential***label*
11. **enrollmentcommand**
12. **parameter***number* {**value***value* | **prompt***string*}
13. **exit**
14. **showcryptopkicertificates**

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 3	**cryptopkitrustpoint***name* 例： `Router(config)# crypto pki trustpoint Entrust`	トラストポイントおよび設定された名前を宣言して、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	enrollment profile label 例： `Router(ca-trustpoint)# enrollment profile E`	登録プロファイルが証明書認証および登録用に使用されるように指定します。
ステップ 5	**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 6	**cryptopkiprofileenrollment***label* 例： `Router(config)# crypto pki profile enrollment E`	登録プロファイルを定義し、ca-profile-enroll コンフィギュレーション モードを開始します。 • *label*：登録プロファイルの名前。登録プロファイル名は、**enrollmentprofile** コマンドで指定された名前と同じである必要があります。

ステップ 7	次のいずれかを実行します。
	
	• **authenticationurl***url*
	• **authenticationterminal**
	
	例：
	
	`Router(ca-profile-enroll)# authentication url http://entrust:81`
	
	例：
	
	`Router(ca-profile-enroll)# authentication terminal`
		証明書認証要求の送信先となる CA サーバの URL を指定します。
		
		• *url*：ルータが認証要求を送信する CA サーバの URL。HTTP を使用する場合、URL は「http://CA_name」という形式にする必要があります。ここで、CA_name は CA のホスト DNS 名または IP アドレスです。TFTP を使用する場合、この URL は「tftp://certserver/file_specification」という形式にする必要があります。（URL にファイル指定が含まれない場合、ルータの FQDN が使用されます。）
		
		カットアンドペーストによる手動での証明書認証を指定します。

ステップ 8	**authenticationcommand** 例： `Router(ca-profile-enroll)# authentication command`	（任意）認証のために CA に送信される HTTP コマンドを指定します。

	コマンドまたはアクション	目的

ステップ 9	次のいずれかを実行します。
	• **enrollmenturl***url*
	•
	• **enrollmentterminal**
	
	例：
	`Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe`
	
	例：
	
	例：
	`Router(ca-profile-enroll)# enrollment terminal`
		証明書登録要求を HTTP または TFTP によって送信する CA サーバの URL を指定します。
		カットアンドペーストによる手動での証明書登録を指定します。

ステップ 10	**enrollmentcredential***label* 例： `Router(ca-profile-enroll)# enrollment credential Entrust`	（任意）Cisco IOS CA に登録されるサードパーティ ベンダー製 CA トラストポイントを指定します。 （注） 手動での証明書登録が使用されている場合、このコマンドは発行できません。
ステップ 11	**enrollmentcommand** 例： `Router(ca-profile-enroll)# enrollment command`	（任意）登録のために CA に送信される HTTP コマンドを指定します。
ステップ 12	**parameter***number* {**value***value* | **prompt***string*} 例： `Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc`	（任意）登録プロファイルのパラメータを指定します。 • このコマンドを繰り返して使用すると、複数の値を指定できます。
ステップ 13	**exit** 例： `Router(ca-profile-enroll)# exit`	（任意）ca-profile-enroll コンフィギュレーション モードを終了します。 • グローバル コンフィギュレーション モードを終了するため、このコマンドをもう一度入力します。

	コマンドまたはアクション	目的
ステップ 14	**showcryptopkicertificates** 例： `Router# show crypto pki certificates`	（任意）証明書、CA の証明書、および RA 証明書に関する情報を表示します。

次の作業

Cisco IOS CA に再登録するようにルータを設定した場合にこの機能を活用するには、指定されたサードパーティ ベンダー製 CA トラストポイントに登録されたクライアントからだけ登録要求を受け入れるように Cisco IOS 証明書サーバを設定する必要があります。詳細については、「Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment」を参照してください。

PKI 証明書登録要求の設定例

証明書登録または自動登録の設定例

次の例では、「mytp-A」証明書サーバおよび関連付けられたトラストポイントの設定を示します。この例では、トラストポイントの初期の自動登録によって生成された RSA キーが USB トークン「usbtoken0」に保管されます。

```
crypto pki server mytp-A
   database level complete
   issuer-name CN=company, L=city, C=country
   grant auto
! Specifies that certificate requests will be granted automatically.
!
crypto pki trustpoint mytp-A
   revocation-check none
   rsakeypair myTP-A
   storage usbtoken0:
! Specifies that keys will be stored on usbtoken0:.
   on usbtoken0:
```

! Specifies that keys generated on initial auto enroll will be generated on and stored o**n !** usbtoken0:

自動登録の設定例

次の例では、自動ロールオーバーをイネーブルにして、ルータが起動時に自動的に CA に登録されるように設定する方法、および必要なすべての登録情報を設定に指定する方法を示します。

```
crypto pki trustpoint trustpt1
 enrollment url http://trustpt1.example.com//
 subject-name OU=Spiral Dept., O=example.com
```

```
 ip-address ethernet-0
 serial-number none
 usage ike
 auto-enroll regenerate
 password password1
 rsa-key trustpt1 2048
!
crypto pki certificate chain trustpt1
certificate pki 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

（注） この例では、キーは再生もロールオーバーもされません。

証明書自動登録とキー再生の設定例

次の例では、ルータが起動時に「trustme1」という CA に自動的に登録され、自動ロールオーバーがイネーブルになるように設定する方法を示します。**regenerate** キーワードが発行されるため、自動ロールオーバー プロセスが開始されると、新しいキーが証明書に対して生成され、再発行されます。更新パーセンテージが 90 に設定されているため、証明書の有効期間が 1 年の場合は、古い証明書が失効する 36.5 日前に新しい証明書が要求されます。実行コンフィギュレーションを変更しても、NVRAM に書き込まないかぎり自動登録によって NVRAM が更新されないため、実行コンフィギュレーションの変更はNVRAM スタートアップ コンフィギュレーションに保存されます。

```
crypto pki trustpoint trustme1
 enrollment url http://trustme1.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 auto-enroll 90 regenerate
 password password1
 rsakeypair trustme1 2048
 exit
crypto pki authenticate trustme1
copy system:running-config nvram:startup-config
```

カットアンドペーストによる証明書登録の設定例

次の例では、カットアンドペーストによる手動での登録方式を使用して、証明書登録を設定する方法を示します。

```
Router(config)#
crypto pki trustpoint TP
Router(ca-trustpoint)#
enrollment terminal
Router(ca-trustpoint)#
crypto pki authenticate TP
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
-----BEGIN CERTIFICATE-----
MIICNDCCAd6gAwIBAgIQOsCmXpVHwodKryRoqULV7jANBgkqhkiG9w0BAQUFADA5
MQswCQYDVQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJ
bXNjYS1yb290MB4XDTAyMDIxNDAwNDYwMVoXDTA3MDIxNDAwNTQ0OFowOTELMAkG
A1UEBhMCVVMxFjAUBgNVBAoTDUNpc2NvIFN5c3RlbXMxEjAQBgNVBAMTCW1zY2Et
cm9vdDBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCix8nIGFg+wvy3BjFbVi25wYoG
K2N0HWWHpqxFuFhqyBnIC0OshIn9CtrdN3JvUNHr0NIKocEwNKUGYmPwWGTfAgMB
AAGjgcEwgb4wCwYDVR0PBAQDAgHGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYE
FKIacsl6dKAfuNDVQymlSp7esf8jMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9t
c2NhLXJvb3QvQ2VydEVucm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8v
XFxtc2NhLXJvb3RcQ2VydEVucm9sbFxtc2NhLXJvb3QuY3JsMBAGCSsGAQQBgjcV
AQQDAgEAMA0GCSqGSIb3DQEBBQUAA0EAeuZkZMX9qkoLHfETYTpVWjZPQbBmwNRA
oJDSdYdtL3BcI/uLL5q7EmODyGfLyMGxuhQYx5r/40aSQgLCqBq+yg==
-----END CERTIFICATE-----
Certificate has the following attributes:
Fingerprint: D6C12961 CD78808A 4E02193C 0790082A
% Do you accept this certificate? [yes/no]:
y
Trustpoint CA certificate accepted.
% Certificate successfully imported
Router(config)#
crypto pki enroll TP
% Start certificate enrollment..
% The subject name in the certificate will be:
Router.example.com
% Include the router serial number in the subject name? [yes/no]:
n
% Include an IP address in the subject name? [no]:
n
Display Certificate Request to terminal? [yes/no]:
y
Signature key certificate request -
Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxdhXFDiWAn/hIZs9zfOtssKA
daoWYu0ms9Fe/Pew01dh14vXdxgacstOs2Pr5wk6jLOPxpvxOJPWyQM6ipLmyVxv
ojhyLTrVohrh6Dnqcvk+G/5ohss9o9RxvONwx042pQchFnx9EkMuZC7evwRxJEqR
mBHXBZ8GmP3jYQsjS8MCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgeAMA0GCSqGSIb3DQEBBAUAA4GBAMT6WtyFw95POY7UtF+YIYHiVRUf4SCq
hRIAGrljUePLo9iTqyPU1Pnt8JnIZ5P5BHU3MfgP8sqodaWub6mubkzaohJ1qD06
O87fnLCNid5Tov5jKogFHIki2EGGZxBosUw9lJlenQdNdDPbJc5LIWdfDvciA6jO
Nl8rOtKnt8Q+
!
!
!
Redisplay enrollment request? [yes/no]:
Encryption key certificate request -
Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTYW5kQmFnZ2VyLmNpc2NvLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwG60QojpDbzbKnyj8FyTiOcv
THkDP7XD4vLT1XaJ409z0gSIoGnIcdFtXhVlBWtpq3/O9zYFXr1tH+BMCRQi3Lts
0IpxYa3D9iFPqev7SPXpsAIsY8a6FMq7TiwLObqiQjLKL4cbuV0Frjl0Yuv5A/Z+
kqMOm7c+pWNWFdLe9lsCAwEAAaAhMB8GCSqGSIb3DQEJDjESMBAwDgYDVR0PAQH/
BAQDAgUgMA0GCSqGSIb3DQEBBAUAA4GBACF7feURj/fJMojPBlR6fa9BrlMJx+2F
H91YM/CIiz2n4mHTeWTWKhLoT8wUfa9NGOk7yi+nF/F7035twLfq6n2bSCTW4aem
```

```
8jLMMaeFxwkrV/ceQKrucmNC1uVx+fBy9rhnKx8j60XE25tnp1U08r6om/pBQABU
eNPFhozcaQ/2
!
!
!
Redisplay enrollment request? [yes/no]:
n
Router(config)#
crypto pki import TP certificate
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0MloXDTAzMDYwODAxMjY0MlowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMXYVxQ4lgJ/4SGbPc3zrbLCgHWqFmLtJrPRXvz3sNNXYdeL13cYGnLL
TrNj6+cJOoyzj8ab8TiT1skDOoqS5slcb6I4ci061aIa4eg56nL5Phv+aIbLPaPU
cbzjcMdONqUHIRZ8fRJDLmQu3r8EcSRKkZgR1wWfBpj942ELI0vDAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFL8Quz8dyz4EGIeKx9A8UMNHLE4s
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAJo2
r6sHPGBdTQX2EDoJpR/A2UHXxRYqVSHkFKZw0z31r5JzUM0oPNUETV7mnZlYNVRZ
CSEX/G8boi3WOjz9wZo=
% Router Certificate successfully imported
Router(config)#
crypto pki import TP cert
Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
MIIDajCCAxSgAwIBAgIKFN7OBQAAAAAMSDANBgkqhkiG9w0BAQUFADA5MQswCQYD
VQQGEwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1y
b290MB4XDTAyMDYwODAxMTY0NVoXDTAzMDYwODAxMjY0NVowJTEjMCEGCSqGSIb3
DQEJAhMUU2FuZEJhZ2dlci5jaXNjby5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0A
MIGJAoGBAMButEKI6Q282yp8o/Bck4jnL0x5Az+1w+Ly09V2ieNPc9IEiKBpyHHR
bV4VZQVraat/zvc2BV69bR/gTAkUIty7bNCKcWGtw/YhT6nr+0j16bACLGPGuhTK
u04sCzm6okIyyi+HG7ldBa45dGLr+QP2fpKjDpu3PqVjVhXS3vZbAgMBAAGjggHM
MIIByDALBgNVHQ8EBAMCBSAwHQYDVR0OBBYEFPDO29oRdlEUSgBMg6jZR+YFRWlj
MHAGA1UdIwRpMGeAFKIacsl6dKAfuNDVQymlSp7esf8joT2kOzA5MQswCQYDVQQG
EwJVUzEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczESMBAGA1UEAxMJbXNjYS1yb290
ghA6wKZelUfCh0qvJGipQtXuMCIGA1UdEQEB/wQYMBaCFFNhbmRCYWdnZXIuY2lz
Y28uY29tMG0GA1UdHwRmMGQwL6AtoCuGKWh0dHA6Ly9tc2NhLXJvb3QvQ2VydEVu
cm9sbC9tc2NhLXJvb3QuY3JsMDGgL6AthitmaWxlOi8vXFxtc2NhLXJvb3RcQ2Vy
dEVucm9sbFxtc2NhLXJvb3QuY3JsMIGUBggrBgEFBQcBAQSBhzCBhDA/BggrBgEF
BQcwAoYzaHR0cDovL21zY2Etcm9vdC9DZXJ0RW5yb2xsL21zY2Etcm9vdF9tc2Nh
LXJvb3QuY3J0MEEGCCsGAQUFBzAChjVmaWxlOi8vXFxtc2NhLXJvb3RcQ2VydEVu
cm9sbFxtc2NhLXJvb3RfbXNjYS1yb290LmNydDANBgkqhkiG9w0BAQUFAANBAHaU
hyCwLirUghNxCmLzXRG7C3W1j0kSX7a4fX9OxKR/Z2SoMjdMNPPyApuh8SoT2zBP
ZKjZU2WjcZG/nZF4W5k=
% Router Certificate successfully imported
```

証明書が正常にインポートされたかどうかを確認するには、**show crypto pki certificates** コマンドを発行します。

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 14DECE05000000000C48
  Certificate Usage: Encryption
  Issuer:
    CN = TPCA-root
     O = Company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
```

```
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:45 PDT Jun 7 2002
    end   date: 18:26:45 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP
Certificate
  Status: Available
  Certificate Serial Number: 14DEC2E9000000000C47
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = company
     C = US
  Subject:
    Name: Router.example.com
    OID.1.2.840.113549.1.9.2 = Router.example.com
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 18:16:42 PDT Jun 7 2002
    end   date: 18:26:42 PDT Jun 7 2003
    renew date: 16:00:00 PST Dec 31 1969
  Associated Trustpoints: TP
CA Certificate
  Status: Available
  Certificate Serial Number: 3AC0A65E9547C2874AAF2468A942D5EE
  Certificate Usage: Signature
  Issuer:
    CN = tpca-root
     O = Company
     C = US
  Subject:
    CN = tpca-root
     O = company
     C = US
  CRL Distribution Point:
    http://tpca-root/CertEnroll/tpca-root.crl
  Validity Date:
    start date: 16:46:01 PST Feb 13 2002
    end   date: 16:54:48 PST Feb 13 2007
  Associated Trustpoints: TP
```

キー再生を使用した手動での証明書登録の設定例

次の例では、「trustme2」という名前の CA から手動で証明書を登録して、新しいキーを再生する方法を示します。

```
crypto pki trustpoint trustme2
 enrollment url http://trustme2.example.com/
 subject-name OU=Spiral Dept., O=example.com
 ip-address ethernet0
 serial-number none
 regenerate
 password password1
 rsakeypair trustme2 2048
 exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
```

永続的自己署名の証明書の作成および検証例

次の例では、「local」という名前のトラストポイントを宣言して登録し、IP アドレスを含む自己署名証明書を生成する方法を示します。

```
crypto pki trustpoint local
 enrollment selfsigned
 end
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

（注） ルータに設定できる自己署名証明書は 1 つだけです。自己署名証明書がすでに存在する場合に、別の自己署名証明書用に設定されたトラストポイントを登録しようとすると、通知が表示され、自己署名証明書を置き換えるかどうか尋ねられます。置き換える場合は、新しい自己署名証明書が生成され、既存の自己署名証明書と置き換えられます。

HTTPS サーバのイネーブル化の例

次の例では、以前に HTTPS サーバが設定されていなかったため、HTTPS サーバをイネーブルにし、デフォルト トラストポイントを生成する方法を示します。

```
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ip http secure-server
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified.  Issue "write memory"
to save new certificate
Router(config)#
```

（注） 自己署名証明書を保持し、次にルータをリロードしたときに HTTPS サーバをイネーブルにする場合は、コンフィギュレーションを NVRAM に保存する必要があります。

次のメッセージも表示されます。

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled
```

（注） 自己署名証明書で使用されたキー ペアを作成すると、Secure Shell（SSH）サーバが起動します。この動作は抑制できません。ご使用のアクセス コントロール リスト（ACL）を変更して、ルータへの SSH アクセスを許可または拒否できます。**ip ssh rsa keypair-name** *unexisting-key-pair-name* コマンドを使用し、SSH サーバをディセーブルにできます。

自己署名証明書設定の検証例

次の例では、作成した自己署名証明書に関する情報を表示します。

```
Router# show crypto pki certificates
Router Self-Signed Certificate
  Status: Available
  Certificate Serial Number: 01
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
  Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
  Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
  Associated Trustpoints: TP-self-signed-3326000105
```

（注） 上記の 3326000105 という数値はルータのシリアル番号で、これはルータの実際のシリアル番号によって異なります。

次の例では、自己署名証明書に対応するキー ペアに関する情報を表示します。

```
Router# show crypto key mypubkey rsa
% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
  6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
  BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
  6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
  2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
  463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
  8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
  34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
```

（注） TP-self-signed-3326000105.server という 2 番目のキー ペアは、SSH キー ペアです。ルータに任意のキー ペアが作成されて SSH が起動すると、生成されます。

次の例では、「local」というトラストポイントに関する情報を表示します。

```
Router# show crypto pki trustpoints
Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.example.com
          Serial Number: 01
    Persistent self-signed certificate trust point
```

HTTP による直接登録の設定例

次の例では、HTTP による CA サーバへの直接登録ための登録プロファイルを設定する方法を示します。

```
crypto pki trustpoint Entrust
 enrollment profile E
 serial
crypto pki profile enrollment E
 authentication url http://entrust:81
 authentication command GET /certs/cacert.der
 enrollment url http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command POST reference_number=$P2&authcode=$P1
 &retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
USB トークンによる RSA 処理：USB トークンを使用するメリット	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Storing PKI Credentials」モジュール
USB トークンによる RSA 処理：証明書サーバの設定	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment」項 「Generating a Certificate Server RSA Key Pair」項、「Configuring a Certificate Server Trustpoint」項、および関連する例を参照してください。
PKI の概要（RSA キー、証明書登録、および CA を含む）	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Cisco IOS PKI Overview: Understanding and Planning a PKI」モジュール
安全なデバイスプロビジョニング：機能概要および設定作業	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI」項
RSA キーの生成および展開	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Deploying RSA Keys Within a PKI」モジュール

関連項目	マニュアル タイトル
Cisco IOS 証明書サーバの概要および設定作業	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment」モジュール
USB トークンの設定および使用	『Cisco IOS Security Configuration Guide: Secure Connectivity』の「Storing PKI Credentials」モジュール
Cisco IOS セキュリティ コマンド	『*Cisco IOS Security Command Reference*』
Suite-B の ESP トランスフォーム	『Configuring Security for VPNs with IPsec』フィーチャ モジュール
Suite-B SHA-2 ファミリ（HMAC バリアント）および Elliptic Curve（EC）キー ペアの設定。	『Configuring Internet Key Exchange for IPsec VPNs』フィーチャ モジュール
Suite-B 整合性アルゴリズム タイプのトランスフォームの設定	『Configuring Internet Key Exchange Version 2 (IKEv2)』フィーチャ モジュール
IKEv2 用の Suite-B の Elliptic Curve Digital Signature Algorithm（ECDSA）signature（ECDSA-sig）認証方式の設定	『Configuring Internet Key Exchange Version 2 (IKEv2)』フィーチャ モジュール
IPsec SA ネゴシエーションでの Suite-B の Elliptic Curve Diffie-Hellman（ECDH）のサポート	『Configuring Internet Key Exchange for IPsec VPNs』および『Configuring Internet Key Exchange Version 2 (IKEv2)』フィーチャ モジュール
推奨暗号化アルゴリズム	『Next Generation Encryption』

MIB

MIB	MIB のリンク
なし。	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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 証明書登録の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *4*：*PKI* 証明書登録の機能情報

機能名	リリース	機能情報
証明書の自動登録	Cisco IOS XE Release 2.1	この機能では、証明書の自動登録が導入されています。これにより、ルータは、設定内のパラメータを使用する CA から自動的に証明書を要求できます。 次のコマンドがこの機能で導入されました。**auto-enroll**、**rsakeypair**、**showcryptocatimers**。

機能名	リリース	機能情報
証明書登録の拡張機能	Cisco IOS XE Release 2.1	この機能では、5 つの新しい **cryptocatrustpoint** コマンドが導入されています。これらのサブコマンドでは、証明書要求用に新しいオプションが提供されているので、ユーザはプロンプトを最後まで進む必要はなく、設定でフィールドを指定できます。 この機能により、**ip-address**（CA トラストポイント）、**password**（CA トラストポイント）、**serial-number**、**subject-name**、**usage** の各コマンドが導入されました。
HTTP による CA サーバへの直接登録	Cisco IOS XE Release 2.1	ユーザの CA サーバが SCEP をサポートしておらず、また RA モード CS を使用しない場合、この機能を使用すると、登録プロファイルを設定できます。登録プロファイルにより、HTTP 要求を RA モード CS ではなく CA サーバに直接送信できます。 次のコマンドがこの機能で導入されました。**authenticationcommand**、**authenticationterminal**、**authenticationurl**、**cryptocaprofileenrollment**、**enrollmentcommand**、**enrollmentprofile**、**enrollmentterminal**、**enrollmenturl**、**parameter**。

機能名	リリース	機能情報
RSA キーペアおよびPEM形式証明書のインポート	Cisco IOS XE Release 2.1	この機能を使用すると、証明書要求を発行したり、PEM 形式ファイルで発行された証明書を受け取ることができます。 この機能により、**enrollment** および **enrollmentterminal** コマンドが変更されました。
証明書更新用のキーロールオーバー	Cisco IOS XE Release 2.1	この機能では、証明書が失効する前に証明書の更新要求を行い、新しい証明書が使用可能になるまで古いキーと証明書を保持できます。 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。**auto-enroll** および **regenerate**
手動での証明書登録（TFTP によるカットアンドペースト）	Cisco IOS XE Release 2.1	この機能では、TFTP サーバまたは手動でのカットアンドペースト操作により、証明書要求を生成し、CA 証明書およびルータの証明書を受け取ることができます。 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。**cryptocaimport**、**enrollment**、および **enrollmentterminal**
永続的自己署名証明書	Cisco IOS XE Release 2.1	この機能により、HTTPS サーバは自己署名証明書を生成し、ルータのスタートアップ コンフィギュレーションに保存できます。そのため、それ以降のクライアントと HTTPS サーバ間の SSL ハンドシェイクで、ユーザが介入しなくても同じ自己署名証明書が使用されます。 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。**enrollmentselfsigned**、**showcryptopkicertificates**、**showcryptopkitrustpoints**。

機能名	リリース	機能情報
PKI ステータス	Cisco IOS XE Release 2.1	この拡張では、**showcryptopkitrustpoints** コマンドに **status** キーワードが追加されました。これにより、トラストポイントの現在のステータスを表示できます。 （注） これはマイナーな拡張です。マイナーな拡張は、通常 Feature Navigator に記載されません。
既存の証明書を使用した再登録	Cisco IOS XE Release 2.1	この機能では、既存の証明書を使用して、ルータをサードパーティ ベンダー製の CA から Cisco IOS CA に再登録できます。 この機能により、**enrollmentcredential** および **grantautotrustpoint** コマンドが導入されました。

機能名	リリース	機能情報
IOS SW の暗号化での Suite-B のサポート	Cisco IOS XE Release 3.7S	Suite-B によって、PKI の証明書登録に次のサポートが追加されます。 • X.509 証明書内の署名操作で、Elliptic Curve Digital Signature Algorithm（ECDSA）（256 ビットおよび384 ビットの曲線）が使用されます。 • ECDSA の署名を使用した X.509 証明書の確認で PKI がサポートされます。 • ECDSA の署名を使用した証明書要求の生成、および発行された証明書の IOS へのインポートで、PKI がサポートされます。 Suite B の要件は、IKE および IPSec で使用するための暗号化アルゴリズムの 4 つのユーザインターフェイススイートで構成され、RFC 4869 に記述されています。各スイートは、暗号化アルゴリズム、デジタル署名アルゴリズム、キー合意アルゴリズム、ハッシュまたはメッセージダイジェストアルゴリズムで構成されています。Cisco IOS での Suite-B サポートに関する詳細については、『*Configuring Security for VPNs with IPsec*』フィーチャモジュールを参照してください。
トラストポイント CLI	Cisco IOS XE Release 2.1	この機能では、**cryptopkitrustpoint** コマンドが導入されています。これにより、トラストポイント CA をサポートできるようになりました。

第 6 章

PKI クレデンシャル失効アラート

PKI クレデンシャル失効アラート機能を使用すると、CA 証明書が失効間近になるとアラート通知の形式で警告メカニズムが提供されます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

PKI クレデンシャル失効アラートの制約事項

アラートは、次の証明書には送信されません。

- 永続的または一時的な自己署名証明書。
- セキュアな固有デバイス識別子（SUDI）証明書。

- トラストプールに属する証明書。トラストプールには独自の失効アラートメカニズムがあります。
- トラストポイントのクローン。

PKI アラート通知の情報

アラート通知の概要

Cisco IOS 認証局（CA）サーバを使用すると、証明書が失効する前に証明書の自動登録が可能になり、認証中にアプリケーションの証明書が利用できるようになります。ただし、ネットワーク停止、クロック更新の問題、および CA の過負荷が証明書の更新に影響を与え、認証に有効な証明書が使用できなくなることでサブシステムがオフラインになります。PKI クレデンシャル失効アラート機能は、証明書の失効が近付くと、CA クライアントが syslog サーバに通知を送信するためのメカニズムを提供します。

通知は次の間隔で送信されます。

- 最初の通知：これは証明書が失効する 60 日前に送信されます。
- 通知の繰り返し：最初の通知の後、証明書が失効する 1 週間前まで後続の通知が毎週送信されます。最後の週には、証明書の失効日まで通知が毎日送信されます。

証明書の有効期限が 1 週間以上ある場合、通知は [warning]モードで送信されます。証明書の有効期限が 1 週間未満の場合、通知は [alert]モードで送信されます。通知には次の情報が含まれます。

- 証明書が関連付けられたトラストポイント
- 証明書タイプ
- 証明書のシリアル番号
- 証明書の発行元名
- 証明書が失効するまでの残り日数
- 証明書の自動登録が有効かどうか
- 対応する証明書のシャドウ証明書が利用可能かどうか

（注） アラート通知は syslog サーバまたは Simple Network Management Protocol（SNMP）トラップを介して送信されます。トラストポイントの自動登録が設定され、対応するシャドウまたはロールオーバー証明書が有効である場合、およびシャドウまたはロールオーバー証明書の開始時刻が証明書の終了時刻と同じまたはそれ以前の場合、通知は停止します。

この機能は無効にできず、設定作業を追加する必要はありません。**show crypto pki timers** コマンドはタイマーの有効期限情報を表示できるようになりました。次に、証明書の失効間近にタイマー

を表示する **show crypto pki timers detail** コマンドの出力例を示します。このタイマーが失効すると、通知が syslog サーバに送信されます。

```
Device# show crypto pki timers detail

PKI Timers
|290d 8:57:16.862
|290d 8:57:16.862 TRUSTPOOL
|985d11:54:50.783 SHADOW tp

Expiry Alert Timers
| 6d23:56:08.241
| 6d23:56:08.241 ID(tp)
|1034d23:54:50.783 CA(tp)
```

次に、デバイスに表示される syslog メッセージを示します。

```
Device#

Dec 16 10:24:13.533: %PKI-4-CERT_EXPIRY_WARNING: ID Certificate belonging to trustpoint tp
 will expire in 60 Days 0 hours 0 mins 0 secs.
Issuer-name cn=CA
Subject-name hostname=Router
Serial-number 02
Auto-Renewal: Not Enabled
```

PKI トラップ

PKI トラップでは、ネットワーク内のデバイスの証明書情報を取得するため、PKI 展開の監視と運用が簡単になります。ルートデバイスは、デバイスに設定されたしきい値に基づいて、ネットワーク管理システム（NMS）に SNMP トラップを定期的に送信します。トラップは次のシナリオで送信されます。

- 新しい証明書がインストールされる場合。SNMP トラップ（新しい証明書通知）は、証明書のシリアル番号、証明書の発行者名、証明書の所有者名、トラストポイント名、証明書タイプ、証明書の開始日と終了日などの情報を含む SNMP サーバに送信されます。
- 証明書が失効間近の場合。SNMP トラップ（証明書失効通知）は、証明書の終了日の 60 日から 1 週間前まで SNMP サーバに定期的に送信されます。証明書が失効する週には、トラップが毎日送信されます。トラップには、証明書のシリアル番号、証明書の発行者名、トラストポイント名、証明書タイプ、証明書の寿命などの証明書情報が含まれます。

PKI トラップを有効にするには、**snmp-server enable traps pki** コマンドを使用します。

（注） シャドウまたはロールオーバー証明書の開始時間が証明書の終了時間よりも遅い場合、シャドウ証明書が有効でないことを示すトラップが送信されます。ただし、同じトラストポイントで利用可能なシャドウ証明書とシャドウ証明書が有効な場合には、トラップは送信されません。

PKI クレデンシャル失効アラートの追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
セキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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 クレデンシャル失効アラートの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 5：***PKI*** **クレデンシャル失効アラートの機能情報**

機能名	リリース	機能情報
PKI クレデンシャル失効アラート	Cisco IOS XE Release 3.15S	PKI クレデンシャル失効アラート機能を使用すると、CA 証明書が失効間近になるとアラート通知の形式で警告メカニズムが提供されます。 次のコマンドが変更されました。**show crypto pki timers**

第 7 章

PKI 展開での Cisco IOS XE 証明書サーバの設定および管理

この章では、Cisco IOS 証明書サーバを設定および管理して、公開キー インフラストラクチャ（PKI）を展開する方法を説明します。証明書サーバは、Cisco ソフトウェアに簡単な証明書サーバを組み込んでいますが、認証局（CA）機能は限定されています。したがって、ユーザには次のようなメリットがあります。

- デフォルト動作の定義による、PKI 展開の簡素化。デフォルト動作が事前に定義されているので、ユーザインターフェイスが簡素化されています。つまり、CA が提供する証明書の拡張子をすべて使用しなくても PKI のスケーリングのメリットを活用できます。これにより、基本的な PKI で保護されたネットワークを簡単にイネーブルにできます。
- Cisco ソフトウェアとの直接統合。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイト ペーパーを参照してください。

- PKI 展開での Cisco IOS XE 証明書サーバの設定および管理に関する機能情報，207 ページ

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

Cisco IOS XE 証明書サーバの設定に関する前提条件

証明書サーバ設定前の PKI の計画

Cisco IOS XE 証明書サーバを設定する前に、PKI 内で使用する設定に対して適切な値（証明書のライフタイムおよび証明書失効リスト（CRL）ライフタイムなど）を考えて、選択することが重要です。証明書サーバに設定値が設定され、証明書が許可されたら、証明書サーバを再設定し、ピアを再登録することで、設定を変更できます。証明書サーバのデフォルト設定と推奨設定に関する詳細については、「証明書サーバのデフォルト値および推奨値」の項を参照してください。

HTTP サーバのイネーブル化

証明書サーバは、HTTP 上で Simple Certificate Enrollment Protocol（SCEP）をサポートします。証明書サーバが SCEP を使用するには、ルータで HTTP サーバをイネーブルにする必要があります（HTTP サーバをイネーブルにするには、**ip http server** コマンドを使用します）。HTTP サーバのイネーブルとディセーブルを切り替えると、証明書サーバは SCEP サービスのイネーブルとディセーブルを自動的に切り替えます。HTTP サーバがイネーブルでない場合は、手動の PKCS10 登録だけがサポートされます。

（注） 証明書サーバのすべてのタイプで自動CA証明書およびキーペアのロールオーバー機能を利用するには、SCEP を登録方式として使用する必要があります。

信頼性の高い時刻サービスの設定

証明書サーバは信頼できる時刻を認識する必要があるので、時刻サービスをルータで実行する必要があります。ハードウェアクロックを利用できない場合、証明書サーバはネットワークタイム

プロトコル（NTP）などの、手動で設定したクロック設定に依存します。ハードウェア クロックがない、あるいはクロックが無効な場合、起動時に次のメッセージが表示されます。

```
% Time has not been set. Cannot start the Certificate server.
```

クロックが設定されると、証明書サーバは実行ステータスに自動的に切り替わります。

クロック設定を手動で設定する方法については、『*Basic System Management Configuration Guide, Cisco IOS XE Release 3S*』の「Setting Time and Calendar Services」モジュールを参照してください。

Cisco IOS XE 証明書サーバの設定に関する制約事項

証明書サーバは、クライアントから受信した証明書要求を変更するメカニズムを備えていません。つまり、証明書サーバから発行される証明書は変更されていないため、その要求された証明書と一致します。名前制約などの固有の証明書ポリシーを発行する必要がある場合は、このポリシーを証明書要求に反映する必要があります。

Cisco IOS XE 証明書サーバの情報

証明書サーバの RSA キー ペアと証明書

証明書サーバは、1024 ビット Rivest、Shamir、Adelman（RSA）キー ペアを自動的に生成します。異なるキー ペア モジュラスが必要な場合は、手動で RSA キー ペアを生成する必要があります。この作業の完了に関する詳細については、「証明書サーバの *RSA* キー ペアの生成」を参照してください。

（注） 証明書サーバの RSA キー ペアで推奨されるモジュラスは、2048 ビットです。

証明書サーバは、CA キーとして通常の Cisco IOS XE RSA キー ペアを使用します。このキー ペアには、証明書サーバと同じ名前を付ける必要があります。証明書サーバがルータ上に作成される前にキーペアを生成していない場合、証明書サーバの設定時に、汎用目的キーペアが自動的に生成されます。

CA 証明書および CA キーが証明書サーバによって一度生成されると、これらを自動的にバックアップできます。その結果、バックアップ目的のエクスポート可能な CA キーを生成する必要はなくなりました。

自動生成キー ペアの処理方法

キーペアが自動的に生成されると、キーペアにエクスポート可能のマークは付けられません。そのため、CA キーをバックアップする場合は、キー ペアをエクスポート可能なものとして手動で生成する必要があります。この作業の完了方法については、「証明書サーバの *RSA* キー ペアの生成」を参照してください。

CA 証明書および CA キーを自動的にアーカイブする方法

CA 証明書および CA キーの原本または元の設定が失われた場合に CA 証明書および CA キーを後で復元できるように、初期の証明書サーバ設定時に、CA 証明書および CA キーの自動アーカイブをイネーブルにできます。

CA 証明書および CA キーは、証明書サーバを初めて起動したときに生成されます。また、自動アーカイブがイネーブルになっている場合、CA 証明書と CA キーはサーバ データベースにエクスポート（アーカイブ）されます。アーカイブは、PKCS12 形式またはプライバシーエンハンスト メール（PEM）形式で実行できます。

（注） このCAキーのバックアップファイルは非常に重要なので、すぐに別の安全な場所に移動する必要があります。

- このアーカイブ処理は、1 回しか実行されません。（1）手動で生成され、エクスポート可能のマークが付けられた CA キー、または（2）証明書サーバによって自動的に生成された CA キーだけがアーカイブされます（このキーには、エクスポート不可能のマークが付けられます）。
- 手動で CA キーを生成し、そのキーに「エクスポート不可能」のマークが付いている場合、自動アーカイブは実行されません。
- CA 証明書および CA キー アーカイブ ファイル以外にも、シリアル番号ファイル（.ser）および CRL ファイル（.crl）を定期的にバックアップする必要があります。証明書サーバを復元する必要がある場合、CA 運用においてシリアル ファイルおよび CRL ファイルは重要です。
- エクスポート不可能な RSA キーまたは手動で生成されたエクスポート不可能な RSA キーを使用するサーバを手動でバックアップできません。自動的に生成された RSA キーには、エクスポート不可能のマークが付いていますが、このキーは一度だけ自動的にアーカイブされます。

証明書サーバ データベース

Cisco IOS XE 証明書サーバは専用のファイルを保管し、他のプロセスに使用するファイルを公開できます。証明書サーバによって生成された、進行中の操作に必要な重要ファイルは、専用のファイル タイプごとに 1 つの場所に保管されます。証明書サーバはこれらのファイルに対して読み取りおよび書き込みを行います。重要な証明書サーバ ファイルは、シリアル番号ファイル（.ser）と CRL 保管場所ファイル（.crl）です。証明書サーバによって書き込みが行われても再度読み取りが行われないファイルは場合によって公開され、他のプロセスで使用できます。公開可能なファイルの例には、発行済みの証明書ファイル（.crt）があります。

証明書サーバのパフォーマンスは、次の要因から影響を受ける場合があります。証明書サーバ ファイルに対して、保管オプションおよび公開オプションを選択するときには、これらの要因を考慮する必要があります。

- 選択する保管場所または公開場所が証明書サーバのパフォーマンスに影響を与えることがあります。ネットワーク ロケーションから読み取ると、ルータのローカル ストレージ デバイスから直接読み取るよりも時間がかかります。
- 特定の場所では、保管または公開するファイルの数によって証明書サーバのパフォーマンスが影響を受けることがあります。ローカルの Cisco IOS XE ファイル システムは、必ずしも大量のファイルに適していません。
- 保管または公開するファイル タイプが証明書サーバのパフォーマンスに影響を与えることがあります。特定のファイル（.crl ファイルなど）は非常に大きくなる可能性があります。

（注） ローカルの Cisco IOS XE ファイル システムに .ser および .crl ファイルを保管し、リモート ファイル システムに.crt ファイルを公開することを推奨します。

証明書サーバ データベース ファイルの保管

証明書サーバは、その柔軟性により、設定されたデータベース レベルに応じて、さまざまな種類の重要なファイルをさまざまな保管場所に保管できます（詳細については、**databaselevel** コマンドを参照してください）。保管場所を選択するときは、必要なファイル セキュリティおよびサーバのパフォーマンスを考慮してください。たとえば、シリアル番号ファイルおよびアーカイブ ファイル（.p12 または .pem）では、発行された証明書ファイル（.crt）の保管場所または名前ファイル（.cnm）の保管場所よりもセキュリティ上の制約事項が多くなることがあります。

次の表に、特定の場所に保管される重要な証明書サーバ ファイルのタイプをファイル拡張子別に示します。

表 *6*：証明書サーバの保管場所と重要なファイル タイプ

ファイル拡張子	ファイル タイプ
.ser	メイン証明書サーバのデータベース ファイル
.crl	CRL の保管場所
.crt	発行された証明書の保管場所
.cnm	証明書名および失効ファイルの保管場所
.p12	PKCS12 形式の証明書サーバ証明書アーカイブ ファイルの保管場所
.pem	PEM 形式の証明書サーバ証明書アーカイブ ファイルの保管場所

Cisco IOS XE 証明書サーバ ファイルには、次の 3 つのレベルで保管場所を指定できます。

- デフォルトの場所（NVRAM）
- すべての重要ファイルに対して指定されたプライマリ保管場所
- 特定の重要ファイルに対して指定された保管場所

ファイルは、一般的な保管場所よりも、具体的に設定した保管場所に優先的に保管されます。たとえば、証明書サーバファイルの保管場所を指定しなかった場合、すべての証明書サーバファイルが NVRAM に保管されます。名前ファイルの保管場所を指定すると、名前ファイルだけがそこに保管され、その他すべてのファイルはNVRAM に保管されます。プライマリ ロケーションを指定すると、名前ファイル以外のすべてのファイルが、NVRAM の代わりに、この場所に保管されます。

（注） .p12 または .pem のいずれかを指定できますが、両方のタイプのアーカイブ ファイルは一度に指定できません。

証明書サーバ データベース ファイルの公開

公開ファイルは元のファイルのコピーで、他のプロセスまたはユーザ用に使用できます。証明書サーバがファイルの公開に失敗すると、サーバはシャットダウンします。発行された証明書ファイルおよび名前ファイルに 1 つの公開場所を、CRL ファイルに複数の公開場所を指定できます。公開可能なファイルタイプについては、次の表を参照してください。設定されたデータベース レベルに関係なく、ファイルを公開できます。

表 7: 証明書サーバの公開ファイル タイプ

ファイル拡張子	ファイル タイプ
.crl	CRL の公開場所
.crt	発行された証明書の公開場所
.cnm	証明書名および失効ファイルの公開場所

証明書サーバのトラストポイント

自動的に生成された同じ名前のトラストポイントも証明書サーバにある場合、そのトラストポイントが証明書サーバの証明書を保管します。証明書サーバの証明書を保管するためにトラストポイントが使用されていることを、ルータが検出すると、トラストポイントはロックされ変更できなくなります。

証明書サーバを設定する前に、次の操作を行います。

- このトラストポイントを手動で作成し、設定します（**crypto pki trustpoint** コマンドを使用）。これにより、代替 RSA キー ペアを指定できます（ **rsakeypair** コマンドを使用）。
- **on** コマンドを使用して、設定済みの利用可能な USB トークンなどの特定のデバイス上に初期の自動登録キー ペアが生成されるように指定します。

（注） 自動的に生成されたトラストポイントおよび証明書サーバ証明書は、証明書サーバ デバイスのアイデンティティには使用できません。したがって、CA トラストポイントを指定して証明書を入手して接続しているクライアントの証明書を認証するために使用されるコマンドライン インターフェイス（CLI）（**ip http secure-trustpoint** コマンドなど）は、証明書サーバ デバイス上に設定された追加のトラストポイントをポイントする必要があります。

サーバがルート証明書サーバの場合、このサーバはRSA キー ペアおよびその他いくつかの属性を使用して自己署名証明書を生成します。関連付けられる CA 証明書には、デジタル署名、証明書署名および CRL 署名といった拡張キー用途があります。

CA 証明書の生成後の属性変更は、証明書サーバが壊れた場合に限りできます。

（注） **auto-enroll** コマンドを使用して、証明書サーバ トラストポイントを自動的に登録しないでください。証明書サーバの初期登録を手動で開始する必要があります。また、**auto-rollover** コマンドを使用して、進行中の自動ロールオーバー機能を設定できます。自動ロールオーバー機能の詳細については、自動 CA 証明書およびキー ロールオーバー, （162 ページ）の項を参照してください。

証明書失効リスト（CRL）

デフォルトでは、CRL は 168 時間（1 週間）に 1 度発行されます。CRL を発行するために、デフォルト値以外の値を指定するには、**lifetimecrl** コマンドを実行します。CRL は発行されると、*ca-label*.crl として指定されたデータベースの場所に書き込まれます。この *ca-label* は、証明書サーバの名前です。

CRL は、設定済みで利用可能な場合、SCEP（デフォルト方式）または CRL 配布ポイント（CDP）を介して配布できます。CDP を設定する場合は、**cdp-url** コマンドを使用して、CDP の場所を指定します。**cdp-url** コマンドが指定されていない場合、証明書サーバによって発行される証明書には CDP 証明書拡張子が含まれません。CDP の場所が指定されていない場合は、Cisco IOS PKI クライアントは SCEP GetCRL メッセージを使用して証明書サーバから自動的に CRL を要求します。CA は、SCEP CertRep メッセージで CRL をクライアントに返します。すべての SCEP メッセージは、エンベロープ化された署名付き PKCS#7 データであるため、証明書サーバから CRL の SCEP を取得すると、コストがかかるうえに、拡張性はあまり高くありません。非常に大規模なネットワークでは、HTTP CDP の方が拡張性が向上するため、CRL をチェックするピア デバイスが多い場合は、HTTP CDP を推奨します。たとえば、次のように簡単な HTTP URL ストリングによって CDP の場所を指定できます。

cdp-url http://my-cdp.company.com/filename.crl

証明書サーバは、CDP を 1 つだけサポートします。したがって、発行される証明書には、すべて同じ CDP が含まれます。

Cisco IOS ソフトウェアを実行せず、SCEP GetCRL 要求をサポートしない PKI クライアントがある状態で CDP を使用する場合、外部サーバを設定して CRL を配布し、このサーバをポイントするように CDP を設定できます。または、次の形式の URL で **cdp-url** コマンドを指定すると、証明書サーバから CRL を取得するために非 SCEP 要求を指定できます。この *cs-addr* は証明書サーバの場所です。

cdp-url http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL

（注） また、Cisco IOS XE CA が HTTP CDP サーバとしても設定されている場合、**cdp-url** http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL コマンド構文を使用して CDP を指定してください。

cdp-url コマンドによって指定された場所から CRL を利用できるかどうかは、ネットワーク管理者が確認してください。

指定された場所内に埋め込まれた疑問符を保持するようパーサーに強制するには、疑問符の前に Ctrl+V キーを入力します。この処理を実行しないと、HTTP による CRL 取得でエラー メッセージが返されます。

CDP の場所は、証明書サーバが実行されてから、**cdp-url** コマンドによって変更できます。新しい証明書には、更新された CDP の場所が含まれていますが、既存の証明書は、新たに指定された CDP 場所を含まない状態で再発行されます。新しいCRLが発行されると、証明書サーバは、キャッシュされた現在の CRL を使用して新しい CRL を生成します（証明書サーバが再起動されると、データベースから現在の CRL をリロードします）。現在の CRL が失効するまで、新しい CRL は発行できません。現在の CRL が失効すると、CLI から証明書を無効にしたときにだけ、新しい CRL が発行されます。

証明書サーバのエラー状態

証明書サーバは起動時、証明書を発行する前に現在の設定をチェックします。証明書サーバは、**show crypto pki server** コマンドの出力で、最後に認識されたエラー状態を報告します。たとえば、エラー状態には次のものがあります。

- 保管場所にアクセスできない
- HTTP サーバを待機する
- 時間設定を待機する

証明書サーバに、CRL の公開に失敗するなどの重大な障害が発生した場合、証明書サーバは自動的に使用不可状態になります。この場合、ネットワーク管理者がエラー状態を解消できます。エラーを解消すると、証明書サーバは直前の正常な状態に戻ります。

証明書サーバを使用した証明書登録

証明書登録要求は、次のように機能します。

- 証明書サーバがエンドユーザから登録要求を受け取ると、次の処理が発生します。
  - 要求エントリが、初期状態で登録要求データベースに作成されます（証明書登録の要求状態のリストについては、次の表を参照してください）。
  - 証明書サーバは、CLI 設定（パラメータが指定されていない場合は、デフォルト動作）を参照して、要求を許可するかどうか決定します。その後、登録要求の状態は登録要求データベースで更新されます。
- SCEP クエリーごとに応答するため、証明書サーバは現在の要求を調べ、次のいずれかの処理を実行します。
  - エンドユーザに「保留」または「拒否」状態で応答します。
  - 適切な証明書を生成して署名し、証明書を登録要求データベースに保管します。

クライアントの接続が終了すると、証明書サーバは、クライアントが別の証明書を要求するまで待機します。

すべての登録要求は、次の表に定義する証明書登録状態に移行します。現在の登録要求を表示するには、**cryptopkiserverrequestpkcs10** コマンドを使用します。

表 *8*：証明書登録要求状態の説明

証明書登録の状態	説明
許可	証明書サーバは要求を認可しました。
拒否	証明書サーバは、ポリシー上の理由で要求を拒否しました。
付与	CA コアは、証明書要求に対して適切な証明書を生成しました。
初期	SCEP サーバによって要求が作成されました。
形式異常	証明書サーバは、暗号化上の理由により、要求が無効であると判断しました。
保留中	ネットワーク管理者が登録要求を手動で受け入れる必要があります。

SCEP 登録

すべての SCEP 要求は新しい証明書の登録要求として処理されます。SCEP 要求で前の証明書要求と重複する所有者名または公開のキー ペアが指定された場合も同様です。

CA サーバのタイプ：下位および登録局（RA）

CA サーバは、下位の証明書サーバまたは RA モード証明書サーバとして設定できるように柔軟性を備えています。

下位 CA を設定する理由とは

下位証明書サーバは、ルート証明書サーバと同じ機能を提供します。ルート RSA キー ペアは、PKI 階層構造においてきわめて重要で、多くの場合、このキー ペアをオフラインにしておくか、アーカイブしておくことが得策です。この要件をサポートするために、PKI 階層に、ルート権限で署名された下位 CA を組み込めます。このように、通常の動作時には、ルート権限をオフラインにして（特別な CRL 更新を発行する場合を除く）、下位 CA を使用できます。

RA モード証明書サーバを設定する理由とは

Cisco IOS XE 証明書サーバは、RA モードで実行できるように設定できます。RA は、CA から認証および認可責任をオフロードします。RA が SCEP または手動での登録要求を受信すると、管理者はローカル ポリシーごとに要求を拒否または許可できます。要求が許可されると、その要求は発行元 CA に転送され、CA は自動的に証明書を生成して RA に返します。クライアントは、許可された証明書を RA から後で取得できます。

RA とは、CA が証明書を発行するために必要なデータの一部またはすべてを記録あるいは検証する役割を担う機関です。多くの場合、CA は RA の機能自体をすべて請け負いますが、CA が広範囲の地理的エリアで運用されている、あるいは CA がネットワーク アクセスに直接さらされるというセキュリティ上の懸念がある場合、管理上好ましいのは、作業の一部を RA に委任して、CA が基本作業である証明書および CRL の署名に集中できるようにすることです。

CA サーバの互換性

CA サーバの互換性によって、RA モードの IOS CA サーバは複数のタイプの CA サーバと相互運用できます。詳細については、「証明書サーバを *RA* モードで実行するように設定」を参照してください。

自動 CA 証明書およびキー ロールオーバー

CA（ルート CA、下位 CA、および RA モード CA）は、クライアントと同様、有効期限付きの証明書とキー ペアを持っており、これらの証明書とキー ペアは、現在の証明書とキー ペアが失効するときに再発行する必要があります。ルート CA の証明書とキー ペアが失効すると、CA は自己署名付きロールオーバー証明書とキーペアを生成する必要があります。下位 CA または RA モード CA の証明書およびキー ペアが失効すると、CA は、その上位 CA からロールオーバー証明書と

キーペアを要求すると同時に上位CAの新しい自己署名付きロールオーバー証明書を取得します。CAは、そのすべてのピアに新しいCAロールオーバー証明書とキーを配布する必要があります。CA およびそのクライアントが失効する CA 証明書とキー ペアから新しい CA 証明書とキー ペアに切り替えている間に、ロールオーバーと呼ばれるプロセスにより、ネットワークは中断せずに動作します。

ロールオーバーは、PKI インフラストラクチャの信頼関係の要件および同期化されたクロックに依存します。PKI の信頼関係により、（1）新しい CA 証明書の認証が可能になり、（2）セキュリティが損なわれることなく、ロールオーバーを自動的に実行できます。同期化されたクロックにより、ロールオーバーをネットワーク全体で調整できます。

自動 CA 証明書ロールオーバーの動作原理

CA サーバには、ロールオーバーが設定されている必要があります。すべてのレベルのCAを自動的に登録し、**自動ロールオーバー**をイネーブルにする必要があります。CAクライアントは、自動的に登録されると、自動的にロールオーバーをサポートします。クライアントおよび自動ロールオーバーの詳細については、「PKI の証明書登録の設定」の章にある「自動証明書登録」を参照してください。

CA がロールオーバーをイネーブルにして、そのクライアントが自動的に登録された後に、3 段階の自動 CA 証明書ロールオーバー プロセスがあります。

1 段階：アクティブな CA 証明書およびキー ペアのみ

1 段階には、アクティブな CA 証明書およびキー ペアだけがあります。

2 段階：CA 証明書のロールオーバーおよびキー ペアの生成と配布

2 段階では、ロールオーバー CA 証明書およびキー ペアが生成され、配布されます。上位 CA はロールオーバー証明書とキー ペアを生成します。CA が正常にアクティブな設定を保存すると、CA はロールオーバー証明書およびキー ペアのクライアント要求に応答する準備が完了です。上位CA がクライアントから新しいCA 証明書とキー ペアに対する要求を受信すると、CA は、新しいロールオーバー CA 証明書とキー ペアを要求元クライアントに送信して応答します。クライアントは、ロールオーバー CA 証明書とキー ペアを保管します。

（注） CA は、ロールオーバー証明書とキー ペアを生成したときに、そのアクティブな設定を保存できる必要があります。現在の設定が変更された場合、ロールオーバー証明書とキー ペアは自動的には保存されません。この場合、管理者は手動で設定を保存する必要があります。保存しない場合、ロールオーバー情報は失われます。

3 段階：ロールオーバー CA 証明書とキー ペアがアクティブな CA 証明書とキー ペアになる

3 段階では、ロールオーバー CA 証明書とキー ペアがアクティブな CA 証明書とキー ペアになります。有効なロールオーバー CA 証明書を保管しているすべてのデバイスは、ロールオーバー証明書をアクティブな証明書の名前に変更し、それまでアクティブだった証明書とキー ペアは削除されます。

CA 証明書のロールオーバー後、通常の証明書のライフタイムおよび更新時間との間に次のような時間の違いがあることがわかる場合があります。

- ロールオーバー中に発行された証明書のライフタイムは、あらかじめ設定された値よりも低くなります。
- 特定の条件下では、更新時間が実際のライフタイムの設定割合よりも低くなる場合があります。証明書のライフタイムが 1 時間未満の場合に確認される違いは、20% までになることがあります。

このような違いがあるのは通常の状態であり、証明書サーバ上のアルゴリズムで発生する **jitter**（ランダムな時間の変動）によるものです。この作業は、PKI に参加するホストが自分の登録タイマーと同期しないようにするために実行します。同期すると、証明書サーバで輻輳が発生する場合があります。

（注） 発生するライフタイムの変動は、常にライフタイムが短くなるように発生し、証明書に対して設定された最大ライフタイム内に収まるため、PKI の適切な動作に悪影響を与えることはありません。

暗号化ハッシュ関数を指定するためのサポート

セキュア ハッシュ アルゴリズム（SHA）を使用すると、ユーザは Cisco IOS XE 証明書サーバおよびクライアントの暗号化ハッシュ関数を指定できます。指定できる暗号化ハッシュ関数は、メッセージ ダイジェスト アルゴリズム 5（MD5）、SHA-1、SHA-256、SHA-384、または SHA-512 です。

（注） シスコは MD5 の使用を推奨しません。その代わりに SHA-256 を使用する必要があります。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイト ペーパーを参照してください。

この機能の実装に使用される **hash** (ca-trustpoint) および **hash** (cs-server) コマンドの指定に関する詳細については、「下位証明書サーバの設定」の作業を参照してください。

HA サポート

ハイ アベイラビリティ（HA）は、Cisco ASR 1000 シリーズ アグリゲーション サービス ルータの証明書サーバでサポートされます。HA を使用すると、アクティブ プロセッサとスタンバイ プロセッサ間の設定、CRL ファイル、およびシリアル番号ファイルをデバイスで同期できます。

PKI サーバ向けの Cisco ASR 1000 シリーズ アグリゲーション サービス ルータは、ルート間プロセッサ（RP）の PKI 冗長性のみをサポートします。RP 間の PKI は暗示的な処理であり、特別な設定は必要ありません。Cisco IOS XE リリース 3.15S 以降のリリースでは、PKI サーバのハイ ア

ベイラビリティ（HA）aka 冗長性を使用して、Cisco ASR 1000 シリーズ アグリゲーション サービス ルータを PKI サーバのように機能させることができます。

Cisco IOS XE 証明書サーバの設定および展開方法

証明書サーバの RSA キー ペアの生成

証明書サーバの 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。証明書サーバの RSA キーペアを手動で生成すると、生成しようとするキーペアのタイプの指定、バックアップ目的のエクスポート可能なキーペアの作成、キーペアの保管場所の指定、またはキー生成場所の指定ができます。

（注） バックアップまたはアーカイブ目的でエクスポート可能な証明書サーバ キー ペアを作成するとします。この作業を実行しない場合、証明書サーバは自動的にキーペアを生成し、このキーペアにはエクスポート可能のマークが付けられます。

デバイスで USB トークンを設定し、それが利用可能な場合、USB トークンは、ストレージ デバイスとしてだけでなく、暗号化デバイスとしても使用できます。USB トークンを暗号化装置として使用すると、USB トークンでクレデンシャルのキー生成、署名、認証などの RSA 操作を実行できます。秘密キーは決して USB トークンから出ないようになっており、エクスポートできません。公開キーはエクスポート可能です。USB トークンの設定および暗号装置としての使用に関する具体的なマニュアルのタイトルについては、「関連資料」を参照してください。

（注） 秘密キーを安全な場所に保管し、定期的に証明書サーバ データベースをアーカイブすることを推奨します。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイト ペーパーを参照してください。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptokeygeneratersa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label***key-label*] [**exportable**] [**modulus***modulus-size*] [**storage***devicename:*] [**on***devicename:*]
4. **cryptokeyexportrsa***key-label***pem** {**terminal** | **url***url*} {**3des** | **des**} *passphrase*
5. **cryptokeyimportrsa***key-label***pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url***url*} [**exportable**] [**on***devicename:*] *passphrase*
6. **exit**
7. **showcryptokeymypubkeyrsa**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**cryptokeygeneratersa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label***key-label*] [**exportable**] [**modulus***modulus-size*] [**storage***devicename:*] [**on***devicename:*] 例： `Device(config)# crypto key generate rsa label mycs exportable modulus 2048`	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、**cryptopkiserver***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、ルータの完全修飾ドメイン名（FQDN）であるデフォルト値が使用されます。 **noshutdown** 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、**cryptocaexportpkcs12** 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 RSA キーのモジュラス サイズは 1024 ビットです。推奨される CA RSA キーのモジュラスは 2048 ビットです。CA RSA キーのモジュラス サイズの範囲は 350 ～ 4096 ビットです。

	コマンドまたはアクション	目的
		• **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。この装置にはユニバーサルシリアルバス（USB）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	**cryptokeyexportrsa***key-label***pem** {**terminal** | **url***url*} {**3des** | **des**} *passphrase* 例： `Device(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	（任意）生成された RSA キー ペアをエクスポートします。 生成されたキーをエクスポートできます。
ステップ 5	**cryptokeyimportrsa***key-label***pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url***url*} [**exportable**] [**on***devicename:*] *passphrase* 例： `Device(config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD`	（任意）RSA キー ペアをインポートします。 USB トークンにインポートするキーを作成するには、使用する **on** キーワードおよび適切なデバイスの場所を指定します。 **exportable** キーワードを使用して RSA キーをエクスポートし、RSA キー ペアをエクスポート不可に変更する場合は、**exportable** キーワードを使用せずに証明書サーバにキーを再度インポートします。キーを再度エクスポートできません。
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーションを終了します。
ステップ 7	**showcryptokeymypubkeyrsa** 例： `Device# show crypto key mypubkey rsa`	ルータの RSA 公開キーを表示します。

例

次の例では、「ms2」というラベルの USB トークンに汎用 1024 ビット RSA キー ペアを生成し、それとともに表示される暗号エンジンのデバッギング メッセージを示します。

```
Device(config)# crypto key generate rsa on usbtoken0 label ms2 modulus 2048
The name for the keys will be: ms2
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:41:40.895: crypto_engine: Generate public/private keypair [OK]
```

```
Jan 7 02:44:09.623: crypto_engine: Create signature
Jan 7 02:44:10.467: crypto_engine: Verify signature
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_ISA_RSA_PUB_DECRYPT(hw)(ipsec)
```

これで、「ms2」というラベルが付けられた、トークン上のキーを登録に使用できます。

次の例では、設定済みの利用可能な USB トークンに正常にインポートされた暗号キーを示します。

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# crypto key import rsa encryption on usbtoken0 url nvram:e password

% Importing public Encryption key or certificate PEM file...
filename [e-encr.pub]?
Reading file from nvram:e-encr.pub
% Importing private Encryption key PEM file...
Source filename [e-encr.prv]?
Reading file from nvram:e-encr.prv
% Key pair import succeeded.
```

証明書サーバの設定

自動 CA 証明書ロールオーバーに関する前提条件

証明書サーバを設定する場合、自動CA証明書ロールオーバーが正常に実行するために、CA サーバに次の前提条件が適用されます。

- CA サーバは、イネーブルにされ、信頼できる時刻、利用可能なキー ペア、キー ペアに関連付けられた自己署名付きの有効な CA 証明書、CRL、アクセス可能なストレージ デバイス、およびアクティブな HTTP/SCEP サーバとともに完全に設定されている必要があります。
- CA クライアントでは、自動登録が正常に完了しており、同じ証明書サーバへの自動登録がイネーブルになっている必要があります。

自動 CA 証明書ロールオーバーに関する制約事項

証明書サーバを設定する場合、自動 CA 証明書ロールオーバーを正常に実行するために、次の制約事項が適用されます。

- SCEPを使用してロールオーバーをサポートする必要があります。SCEPの代わりに証明書管理プロトコルまたはメカニズム（登録プロファイル、手動での登録、または TFTP による登録など）を使用して、PKI に登録する装置では、SCEP で提供されているロールオーバー機能を利用できません。
- ネットワークに自動アーカイブを設定していてもアーカイブが失敗する場合、証明書サーバがロールオーバー状態にならず、ロールオーバー証明書およびキー ペアが自動的に保存されないため、ロールオーバーは発生しません。

証明書サーバの設定

Cisco IOS XE 証明書サーバを設定し、自動ロールオーバーをイネーブルにするには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **iphttpserver**
4. **cryptopkiserver***cs-label*
5. **noshutdown**
6. **auto-rollover** [*time-period*]

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**iphttpserver** 例： `Device(config)# ip http server`	ご使用のシステムの HTTP サーバをイネーブルにします。
ステップ 4	**cryptopkiserver***cs-label* 例： `Device(config)# crypto pki server server-pki`	証明書サーバのラベルを定義し、証明書サーバ コンフィギュレーション モードを開始します。 （注） 手動で RSA キー ペアを生成した場合、*cs-label* 引数はキー ペアの名前と一致する必要があります。
ステップ 5	**noshutdown** 例： `Device(cs-server)# no shutdown`	（任意）証明書サーバをイネーブルにします。 （注） デフォルト機能を使用する場合は、この時点ではこのコマンドだけを使用します。つまり、デフォルト設定のいずれかを「証明書サーバ機能の設定」の作業に従って変更する場合、まだこのコマンドを発行しないでください。

	コマンドまたはアクション	目的
ステップ 6	**auto-rollover** [*time-period*] 例： `Device(cs-server)# auto-rollover 90`	（任意）自動CA証明書ロールオーバー機能をイネーブルにします。 • *time-period*：デフォルトは 30 日です。

例

次の例では、証明書サーバ「ms2」を設定する方法について示します。ms2 は 2048 ビット RSA キー ペアのラベルです。

```
Device(config)# crypto pki server ms2
Device(cs-server)# no shutdown

% Once you start the server, you can no longer change some of
% the configuration.
Are you sure you want to do this? [yes/no]:
yes
% Certificate Server enabled.
Device(cs-server)# end
!
Device# show crypto pki server ms2
Certificate Server ms2:
    Status: enabled, configured
    CA cert fingerprint: 5A856122 4051347F 55E8C246 866D0AC3
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 19:44:57 GMT Oct 14 2006

CRL NextUpdate timer: 19:45:25 GMT Oct 22 2003
    Current storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

次の例では、**auto-rollover** コマンドを使用して、サーバ ms2 の自動 CA 証明書ロールオーバーをイネーブルにする方法を示します。**show crypto pki server** コマンドを実行すると、自動ロールオーバーが 25 日のオーバラップ期間でサーバ mycs に設定されたことが示されます。

```
Device(config)# crypto pki server ms2
Device(cs-server)# auto-rollover 25
Device(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Exporting Certificate Server signing certificate and keys...
% Certificate Server enabled.
Device(cs-server)#
Device# show crypto pki server ms2
Certificate Server ms2:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
    CA cert fingerprint:70AFECA9 211CDDCC 6AA9D7FF 3ADB03AE
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:00:49:26 PDT Jun 20 2008
    CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
    Auto-Rollover configured, overlap period 25 days
    Autorollover timer:00:49:26 PDT May 26 2008
```

下位証明書サーバの設定

すべて、または特定の SCEP 証明書要求あるいは手動の証明書要求を許可するために下位証明書サーバを設定し、自動ロールオーバーをイネーブルにするには、次の作業を実行します。

（注） セキュリティの脅威と、それに対抗するための暗号化技術は常に変化しています。シスコの最新の暗号化に関する推奨事項については、『Next Generation Encryption（NGE）』ホワイトペーパーを参照してください。

はじめる前に

- ルート証明書サーバは、Cisco IOS XE 証明書サーバである必要があります。
- 下位の認証局（CA）の場合、ルート CA またはアップストリーム CA への登録は SCEP を介してのみ有効です。アップストリーム CA は、アップストリーム CA への登録が完了するまでオンラインである必要があります。ルート CA またはアップストリーム CA に下位 CA を手動で登録することはできません。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint** *name*
4. **enrollment** [**mode**] [**retry period***minutes*] [**retry count***number*] **url***url* [**pem**]
5. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
6. **exit**
7. **cryptopkiserver***cs-label*
8. **issuername***DN-string*
9. **mode sub-cs**
10. **auto-rollover** [*time-period*]
11. **grantautorollover** {**ca-cert** | **ra-cert**}
12. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
13. **noshutdown**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# 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	**cryptopkitrustpoint** *name* 例： `Device(config)# crypto pki trustpoint sub`	下位の証明書サーバが使用するトラストポイントを宣言し、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	**enrollment** [**mode**] [**retry period***minutes*] [**retry count***number*] **url***url* [**pem**]
	
	例：
	`Device(ca-trustpoint)# enrollment url http://caserver.myexample.com`
	または
	`Device(ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80`
		CA の次の登録パラメータを指定します。
		• （任意）CA システムが登録局（RA）を提供する場合、**mode** キーワードとして登録局（RA）モードを指定します。デフォルトでは、RA モードはディセーブルです。
		• （任意）**retry period** キーワードおよび *minutes* 引数は、CA に別の証明書要求を送信するまでルータが待機する期間を分単位で指定します。有効値は 1 ～ 60 です。デフォルトは 1 です。
		• （任意）**retry count** キーワードおよび *number* 引数は、直前の要求に対する応答をルータが受信しない場合、ルータが証明書要求を再送信する回数を指定します。有効な値は、1 ～ 100 です。デフォルトは 10 です。
		• *url* 引数は、ルータが証明書要求を送信する CA の URL です。
		（注） IPv6 アドレスは **http:** 登録方式に追加できます。たとえば、http://[ipv6-address]:80 です。URL 内の IPv6 アドレスは括弧で囲む必要があります。使用できるその他の登録方式に関する詳細については、*enrollment url (ca-trustpoint)* コマンド ページを参照してください。
		• （任意）**pem** キーワードは、証明書要求に Privacy Enhanced Mail（PEM）の境界を追加します。

ステップ 5	**hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**} 例： `Device(ca-trustpoint)# hash sha384`	（オプション）Cisco IOS XE クライアントが自己署名証明書の署名に使用する署名のハッシュ関数を指定します。デフォルトでは、Cisco IOS XE クライアントは MD5 暗号化ハッシュ関数を自己署名証明書に使用します。 トラストポイントのデフォルト値を上書きするように、次のコマンドアルゴリズム キーワードオプションのいずれかを指定できます。その後、この設定が、自己署名証明書のデフォルトの暗号化ハッシュ アルゴリズム関数になります。

	コマンドまたはアクション	目的
		• **md5**：デフォルトのハッシュ関数 MD5 が使用されるように指定します（推奨しません）。 • **sha1**：SHA-1 ハッシュ関数が RSA キーのデフォルトのハッシュ アルゴリズムとして使用されるように指定します（推奨しません）。 • **sha256**：SHA-256 ハッシュ関数が Elliptic Curve（EC）256 ビット キーのハッシュ アルゴリズムとして使用されるように指定します。 • **sha384**：SHA-384 ハッシュ関数が EC 384 ビット キーのハッシュ アルゴリズムとして使用されるように指定します。 • **sha512**：SHA-512 ハッシュ関数が EC 512 ビット キーのハッシュ アルゴリズムとして使用されるように指定します。
ステップ 6	**exit** 例： `Device(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 7	**cryptopkiserver***cs-label* 例： `Device(config)# crypto pki server sub`	Cisco IOS XE 証明書サーバをイネーブルにし、CS サーバ コンフィギュレーション モードを開始します。 （注） 下位のサーバには、上記ステップ 3 で作成されたトラストポイントと同じ名前を付ける必要があります。
ステップ 8	**issuername***DN-string* 例： `Device(cs-server)# issuer-name CN=sub CA, O=Cisco, C=us`	（任意）証明書サーバの CA 発行者名として DN を指定します。
ステップ 9	**mode sub-cs** 例： `Device(cs-server)# mode sub-cs`	PKI サーバをサブ証明書サーバ モードにします。 • 下位 CA と CA との関係は、ネットワーク上のすべてのデバイスが Cisco IOS XE デバイス タイプに含まれる場合のみサポートされます。そのため、Cisco IOS XE の下位 CA は、サード パーティの CA サーバに登録することはできません。
ステップ 10	**auto-rollover** [*time-period*] 例： `Device(cs-server)# auto-rollover 90`	（任意）自動 CA 証明書ロールオーバー機能をイネーブルにします。 • *time-period*：デフォルトは 30 日です。

	コマンドまたはアクション	目的
ステップ 11	**grantautorollover** {**ca-cert** | **ra-cert**} 例： `Device(cs-server)# grant auto rollover ca-cert`	（任意）オペレータが介入せずに、下位の CA および RA モード CA の再登録要求を自動的に許可します。 • **ca-cert**：下位の CA ロールオーバー証明書が自動的に付与されるように指定します。 • **ra-cert**：RA モード CA ロールオーバー証明書が自動的に付与されるように指定します。 （注） これが、初めて下位の証明書サーバをイネーブルにし、登録するときであれば、証明書要求を手動で許可する必要があります。
ステップ 12	**hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**} 例： `Device(cs-server)# hash sha384`	（任意）Cisco IOS XE 認証局（CA）はサーバから発行されたすべての証明書の署名に使用する署名のハッシュ関数を設定します。 • **md5**：デフォルトのハッシュ関数 MD5 が使用されるように指定します（推奨しません）。 • **sha1**：SHA-1 ハッシュ関数が使用されるように指定します（推奨しません）。 • **sha256**：SHA-256 ハッシュ関数が使用されるように指定します。 • **sha384**：SHA-384 ハッシュ関数が使用されるように指定します。 • **sha512**：SHA-512 ハッシュ関数が使用されるように指定します。
ステップ 13	**noshutdown** 例： `Device(cs-server)# no shutdown`	証明書サーバをイネーブルまたは再イネーブル化します。 これが下位の証明書サーバを初めてイネーブルにするときであれば、証明書サーバはキーを生成し、ルート証明書サーバから署名付き証明書を取得します。

例

証明書サーバがイネーブルにならない、あるいは証明書サーバが設定された要求を処理する際にトラブルが発生した場合は、**debugcryptopkiserver** コマンドを使用すると、次に示すように（「クロックが未設定」および「トラストポイントが未設定」）設定をトラブルシューティングできます。ここでは、「ms2」は 2048 ビットの RSA キー ペアのラベルを示します。

```
Router# debug crypto pki server
```

クロックが未設定

```
Router(config)# crypto pki server ms2
```

```
Router(cs-server)# mode sub-cs
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
*Jan  6 20:57:37.667: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
*Jan  6 20:57:45.303: CRYPTO_CS: starting enabling checks
*Jan  6 20:57:45.303: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
% Time has not been set. Cannot start the Certificate server
```

トラストポイントが未設定

```
Router(config)# crypto pki server ms2
Router(cs-server)# mode sub-cs
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key or type Return to exit
Password:
Jan  6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal no shut.
Jan  6 21:03:34.309: CRYPTO_CS: enter FSM: input state initial, input signal time set.
Jan  6 21:03:34.313: CRYPTO_CS: exit FSM: new state initial.
Jan  6 21:03:34.313: CRYPTO_CS: cs config has been unlocked
Re-enter password:
Jan  6 21:03:44.413: CRYPTO_CS: starting enabling checks
Jan  6 21:03:44.413: CRYPTO_CS: associated trust point 'sub' does not exist; generated
automatically
Jan  6 21:03:44.417: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan  6 21:04:03.993: CRYPTO_CS: nvram filesystem
Jan  6 21:04:04.077: CRYPTO_CS: serial number 0x1 written.
You must specify an enrollment URL for this CA before you can authenticate it.
% Failed to authenticate the Certificate Authority
```

証明書サーバが署名証明書をルート証明書サーバから取得できない場合は、次の例に示すように、**debugcryptopkitransactions** コマンドを使用して設定をトラブルシューティングできます。

```
Router# debug crypto pki transactions
Jan  6 21:07:00.311: CRYPTO_CS: enter FSM: input state initial, input signal time set
Jan  6 21:07:00.311: CRYPTO_CS: exit FSM: new state initial
Jan  6 21:07:00.311: CRYPTO_CS: cs config has been unlocked no sh
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan  6 21:07:03.535: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan  6 21:07:10.619: CRYPTO_CS: starting enabling checks
Jan  6 21:07:10.619: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan  6 21:07:20.535: %SSH-5-ENABLED: SSH 1.99 has been enabled
Jan  6 21:07:25.883: CRYPTO_CS: nvram filesystem
Jan  6 21:07:25.991: CRYPTO_CS: serial number 0x1 written.
Jan  6 21:07:27.863: CRYPTO_CS: created a new serial file.
Jan  6 21:07:27.863: CRYPTO_CS: authenticating the CA 'sub'
Jan  6 21:07:27.867: CRYPTO_PKI: Sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message=sub HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Cisco PKI)
Jan 6 21:07:27.867: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:27.871: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6 Certificate has the
following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2
% Do you accept this certificate? [yes/no]:
Jan  6 21:07:30.879: CRYPTO_PKI: http connection opened
Jan 6 21:07:30.903: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:30 GMT
Server: server-IOS
Content-Type: application/x-x509-ca-cert
Expires: Thu, 06 Jan 2005 21:07:30 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:30 GMT
Cache-Control: no-store, no-cache, must-revalidate
```

```
Pragma: no-cache
Accept-Ranges: none
Content-Type indicates we have received a CA certificate.
Jan  6 21:07:30.903: Received 507 bytes from server as CA certificate:
Jan  6 21:07:30.907: CRYPTO_PKI: transaction GetCACert completed
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.927: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()
Jan  6 21:07:30.927: CRYPTO_PKI: trustpoint sub authentication status = 0 y Trustpoint CA
certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
Router (cs-server)#
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:52.460: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan 6 21:07:54.348: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan 6 21:07:54.352: CRYPTO_CS: exit FSM: new state check failed
Jan 6 21:07:54.352: CRYPTO_CS: cs config has been locked
Jan 6 21:07:54.356: CRYPTO_PKI: transaction PKCSReq completed
Jan 6 21:07:54.356: CRYPTO_PKI: status:
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 1BA027DB 1C7860C7
EC188F65 64356C80
Jan 6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 840DB52C E17614CB
0C7BE187 0DFC884D D32CAA75
Jan 6 21:07:56.508: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:56.508: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:07:56.516: CRYPTO_PKI: http connection opened
Jan 6 21:07:59.136: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:07:57 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:07:57 GMT
  Last-Modified: Thu, 06 Jan 2005 21:07:57 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
Jan 6 21:07:59.324: The PKCS #7 message has 1 verified signers.
Jan 6 21:07:59.324: signing cert: issuer=cn=root1
Jan 6 21:07:59.324: Signed Attributes:
Jan 6 21:07:59.328: CRYPTO_PKI: status = 102: certificate request pending
Jan 6 21:08:00.788: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:08:00.788: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:08:00.796: CRYPTO_PKI: http connection opened
Jan 6 21:08:11.804: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:08:11.804: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:08:01 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:08:01 GMT
  Last-Modified: Thu, 06 Jan 2005 21:08:01 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
Jan  6 21:08:11.992: The PKCS #7 message has 1 verified signers.
Jan  6 21:08:11.992: signing cert: issuer=cn=root1
Jan  6 21:08:11.996: Signed Attributes:
Jan  6 21:08:11.996: CRYPTO_PKI: status = 102: certificate request pending
Jan  6 21:08:21.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:31.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:41.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:51.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:01.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial, 1
Jan  6 21:09:11.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial for session: 0
Jan  6 21:09:11.996: CRYPTO_PKI: can not resolve server name/IP address
Jan  6 21:09:11.996: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan  6 21:09:12.024: CRYPTO_PKI: http connection opened% Exporting Certificate Server signing
 certificate and keys...
Jan  6 21:09:14.784: CRYPTO_PKI: received msg of 1611 bytes
```

```
Jan  6 21:09:14.784: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:09:13 GMT
  Server: server-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:09:13 GMT
  Last-Modified: Thu, 06 Jan 2005 21:09:13 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none
Jan  6 21:09:14.972: The PKCS #7 message has 1 verified signers.
Jan  6 21:09:14.972: signing cert: issuer=cn=root1
Jan  6 21:09:14.972: Signed Attributes:
Jan  6 21:09:14.976: CRYPTO_PKI: status = 100: certificate is granted
Jan  6 21:09:15.668: The PKCS #7 message contains 1 certs and 0 crls.
Jan  6 21:09:15.688: Newly-issued Router Cert: issuer=cn=root serial=2
Jan  6 21:09:15.688: start date: 21:08:03 GMT Jan 6 2005
Jan  6 21:09:15.688: end date: 21:08:03 GMT Jan 6 2006
Jan  6 21:09:15.688: Router date: 21:09:15 GMT Jan 6 2005
Jan  6 21:09:15.692: Received router cert from CA
Jan  6 21:09:15.740: CRYPTO_CA: certificate not found
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
Jan  6 21:09:15.748: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 21:09:15.748: CRYPTO_CS: starting enabling checks
Jan  6 21:09:15.748: CRYPTO_CS: nvram filesystem
Jan  6 21:09:15.796: CRYPTO_CS: found existing serial file.
Jan  6 21:09:15.820: CRYPTO_CS: old router cert flag 0x4
Jan  6 21:09:15.820: CRYPTO_CS: new router cert flag 0x44
Jan  6 21:09:18.432: CRYPTO_CS: DB version 1
Jan  6 21:09:18.432: CRYPTO_CS: last issued serial number is 0x1
Jan  6 21:09:18.480: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 21:09:18.480: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 21:09:18.532: CRYPTO_CS: SCEP server started
Jan  6 21:09:18.532: CRYPTO_CS: exit FSM: new state enabled
Jan  6 21:09:18.536: CRYPTO_CS: cs config has been locked
Jan  6 21:09:18.536: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
```

証明書サーバがイネーブルにならない、あるいは証明書サーバが設定された要求を処理する際に問題が発生した場合は、**debugcryptopkiserver** コマンドを使用して、登録の進行状況をトラブルシューティングできます。このコマンドは、ルートCAをデバッグする場合にも使用できます（このコマンドは、ルート CA でオンにしてください）。

証明書サーバを RA モードで実行するように設定

Cisco IOS XE 証明書サーバは、Cisco IOS XE CA または別のサード パーティの CA の RA として機能することができます。サードパーティの CA を使用する場合は、**transparent** キーワード オプションに関する手順 8 の詳細を確認してください。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint** *name*
4. **enrollmenturl***url*
5. **subject-name***x.500-name*
6. **exit**
7. **cryptopkiserver***cs-label*
8. **modera** [**transparent**]
9. **auto-rollover** [*time-period*]
10. **grantautorollover** {**ca-cert** | **ra-cert**}
11. **noshutdown**
12. **noshutdown**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**cryptopkitrustpoint** *name* 例： `Device(config)# crypto pki trustpoint ra-server`	RA モード証明書サーバが使用するトラストポイントを宣言し、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	**enrollmenturl***url* 例： `Device(ca-trustpoint)# enrollment url http://ca-server.company.com`	発行元 CA 証明書サーバ（ルート証明書サーバ）の登録 URL を指定します。
ステップ 5	**subject-name***x.500-name* 例： `Device(ca-trustpoint)# subject-name cn=ioscs RA`	RA が使用する所有者名を指定します。 （注） 発行元CA証明書サーバがRAを認識できるように、所有者名に「cn=ioscs RA」または「ou=ioscs RA」を含めます（ステップ 7 を参照）。

	コマンドまたはアクション	目的
ステップ 6	**exit** 例： `Device(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 7	**cryptopkiserver***cs-label* 例： `Device(config)# crypto pki server ra-server`	Cisco IOS XE 証明書サーバをイネーブルにし、CS サーバ コンフィギュレーション モードを開始します。 （注） 証明書サーバには、上記ステップ 3 で作成されたトラストポイントと同じ名前を付ける必要があります。
ステップ 8	**modera** [**transparent**] 例： `Device(cs-server)# mode ra`	PKI サーバを RA 証明書サーバ モードにします。 RA モードの CA サーバが複数のタイプの CA サーバと相互運用できるようにするには、**transparent** キーワードを使用します。**transparent** キーワードを使用すると、元の PKCS#10 登録メッセージは再署名されず、変更せずに転送されます。この登録メッセージによって、IOS RA 証明書サーバは Microsoft CA サーバなどの CA サーバと連携します。
ステップ 9	**auto-rollover** [*time-period*] 例： `Device(cs-server)# auto-rollover 90`	（任意）自動 CA 証明書ロールオーバー機能をイネーブルにします。 • *time-period*：デフォルトは 30 日です。
ステップ 10	**grantautorollover** {**ca-cert** | **ra-cert**} 例： `Device(cs-server)# grant auto rollover ra-cert`	（任意）オペレータが介入せずに、下位の CA および RA モード CA の再登録要求を自動的に許可します。 • **ca-cert**：下位の CA ロールオーバー証明書が自動的に付与されるように指定します。 • **ra-cert**：RA モード CA ロールオーバー証明書が自動的に付与されるように指定します。 これが、初めて下位の証明書サーバをイネーブルにし、登録するときであれば、証明書要求を手動で許可する必要があります。
ステップ 11	**noshutdown** 例： `Device(cs-server)# no shutdown`	証明書サーバをイネーブルにします。 （注） このコマンドが発行されると、RA はルート証明書サーバに自動的に登録されます。RA 証明書が正常に受信されたら、**noshutdown** コマンドを再度発行する必要があります。これにより、証明書サーバが再イネーブル化されます。

	コマンドまたはアクション	目的
ステップ 12	**noshutdown** 例： `Device(cs-server)# no shutdown`	証明書サーバを再イネーブル化します。

RA モード証明書サーバに登録作業を委任するためのルート証明書サーバの設定

発行元証明書サーバを実行しているルータで、次のステップを実行します。具体的には、登録作業を RA モード証明書サーバに委任するルート証明書サーバを設定します。

（注） RA の登録要求を許可することは、本質的にクライアントデバイスの登録要求を許可するプロセスと同じですが、RA の登録要求が **cryptopkiserverinfo-requests** コマンドのコマンド出力の「RA certificate requests」セクションに表示されるという点が異なります。

手順の概要

1. **enable**
2. **cryptopkiserver***cs-label***inforequests**
3. **cryptopkiserver***cs-label***grant***req-id*
4. **configureterminal**
5. **cryptopkiserver***cs-label*
6. **grantra-auto**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**cryptopkiserver***cs-label***inforequests** 例： `Device# crypto pki server root-server info requests`	未処理の RA 証明書要求を表示します。 （注） このコマンドは、発行元証明書サーバを実行しているルータ上で発行されます。

	コマンドまたはアクション	目的
ステップ 3	**cryptopkiserver***cs-label***grant***req-id* 例： `Device# crypto pki server root-server grant 9`	保留の RA 証明書要求を許可します。 （注） 発行元証明書サーバが RA に登録要求の検証作業を委任するので、RA 証明書要求を許可する前に、RA 証明書要求に十分注意を払ってください。
ステップ 4	**configureterminal** 例： `Device# 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 5	**cryptopkiserver***cs-label* 例： `Device(config)# crypto pki server root-server`	Cisco IOS XE 証明書サーバをイネーブルにし、CS サーバ コンフィギュレーション モードを開始します。
ステップ 6	**grantra-auto** 例： `Device(cs-server)# grant ra-auto`	（任意）RA からのすべての登録要求が自動的に許可されるように指定します。 （注） **grant ra-auto** コマンドを機能させるには、RA 証明書の所有者名に「cn=ioscs RA」または「ou=ioscs RA」を含める必要があります（上記のステップ 2 を参照）。

次の作業

証明書サーバを設定したら、デフォルト値を使用するか、証明書サーバの機能用の CLI を使用して値を指定できます。デフォルト値以外の値を指定する場合は、「証明書サーバ機能の設定」の項を参照してください。

証明書サーバ機能の設定

証明書サーバをイネーブルにし、証明書サーバコンフィギュレーションモードになったら、次の作業のいずれかのステップを使用して、基本証明書サーバ機能の値（デフォルト値以外）を設定します。

証明書サーバのデフォルト値および推奨値

証明書サーバのデフォルト値は、比較的小規模のネットワーク（10 台程度のデバイス）に対処することを意図しています。たとえば、データベース設定値が最小に設定されている場合（**database level minimal** コマンドによって）、証明書サーバは SCEP を使用してすべての CRL 要求を処理します。大規模なネットワークでは、考えられる監査および失効目的のためにデータベース設定

「names」または「complete」（**database level** コマンドで示されるように）を使用することを推奨します。さらに大規模なネットワークでは、CRL 確認ポリシーに応じて、外部CDPを使用する必要があります。

証明書サーバ ファイルの保管および公開場所

ファイル タイプをさまざまな保管場所に保管し、さまざまな公開場所で公開できる柔軟性が備わっています。

手順の概要

1. **databaseurl***root-url*
2. **databaseurl**{**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url*
3. **databaseurl** {**cnm** | **crl** | **crt**} **publish***root-url*
4. **databaselevel** {**minimal** | **names** | **complete**}
5. **databaseusernameusername** [**password** [*encr-type*] *password*]
6. **databasearchive** {**pkcs12** | **pem**}[**password***encr-type*] *password*]
7. **issuer-name***DN-string*
8. **lifetime** {**ca-certificate** | **certificate**} *time*
9. **lifetimecrl***time*
10. **lifetimeenrollment-request***time*
11. **cdp-url***url*
12. **noshutdown**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**databaseurl***root-url* 例： `Device(cs-server)# database url tftp://cert-svr-db.company.com`	証明書サーバのデータベース エントリが書き出されるプライマリ ロケーションを指定します。 このコマンドが指定されていない場合、すべてのデータベース エントリは NVRAM に書き込まれます。
ステップ 2	**databaseurl**{**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url* 例： `Device(cs-server)# database url ser nvram:`	証明書サーバの重要なファイルの保管場所をファイル タイプ別に指定します。 （注） このコマンドが指定されていないと、すべての重要ファイルは、（指定されている場合）プライマリ ロケーションに保管されます。プライマリ ロケーションが指定されてない場合は、すべての重要ファイルが NVRAM に保管されます。

	コマンドまたはアクション	目的
ステップ 3	**databaseurl** {**cnm** | **crl** | **crt**} **publish***root-url* 例： `Device(cs-server)# database url crl publish tftp://csdb_specific_crl_files.company.com`	証明書サーバの公開場所をファイルタイプ別に指定します。 （注） このコマンドが指定されていないと、すべての公開ファイルは、（指定されている場合）プライマリ ロケーションに保管されます。プライマリ ロケーションが指定されてない場合は、すべての公開ファイルが NVRAM に保管されます。
ステップ 4	**databaselevel** {**minimal** | **names** | **complete**} 例： `Device(cs-server)# database level complete`	証明書登録データベースに保管されるデータのタイプを制御します。 • **minimal**：新しい証明書を、継続して問題なく発行できる程度の情報が保管されます。これがデフォルト値です。 • **names**：minimal レベルで提供される情報以外に、各証明書のシリアル番号および所有者名を保存します。 • **complete**：minimal レベルおよび names レベルで提供される情報以外に、発行済みの各証明書がデータベースに書き込まれます。 （注） **complete** キーワードを指定すると、大量の情報が生成されます。このキーワードを発行する場合、**databaseurl** コマンドを使用して、データを保管する外部 TFTP サーバも指定する必要があります。
ステップ 5	**databaseusername***username* [**password** [*encr-type*] *password*] 例： `Device(cs-server)# database username user password PASSWORD`	（任意）プライマリ証明書登録データベースの保管場所にアクセスする必要がある場合、ユーザ名とパスワードを設定します。
ステップ 6	**databasearchive** {**pkcs12** | **pem**}[**password***encr-type*] *password*] 例： `Device(cs-server)# database archive pem`	（任意）ファイルを暗号化するための CA キーと CA 証明書のアーカイブ形式およびパスワードを設定します。 デフォルト値は **pkcs12** です。したがって、このサブコマンドが設定されていなくても、自動アーカイブが引き続き実行され、PKCS12 形式が使用されます。 • パスワードの設定は任意です。パスワードが設定されていない場合、サーバを初めて起動したときに、パスワードの入力を求めるプロンプトが表示されます。 （注） アーカイブが完了したら、設定からパスワードを削除することを推奨します。

	コマンドまたはアクション	目的
ステップ 7	**issuer-name***DN-string* 例： `Device(cs-server)# issuer-name my-server`	（任意）指定した識別名（*DN-string*）に CA 発行者名を設定します。デフォルト値は **issuer-namecn**={*cs-label* } です。
ステップ 8	**lifetime** {**ca-certificate** | **certificate**} *time* 例： `Device(cs-server)# lifetime certificate 888`	（任意）CA 証明書または証明書のライフタイム（日数）を指定します。 有効な値の範囲は、1 ～ 1825 日です。CA 証明書のデフォルトのライフタイムは3年、証明書のデフォルトのライフタイムは 1 年です。証明書の最大のライフタイムは、CA 証明書のライフタイムより 1 カ月短い日数です。
ステップ 9	**lifetimecrl***time* 例： `Device(cs-server)# lifetime crl 333`	（任意）証明書サーバが使用する CRL のライフタイム（時間単位）を定義します。 最大ライフタイム値は 336 時間（2 週間）です。デフォルト値は 168 時間（1 週間）です。
ステップ 10	**lifetimeenrollment-request***time* 例： `Device(cs-server)# lifetime enrollment-request 888`	（任意）登録要求が削除されるまで、登録データベースに保管される期間を指定します。 最大ライフタイムは 1000 時間です。
ステップ 11	**cdp-url***url* 例： `Device(cs-server)# cdp-url http://my-cdp.company.com`	（任意）証明書サーバが発行した証明書で使用される CDP の場所を定義します。 • URL は、HTTP URL を使用する必要があります。 Cisco IOS ソフトウェアを実行せず、また SCEP GetCRL 要求をサポートしない PKI クライアントの場合は、次の URL 形式を使用します。 http://server.company.com/certEnroll/filename.crl また、Cisco IOS 証明書サーバが CDP としても設定されている場合は、次の URL 形式を使用します。 http://cs-addr/cgi-bin/pkiclient.exe?operation=GetCRL この *cs-addr* は証明書サーバの場所です。 指定された場所内に埋め込まれた疑問符を保持するようパーサーに強制するには、疑問符の前に Ctrl+V キーを入力します。この処理を実行しないと、HTTP による CRL 取得でエラー メッセージが返されます。 （注） このコマンドは任意ですが、すべての展開シナリオで使用することをぜひ推奨します。

	コマンドまたはアクション	目的
ステップ 12	**noshutdown** 例： `Device(cs-server)# no shutdown`	証明書サーバをイネーブルにします。 このコマンドは、証明書サーバの設定が完了した後で発行する必要があります。

例

次の例では、PKI クライアントが SCEP GetCRL 要求をサポートしない CDP の場所を設定する方法を示します。

```
Device(config)# crypto pki server aaa
Device(cs-server)# database level minimum
Device(cs-server)# database url tftp://10.1.1.1/username1/
Device(cs-server)# issuer-name CN=aaa
Device(cs-server)# cdp-url http://server.company.com/certEnroll/aaa.crl
```

証明書サーバがルータ上でイネーブルになってから、**showcryptopkiserver** コマンドを実行すると、次の出力が表示されます。

```
Device# show crypto pki server

    Certificate Server status:enabled, configured
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:19:31:15 PST Nov 17 2006
    CRL NextUpdate timer:19:31:15 PST Nov 25 2003
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
```

自動 CA 証明書ロールオーバーでの作業

自動 CA 証明書ロールオーバーをただちに開始する

ルート CA サーバ上で自動 CA 証明書ロールオーバー プロセスをただちに開始するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkiserver***cs-label***rollover** [**cancel**]

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**cryptopkiserver***cs-label***rollover** [**cancel**] 例： `Device(config)# crypto pki server mycs rollover`	シャドウ CA 証明書を生成して、CA 証明書ロールオーバープロセスをただちに開始します。 CA 証明書ロールオーバー証明書およびキーを削除するには、**cancel** キーワードを使用します。

証明書サーバ クライアントのロールオーバー証明書の要求

証明書サーバ クライアントのロールオーバー証明書を要求するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkiserver***cs-label***rolloverrequestpkcs10terminal**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**cryptopkiserver***cs-label***rolloverrequestpkcs10terminal** 例： `Device(config)# crypto pki server mycs rollover request pkcs10 terminal`	サーバからクライアント ロールオーバー証明書を要求します。

例

次は、サーバに入力されるロールオーバー証明書要求の例です。

```
Device# crypto pki server mycs rollover request pkcs10 terminal

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE REQUEST-----
MIIBUTCBuwIBADASMRAwDgYDVQQDEwdOZXdSb290MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDMHeev1ERSs320zbLQQk+3lhV/R2HpYQ/iM6uT1jkJf5iy0UPR
wF/X16yUNmG+ObiGiW9fsASF0nxZw+fO7d2X2yh1PakfvF2wbP27C/sgJNOw9uPf
sBxEc40Xe0d5FMh0YKOSAShfZYKOflnyQR2Drmm2x/33QGol5QyRvjkeWQIDAQAB
oAAwDQYJKoZIhvcNAQEEBQADgYEALM90r4d79X6vxhD0qjuYJXfBCOvv4FNyFsjr
aBS/y6CnNVYySF8UBUohXYIGTWf4I4+sj6i8gYfoFUW1/L82djS18TLrUr6wpCOs
RqfAfps7HW1e4cizOfjAUU+C7lNcobCAhwF1o6q2nIEjpQ/2yfK9O7sb3SCJZBfe
eW3tyCo=
-----END CERTIFICATE REQUEST-----
```

CA ロールオーバー証明書のエクスポート

CA ロールオーバー証明書をエクスポートするには、次の作業を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkiexport***trustpoint***pem** {**terminal** | **url***url*} [**rollover**]

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**cryptopkiexport***trustpoint***pem** {**terminal** | **url***url*} [**rollover**] 例： `Device(config)# crypto pki export mycs pem terminal rollover`	CA シャドウ証明書をエクスポートします。

証明書サーバ、証明書、CAの保守、検証、およびトラブルシューティング

登録要求データベースの管理

SCEPは、2つのクライアント認証メカニズム（手動による登録と事前共有キーを使用する登録）をサポートします。手動による登録では、管理者は、CAサーバで具体的に登録要求を認可する必要があります。事前共有キーを使用する登録では、管理者は、ワンタイム パスワード（OTP）を生成することにより、登録要求を事前に許可できます。

次の作業のうち、いずれかのステップを使用して、SCEP で使用される登録処理パラメータの指定、および実行時動作または証明書サーバの制御などの機能を実行すると、登録要求データベースが管理しやすくなります。

手順の概要

1. **enable**
2. **cryptopkiserver***cs-label***grant** {**all** | *req-id*}
3. **cryptopkiserver***cs-label***reject** {**all** | *req-id*}
4. **cryptopkiserver***cs-label***passwordgenerate***minutes*
5. **cryptopkiserver***cs-label***revoke***certificate-serial-number*
6. **cryptopkiserver***cs-label***requestpkcs10** {*url* | **terminal**} [**base64**| **pem**
7. **cryptopkiserver***cs-label***infocrl**
8. **cryptopkiserver***cs-label***inforequests**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**cryptopkiserver***cs-label***grant** {**all** | *req-id*} 例： `Device# crypto pki server mycs grant all`	すべての SCEP 要求または特定の SCEP 要求を許可します。
ステップ 3	**cryptopkiserver***cs-label***reject** {**all** | *req-id*} 例： `Device# crypto pki server mycs reject all`	すべての SCEP 要求または特定の SCEP 要求を拒否します。
ステップ 4	**cryptopkiserver***cs-label***passwordgenerate***minutes* 例： `Device# crypto pki server mycs password generate 75`	SCEP 要求に対して OTP を生成します。 • *minutes*：パスワードの有効時間（分）。有効な値の範囲は、1 ～ 1440 分です。デフォルトは 60 分です。 （注） 有効になる OTP は、一度に 1 つだけです。別の OTP が生成されると、1 番目の OTP は無効になります。

ステップ 5	**cryptopkiserver***cs-label***revoke***certificate-serial-number*
	
	例：
	`Device# crypto pki server mycs revoke 3`
		証明書を証明書のシリアル番号に基づいて無効にします。
		• *certificate-serial-number*--One of the following options:
		  • 0x で始まるストリング。これは 16 進値として処理されます
		  • 0 と no x で始まるストリング。これは 8 進値として処理されます
		  • その他すべてのストリング。これらは 10 進値として処理されます

ステップ 6	**cryptopkiserver***cs-label***requestpkcs10** {*url* | **terminal**} [**base64**| **pem** 例： `Device# crypto pki server mycs request pkcs10 terminal pem`	Base 64 符号化形式または PEM 形式の PKCS10 証明書登録要求を要求データベースに手動で追加します。 証明書が付与されると、証明書は Base 64 符号化を使用してコンソール端末に表示されます。

	コマンドまたはアクション	目的
		• **pem**：要求に PEM ヘッダーが使用されたかどうかにかかわらず、証明書を付与された後、PEM ヘッダーを自動的に追加した証明書を返すように指定します。 • **base64**：要求に PEM ヘッダーが使用されたかどうかにかかわらず、証明書を PEM ヘッダーなしで返すように指定します。
ステップ 7	**cryptopkiserver***cs-label***infocrl** 例： `Device# crypto pki server mycs info crl`	現在の CRL のステータスに関する情報を表示します。
ステップ 8	**cryptopkiserver***cs-label***inforequests** 例： `Device# crypto pki server mycs info requests`	未処理の証明書登録要求をすべて表示します。

登録要求データベースからの要求の削除

証明書サーバは、登録要求を受け取ると、要求を保留状態のままにする、拒否するか、あるいは許可できます。要求は、クライアントが要求の結果を求めて証明書サーバをポーリングするまで、登録要求データベースに1週間保存されます。クライアントが終了し、証明書サーバを絶対にポーリングしない場合は、個々の要求またはすべての要求をデータベースから削除できます。

次の作業を実行して、データベースから要求を削除し、キーおよびトランザクションIDに関してサーバをクリーンな状態に戻せます。また、この作業を実行して、適切に動作しない SCEP クライアントのトラブルシューティングができます。

手順の概要

1. **enable**
2. **cryptopkiserver***cs-label***remove**{**all** | *req-id*}

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**cryptopkiserver***cs-label***remove**{**all** | *req-id*} 例： `Device# crypto pki server mycs remove 15`	登録要求を登録要求データベースから削除します。

証明書サーバの削除

証明書サーバを PKI 設定に残したくない場合、証明書サーバを PKI 設定から削除できます。通常、下位の証明書サーバまたは RA は削除されます。ただし、保存された RSA キーを使用してルート証明書サーバを別のデバイスに移動した場合は、ルート証明書サーバを削除できます。

PKI 設定から証明書サーバを削除するには、次の作業を実行します。

（注） 証明書サーバを削除すると、関連付けられているトラストポイントおよびキーも削除されます。

手順の概要

1. **enable**
2. **configureterminal**
3. **nocryptopkiserver***cs-label*

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# 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	**nocryptopkiserver***cs-label* 例： `Device(config)# no crypto pki server mycs`	証明書サーバおよび関連付けられたトラストポイントとキーを削除します。

証明書サーバと CA ステータスの検証およびトラブルシューティング

証明書サーバまたは CA のステータスを検証するには、次の手順のいずれかを使用します。

手順の概要

1. **enable**
2. **debugcryptopkiserver**
3. **dir***filesystem*:

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debugcryptopkiserver** 例： `Device# debug crypto pki server`	暗号 PKI 証明書サーバのデバッグをイネーブルにします。 • 証明書サーバが応答しない場合、あるいは証明書サーバが設定された要求を処理する際に問題が発生した場合は、このコマンドを使用して登録の進行状況のモニタリングおよびトラブルシューティングができます。
ステップ 3	**dir***filesystem*: 例： `Device# dir slot0:`	ファイル システムのファイル リストを表示します。 • ローカル ファイル システムをポイントするために **databaseurl** コマンドを入力した場合は、このコマンドを使用して、証明書サーバ自動アーカイブファイルを検証できます。少なくともデータベー

	コマンドまたはアクション	目的
		ス内の「*cs-label* .ser」および「*cs-label* .crl」ファイルを参照できる必要があります。

CA 証明書情報の検証

CA 証明書に関連する情報（証明書サーバ ロールオーバー プロセス、ロールオーバー証明書、およびタイマーなど）を入手するには、次のコマンドのいずれかを使用します。

（注） これらのコマンドは、シャドウ証明書情報に対して排他的ではありません。シャドウ証明書が存在しない場合、次のコマンドを実行すると、アクティブな証明書情報だけが表示されます。

手順の概要

1. **cryptopkicertificatechain**
2. **cryptopkiserverinforequests**
3. **showcryptopkicertificates**
4. **showcryptopkiserver**
5. **showcryptopkitrustpoints**

手順の詳細

ステップ 1 **cryptopkicertificatechain**

例：
```
Device(config)# crypto pki certificate chain mica

certificate 06
certificate ca 01
! This is the peer’s shadow PKI certificate.
certificate rollover 0B
! This is the CA shadow PKI certificate
certificate rollover ca 0A
```

証明書チェーンの詳細を表示し、現在のアクティブな証明書と証明書チェーンのロールオーバー証明書を区別します。次の例では、アクティブな CA 証明書を持つ証明書チェーンおよびシャドウ証明書、またはロールオーバー証明書を示します。

ステップ 2 **cryptopkiserverinforequests**

例：
```
Device# crypto pki server myca info requests
```

```
Enrollment Request Database:
RA certificate requests:
  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
RA rollover certificate requests:
  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------
Router certificates requests:
  ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
1      pending    A426AF07FE3A4BB69062E0E47198E5BF hostname=client
  Router rollover certificates requests:
  ReqID  State      Fingerprint                      SubjectName
  --------------------------------------------------------------
  2      pending    B69062E0E47198E5BFA426AF07FE3A4B hostname=client
```

未処理の証明書登録要求をすべて表示します。次に、シャドウPKI証明書情報要求の出力例を示します。

ステップ 3 **showcryptopkicertificates**

例：

```
Device# show crypto pki certificates

Certificate
  Subject Name
    Name: myrouter.example.com
    IP Address: 192.0.2.1
    Serial Number: 04806682
  Status: Pending
  Key Usage: General Purpose
    Fingerprint: 428125BD A3419600 3F6C7831 6CD8FA95 00000000
CA Certificate
  Status: Available
  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F
  Key Usage: Not Set
```

証明書、認証局証明書、シャドウ証明書、および任意の登録認局証明書に関する情報を表示します。次の例では、ルータの証明書および CA の証明書を表示します。利用可能なシャドウ証明書はありません。単一の汎用目的 RSA キー ペアが以前に生成されていましたが、このキー ペアについては、証明書が要求されているものの、受信されていません。ルータの証明書のステータスが「Pending」であることに注意してください。ルータが CA からその証明書を受信すると、**show** 出力の [Status] フィールドが「Available」に変わります。

ステップ 4 **showcryptopkiserver**

例：

```
Device# show crypto pki server

Certificate Server routercs:
   Status: enabled, configured
   Issuer name: CN=walnutcs
   CA cert fingerprint: 800F5944 74337E5B C2DF6C52 9A7B1BDB
   Granting mode is: auto
   Last certificate issued serial number: 0x7
   CA certificate expiration timer: 22:10:29 GMT Jan 29 2007
   CRL NextUpdate timer: 21:50:56 GMT Mar 5 2004
   Current storage dir: nvram:
   Database Level: Minimum - no cert data written to storage
Rollover status: available for rollover
    Rollover CA cert fingerprint: 6AAF5944 74227A5B 23DF3E52 9A7F1FEF
    Rollover CA certificate expiration timer: 22:10:29 GMT Jan 29 2017
```

証明書サーバの現在の状態および設定を表示します。次の例では、証明書サーバ「routercs」にロールオーバーが設定されていることを示します。CA 自動ロールオーバー時間が発生し、ロールオーバーまたはシャドウ証明書、PKI 証明書が利用可能です。ステータスには、ロールオーバー証明書フィンガープリントおよびロールオーバー CA 証明書の失効タイマー情報が示されています。

ステップ 5 **showcryptopkitrustpoints**

例：

```
Device# show crypto pki trustpoints

Trustpoint vpn:
   Subject Name:
   cn=Cisco SSL CA
   o=Cisco Systems
   Serial Number: 0FFEBBDC1B6F6D9D0EA7875875E4C695
   Certificate configured.
   Rollover certificate configured.
   Enrollment Protocol:
   SCEPv1, PKI Rollover
```

デバイスに設定されているトラストポイントを表示します。次の出力は、シャドウ CA 証明書が使用可能であることを示し、最後の登録操作中に報告された SCEP 機能を示します。

証明書サーバを使用するための設定例

例：特定の保管および公開場所の設定

次の例では、証明書サーバが迅速に証明書要求に応答できるように、最低限のローカル ファイル システムの設定を示します。.ser および .crl ファイルは、素早くアクセスできるようにローカルの Cisco IOS XE システムの上に保管され、長時間のロギングでは、.crt ファイルのすべてのコピーがリモートの場所に公開されます。

```
crypto pki server myserver
     !Pick your database level.
     database level minimum
     !Specify a location for the .crt files that is different than the default local
!Cisco IOS file system.
     database url crt publish http://url username user1 password secret
```

（注） .crl ファイルが非常に大きくなる場合に備えて、ローカル ファイル システムの空き容量をモニタリングする必要があります。

次の例では、重要ファイルのプライマリ保管場所、重要ファイルのシリアル番号ファイル固有の保管場所、メイン証明書サーバのデータベース ファイル、および CRL ファイルのパスワード保護されたファイル公開場所の設定を示します。

```
Device(config)# crypto pki server mycs
```

```
Device(cs-server)# database url ftp://cs-db.company.com

!
% Server database url was changed. You need to move the
% existing database to the new location.
!
Device(cs-server)# database url ser nvram:
Device(cs-server)# database url crl publish ftp://crl.company.com username myname password
 mypassword
Device(cs-server)# end
```

次の出力は、指定されたプライマリ保管場所および指定された重要ファイルの保管場所を示します。

```
Device# show

Sep  3 20:19:34.216: %SYS-5-CONFIG_I: Configured from console by user on console
Device# show crypto pki server

Certificate Server mycs:
     Status: disabled
     Server's configuration is unlocked  (enter "no shut" to lock it)
     Issuer name: CN=mycs
     CA cert fingerprint: -Not found-
     Granting mode is: manual
     Last certificate issued serial number: 0x0
     CA certificate expiration timer: 00:00:00 GMT Jan 1 1970
     CRL not present.
     Current primary storage dir: ftp://cs-db.company.com
     Current storage dir for .ser files: nvram:
    Database Level: Minimum - no cert data written to storage The following output displays
 all storage and publication locations. The serial number file (.ser) is stored in NVRAM.
The CRL file will be published to ftp://crl.company.com with a username and password. All
other critical files will be stored to the primary location, ftp://cs-db.company.com.

Device# show running-config

   section crypto pki server
   crypto pki server mycs shutdown database url ftp://cs-db.company.com
   database url crl publish ftp://crl.company.com username myname password 7
12141C0713181F13253920
   database url ser nvram:
Device#
```

例：登録要求データベースからの登録要求の削除

次の例では、現在登録要求データベース内にある両方の登録要求と、これらの登録要求のうち 1 つがデータベースから削除された結果を示します。

例：現在登録要求データベース内にある登録要求

次の例では、現在登録要求データベース内にある登録要求を表示するために、**cryptopkiserverinforequests** コマンドが使用されたことを示します。

```
Device# crypto pki server myserver info requests

Enrollment Request Database:
RA certificate requests:
ReqID    State      Fingerprint                              SubjectName
--------------------------------------------------------------------------
Router certificates requests:
ReqID    State      Fingerprint                              SubjectName
--------------------------------------------------------------------------
2        pending    1B07F3021DAAB0F19F35DA25D01D8567      hostname=host1.company.com
1        denied     5322459D2DC70B3F8EF3D03A795CF636      hostname=host2.company.com
```

例：crypto pki server remove コマンドを使用して 1 つの登録要求を削除する

次の例では、**cryptopkiserverremove** コマンドを使用して、登録要求 1 が削除されたことを示します。

```
Device# crypto pki server myserver remove 1
```

例：登録要求を 1 つ削除した後の登録要求データベース

次の例では、登録要求データベースから登録要求 1 を削除した結果を示します。

```
Device# crypto pki server mycs info requests

Enrollment Request Database:
RA certificate requests:
ReqID    State     Fingerprint                         SubjectName
--------------------------------------------------------------------
Router certificates requests:
ReqID    State     Fingerprint                         SubjectName
--------------------------------------------------------------------
2        pending   1B07F3021DAAB0F19F35DA25D01D8567   hostname=host1.company.com
```

例：証明書サーバのルート キーの自動アーカイブ化

次の出力設定および例では、**databasearchive** コマンドを設定していない（つまりデフォルト値を使用して設定した）場合、パスワードを設定せずに **databasearchive** コマンドを設定して CA 証明書および CA キー アーカイブ形式を PEM にする場合、およびパスワードを設定して **databasearchive** コマンドを設定し、CA 証明書および CA キー アーカイブ形式を PKCS12 にする場合の表示内容を示します。最後の例は、PEM 形式のアーカイブ ファイルのサンプル内容です。次の例の「ms2」は 2048 ビット キー ペアのラベルを示します。

例：database archive コマンド未設定

（注） デフォルトは PKCS12 です。**noshutdown** コマンドを発行すると、パスワードの入力を求めるプロンプトが表示されます。

```
Device(config)# crypto pki server ms2
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Device(cs-server)# end
Device# dir nvram:

Directory of nvram:/
  125  -rw-        1693              <no date>  startup-config
  126  ----           5              <no date>  private-config
    1  -rw-          32              <no date>  myserver.ser
    2  -rw-         214              <no date>  myserver.crl
```

```
! Note the next line, which indicates PKCS12 format.
    3  -rw-        1499              <no date>  myserver.p12
```

例：database archive コマンドおよび pem キーワードを設定

（注） **noshutdown** コマンドを発行すると、パスワードの入力を求めるプロンプトが表示されます。

```
Device(config)# crypto pki server ms2
Device(cs-server)# database archive pem
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
!Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Device(cs-server)# end
Device# dir nvram

Directory of nvram:/
  125  -rw-        1693            <no date>  startup-config
  126  ----           5            <no date>  private-config
    1  -rw-          32            <no date>  myserver.ser
    2  -rw-         214            <no date>  myserver.crl
! Note the next line showing that the format is PEM.
    3  -rw-        1705            <no date>  myserver.pem
```

例：database archive コマンドおよび pkcs12 キーワード（およびパスワード）を設定

（注） パスワードは、入力されると暗号化されます。ただし、アーカイブが完了したら、設定からパスワードを削除することを推奨します。

```
Device(config)# crypto pki server ms2
Device(cs-server)# database archive pkcs12 password cisco123
Device(cs-server)# no shutdown

% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note that you are not being prompted for a password.
% Certificate Server enabled.
Device(cs-server)# end
Device# dir nvram:

Directory of nvram:/
   125   -rw-         1693             <no date>   startup-config
   126   ----            5             <no date>   private-config
     1   -rw-           32             <no date>   myserver.ser
     2   -rw-          214             <no date>   myserver.crl
! Note that the next line indicates that the format is PKCS12.
     3   -rw-         1499             <no date>   myserver.p12
```

例：PEM フォーマットのアーカイブ

次のサンプル出力は、自動アーカイブがPEMファイル形式で設定されたことを示します。アーカイブは、CA証明書とCA秘密キーから成ります。バックアップを使用して証明書サーバを復元するには、PEM 形式の CA 証明書と CA キーを別々にインポートする必要があります。

（注） CA 証明書および CA キー アーカイブ ファイル以外にも、シリアル番号ファイル（.ser）およびCRL ファイル（.crl）を定期的にバックアップする必要があります。証明書サーバを復元する必要がある場合、CA 運用においてシリアル ファイルおよび CRL ファイルは重要です。

```
Device# more nvram:mycs.pem

-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDgyNzAyMzI0NloXDTA3MDgyNzAyMzI0NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1lZpKP4nGDJHgPkpYSkix7lD
nr23aMlZ9Kz5oo/qTBxeZ8mujpjYcZ0T8AZvoOiCuDnYMl796ZwpkMgjz1aZZbL+
BtuVvllsEOfhC+u/Ol/vxfGG5xpshoz/F5J3xdg5ZZuWWuIDAUYu9+QbI5feuG04
Z/BiPIb4AmGTP4B2MM0CAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUKi/cuK6wkz+ZswVtb06vUJboEeEwHQYDVR0O
BBYEFCov3LiusJM/mbMFbW9Or1CW6BHhMA0GCSqGSIb3DQEBBAUAA4GBAKLOmoE2
4+NeOKEXMCXG1jcohK7O2HrkFfl/vpK0+q92PTnMUFhxLOqI8pWIq5CCgC7heace
OrTv2zcUAoH4rzx3Rc2USIxkDokWWQMLujsMm/SLIeHit0G5uj//GCcbgK20MAW6
ymf7+TmblSFljWzstoUXC2hLnsJIMq/KffaD
-----END CERTIFICATE-----

!The private key is protected by the password that is
configured in "database archive pem password pwd" or that
is entered when you are prompted for the password.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,106CE91FFD0A075E

zyiFC8rKv8Cs+IKsQG2QpsVpvDBHqZqBSM4D528bvZv7jzr6WuHj8E6zO+6G8R/A
zjsfTALo+e+ZDg7KMzbryHARvjskbqFdOMLlVIYBhCeSElKsskWB6chOuyPHJInW
JwC5YzZdZwOqcyLBP/xOYXcvjzzNfPAXZzN12VR8vWDNq/kHT+3Lplc8hY++ABMI
M+C9FB3dpNZzu5O1BZCJg46bqbkulaCCmScIDaVt0zDFZwWTSufiemmNxZBG4xS8
t5t+FEhmSfv8DAmwg4f/KVRFTm10phUArcLxQO38Al0W5YHHORdACnuzVUvHgco7
VT4XUTjO7qMhmJgFNWy1pu49fbdS2NnOn5IoiyAq5lk1KUPrz/WABWiCvLMylGnZ
kyMCWoaMtgS/vdx74BBCj09yRZJnLMlIi6SDofjCNTDHfmFEVg4LsSWCd4lP9OP8
0MqhP1D5VIx6PbMNwkWW12lpBbCCdesFRGHjZD2dOu96kHD7ItErx34CC8W04aG4
b7DLktUu6WNV6M8g3CAqJiC0V8ATlp+kvdHZVkXovgND5IU0OJpsj0HhGzKAGpOY
KTGTUekUboISjVVkI6efp1vO6temVL3Txg3KGhzWMJGrq1snghE0KnV8tkddv/9N
d/t1l+we9mrccTq50WNDnkEi/cwHI/0PKXg+NDNH3k3QGpAprsqGQmMPdqc5ut0P
86i4cF9078QwWg4Tpay3uqNH1Zz6UN0tcarVVNmDupFESUxYw10qJrrEYVRadu74
rKAU4Ey4xkAftB2kuqvr21Av/L+jne4kkGIoZYdB+p/M98pQRgkYyg==
-----END RSA PRIVATE KEY-----
```

例：証明書サーバ バックアップ ファイルからの証明書サーバの復元

次の例は、PKCS12 アーカイブから復元され、データベース URL が NVRAM（デフォルト）であることを示します。

```
Device# copy tftp://192.0.2.71/backup.ser nvram:mycs.ser

Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)

Device# copy tftp://192.0.2.71/backup.crl nvram:mycs.crl
```

```
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)

Device# configure terminal
Device(config)# crypto pki import mycs pkcs12 tftp://192.0.2.71/backup.p12 cisco123

Source filename [backup.p12]?
CRYPTO_PKI: Imported PKCS12 file successfully.

Device(config)# crypto pki server mycs
! fill in any certificate server configuration here

Device(cs-server)# no shutdown
% Certificate Server enabled.

Device(cs-server)# end
Device# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 34885330 B13EAD45 196DA461 B43E813F
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 01:49:13 GMT Aug 28 2007
    CRL NextUpdate timer: 01:49:16 GMT Sep 4 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

次の例は、PEM アーカイブから復元され、データベース URL が flash であることを示します。

```
Device# copy tftp://192.0.2.71/backup.ser flash:mycs.ser

Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://192.0.2.71/backup.crl flash:mycs.crl
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)
Device# configure terminal

! Because CA cert has Digital Signature usage, you need to import using the "usage-keys"
keyword

Device(config)# crypto ca import mycs pem usage-keys terminal cisco123
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----
% Enter PEM-formatted encrypted private SIGNATURE key.
% End with "quit" on a line by itself.
! Paste the CA private key from .pem archive.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,5053DC842B04612A

1CnlF5Pqvd0zp2NLZ7iosxzTy6nDeXPpNyJpxB5q+V29IuY8Apb6TlJCU7YrsEB/
nBTK7K76DCeGPlLpcuyEI171QmkQJ2gA0QhC0LrRo09WrINVH+b4So/y7nffZkVb
p2yDpZwqoJ8cmRH94Tie0YmzBtEh6ayOud11z53qbrsCnfSEwszt1xrW1MKrFZrk
/fTy6loHzGFzl3BDj4r5gBecExwcPp74ldHO+Ld4Nc9egG8BYkeBCsZZOQNVhXLN
```

```
I0tODOs6hP915zb6OrZFYv0NK6grTBO9D8hjNZ3U79jJzsSP7UNzIYHNTzRJiAyu
i56Oy/iHvkCSNUIK6zeIJQnW4bSoM1BqrbVPwHU6QaXUqlNzZ8SDtw7ZRZ/rHuiD
RTJMPbKquAzeuBss1132OaAUJRStjPXgyZTUbc+cWb6zATNws2yijPDTR6sRHoQL
47wHMr2Yj80VZGgkCSLAkL88ACz9TfUiVFhtfl6xMC2yuFl+WRk1XfF5VtWe5Zer
3Fn1DcBmlF7O86XUkiSHP4EV0cI6n5ZMzVLx0XAUtdAllgD94y1V+6p9PcQHLyQA
pGRmj5IlSFw90aLafgCTbRbmC0ChIqHy91UFa1ub0130+yu7LsLGRlPmJ9NE61JR
bjRhlUXItRYWY7C4M3m/0wz6fmVQNSumJM08RHq6lUB3olzIgGIZlZkoaESrLG0p
qq2AENFemCPF0uhyVS2humMHjWuRr+jedfc/IMl7sLEgAdqCVCfV3RZVEaNXBud1
4QjkuTrwaTcRXVFbtrVioT/puyVUlpA7+k7w+F5TZwUV08mwvUEqDw==
-----END RSA PRIVATE KEY-----
quit
% Enter PEM-formatted SIGNATURE certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive again.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private ENCRYPTION key.
% End with "quit" on a line by itself.
! Because the CA cert only has Digital Signature usage, skip the encryption part.
quit
% PEM files import succeeded.
Device(config)# crypto pki server mycs
Device(cs-server)# database url flash:

! Fill in any certificate server configuration here.
Device(cs-server)# no shutdown

% Certificate Server enabled.
Device(cs-server)# end
Device# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: F04C2B75 E0243FBC 19806219 B1D77412
    Granting mode is: manual
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 21:02:55 GMT Sep 2 2007
    CRL NextUpdate timer: 21:02:58 GMT Sep 9 2004
    Current storage dir: flash:
    Database Level: Minimum - no cert data written to storage
```

例：下位証明書サーバ

次の設定および出力は、下位の証明書サーバを設定した後で、通常表示されるものです。「ms2」は前述の手順で生成した 2048 ビット RSA キーを表します。

```
Device(config)# crypto pki trustpoint sub
Device(ca-trustpoint)# enrollment url http://192.0.2.6
Device(ca-trustpoint)# rsa keypair ms2 2048
Device(ca-trustpoint)# exit
Device(config)# crypto pki server sub
Device(cs-server)# mode sub-cs
Device(ca-server)# no shutdown
```

```
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password:
Jan  6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan  6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan  6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically [OK]
Jan  6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2
% Do you accept this certificate? [yes/no]:
Jan  6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan  6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan  6 22:32:46.798: CRYPTO_CS: created a new serial file.
Jan  6 22:32:46.798: CRYPTO_CS: authenticating the CA 'sub'y
Trustpoint CA certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
Router (cs-server)#
Jan  6 22:33:30.562: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan  6 22:33:32.450: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan  6 22:33:32.454: CRYPTO_CS: exit FSM: new state check failed
Jan  6 22:33:32.454: CRYPTO_CS: cs config has been locked
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint MD5: CED89E5F 53B9C60E
> AA123413 CDDAD964
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 70787C76 ACD7E67F
7D2C8B23 98CB10E7 718E84B1
% Exporting Certificate Server signing certificate and keys...
Jan  6 22:34:53.839: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 22:34:53.843: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan  6 22:34:53.843: CRYPTO_CS: starting enabling checks
Jan  6 22:34:53.843: CRYPTO_CS: nvram filesystem
Jan  6 22:34:53.883: CRYPTO_CS: found existing serial file.
Jan  6 22:34:53.907: CRYPTO_CS: old router cert flag 0x4
Jan  6 22:34:53.907: CRYPTO_CS: new router cert flag 0x44
Jan  6 22:34:56.511: CRYPTO_CS: DB version
Jan  6 22:34:56.511: CRYPTO_CS: last issued serial number is 0x1
Jan  6 22:34:56.551: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 22:34:56.551: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 22:34:56.603: CRYPTO_CS: SCEP server started
Jan  6 22:34:56.603: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:34:56.603: CRYPTO_CS: cs config has been locked
Jan  6 22:35:02.359: CRYPTO_CS: enter FSM: input state enabled, input signal time set
Jan  6 22:35:02.359: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:35:02.359: CRYPTO_CS: cs config has been locked
```

例：ルート証明書サーバの区別

証明書を発行するとき、ルート証明書サーバ（親の下位証明書サーバ）は、次のサンプル出力に示すように、証明書要求を「Sub CA」、「RA」およびピアの各要求に区別します。

```
Device# crypto pki server server1 info req

Enrollment Request Database:
RA certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
Subordinate CS certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
1     pending        CB9977AD8A73B146D3221749999B0F66 hostname=host-subcs.company.com
RA certificate requests:
ReqID     State        Fingerprint                                 SubjectName
--------------------------------------------------------------------------------
Router certificate requests:
```

```
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
```

例：下位証明書サーバの出力表示

次の **showcryptopkiservercommand** 出力は、下位の証明書サーバが設定されたことを示しています。

```
Device# show crypto pki server

Certificate Server sub:
  Status: enabled
  Server's configuration is locked  (enter "shut" to unlock it)
  Issuer name: CN=sub
  CA cert fingerprint: 11B586EE 3B354F33 14A25DDD 7BD39187
  Server configured in subordinate server mode
  Upper CA cert fingerprint: 328ACC02 52B25DB8 22F8F104 B6055B5B
  Granting mode is: manual
  Last certificate issued serial number: 0x1
  CA certificate expiration timer: 22:33:44 GMT Jan 6 2006
  CRL NextUpdate timer: 22:33:29 GMT Jan 13 2005
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

例：RA モード証明書サーバ

次の出力は、RA モード証明書サーバの設定後に、通常表示される内容です。

```
Device-ra(config)# crypto pki trustpoint myra
Device-ra(ca-trustpoint)# enrollment url http://192.0.2.17
! Include "cn=ioscs RA" or "ou=ioscs RA" in the subject-name.
Device-ra(ca-trustpoint)# subject-name cn=myra, ou=ioscs RA, o=company, c=us
Device-ra(ca-trustpoint)# exit
Device-ra(config)# crypto pki server myra
Device-ra(cs-server)# mode ra
Device-ra(cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
Certificate has the following attributes:
Fingerprint MD5: 32661452 0DDA3CE5 8723B469 09AB9E85
Fingerprint SHA1: 9785BBCD 6C67D27C C950E8D0 718C7A14 C0FE9C38
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Ready to request the CA certificate.
%Some server settings cannot be changed after the CA certificate has been requested.
Are you sure you want to do this? [yes/no]: yes
%
% 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 subject name in the certificate will include: cn=myra, ou=ioscs RA, o=company, c=us
% The subject name in the certificate will include: Router-ra.company.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto pki certificate' command will also show the fingerprint.
% Enrollment in progress...
Device-ra (cs-server)#
```

```
Sep 15 22:32:40.197: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 82B41A76 AF4EC87D
AAF093CD 07747D3A
Sep 15 22:32:40.201: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 897CDF40 C6563EAA
0FED05F7 0115FD3A 4FFC5231
Sep 15 22:34:00.366: %PKI-6-CERTRET: Certificate received from Certificate Authority

Device-ra(cs-server)# end
Device-ra# show crypto pki server

Certificate Server myra:
    Status: enabled
    Issuer name: CN=myra
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server is running in RA mode
    Server configured in RA mode
    RA cert fingerprint: C65F5724 0E63B3CC BE7AE016 BE0D34FE
    Granting mode is: manual
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

次の出力は、RA がイネーブルになった後の、発行元証明書サーバの登録要求データベースを示します。

（注） 所有者名に「ou=ioscs RA」が表示されていることから、RA 証明書要求は発行元証明書サーバによって認識されています。

```
Device-ca# crypto pki server mycs info request

Enrollment Request Database:
Subordinate CA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
! The request is identified as RA certificate request.
RA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
12     pending    88F547A407FA0C90F97CDE8900A30CB0
hostname=Router-ra.company.com,cn=myra,ou=ioscs RA,o=company,c=us
Router certificates requests:
ReqID   State     Fingerprint                      SubjectName
--------------------------------------------------------------
! Issue the RA certificate.
Device-ca# crypto pki server mycs grant 12
```

次の出力は、要求が RA から出された場合に、発行元証明書サーバが自動的に証明書を発行するように設定されていることを示します。

```
Device-ca(config)# crypto pki server mycs
Device-ca(cs-server)# grant ra-auto

% This will cause all certificate requests already authorized by known RAs to be automatically
 granted.
Are you sure you want to do this? [yes/no]: yes
Router-ca (cs-server)# end
Device-ca# show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server will issue certificate for requests from the RA.
    Granting mode is: auto for RA-authorized requests, manual otherwise
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 22:29:37 GMT Sep 15 2007
```

```
    CRL NextUpdate timer: 22:29:39 GMT Sep 22 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

次の例は、「myra」の設定（RA サーバ）が自動ロールオーバーを「myca」（CA）からサポートするように設定されていることを示します。RA サーバが設定されると、証明書再登録要求の自動許可がイネーブルになります。

```
crypto pki trustpoint myra
 enrollment url
http://myca
 subject-name ou=iosca RA
 rsakeypair myra
crypto pki server myra
 mode ra
 auto-rollover
crypto pki server mycs
 grant auto rollover ra-cert
 auto-rollover 25
```

例：CA 証明書ロールオーバーを有効にしてただちに開始する

次の例では、**cryptopkiserver** コマンドを使用して、サーバ mycs の自動 CA 証明書ロールオーバーをイネーブルにする方法を示します。**showcryptopkiserver** コマンドを実行すると、mycs サーバの現在の状態と、ロールオーバー証明書が現在ロールオーバーに使用可能であることが示されます。

```
Device(config)# crypto pki server mycs rollover

Jun 20 23:51:21.211:%PKI-4-NOSHADOWAUTOSAVE:Configuration was
modified.  Issue "write memory" to save new IOS CA certificate
! The config has not been automatically saved because the config has been changed.
Device# show crypto pki server

Certificate Server mycs:
    Status:enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name:CN=mycs
    CA cert fingerprint:E7A5FABA 5D7AA26C F2A9F7B3 03CE229A
    Granting mode is:manual
    Last certificate issued serial number:0x2
    CA certificate expiration timer:00:49:26 PDT Jun 20 2008
    CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
    Rollover status:available for rollover
    ! Rollover certificate is available for rollover.
    Rollover CA certificate fingerprint:9BD7A443 00A6DD74 E4D9ED5F B7931BE0
    Rollover CA certificate expiration time:00:49:26 PDT Jun 20 2011
    Auto-Rollover configured, overlap period 25 days
```

次の作業

証明書サーバが正常に実行されたら、登録元クライアントを手動のメカニズムによって（「*PKI の証明書登録の設定*」の説明に従って）開始する、または Web ベースの登録インターフェイスである SDP の設定を（「*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*」の説明に従って）開始できます。

PKI 展開での Cisco IOS XE 証明書サーバの設定および管理に関する追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
PKI およびセキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』
USB トークンによる RSA 処理：初期の自動登録用の USB トークンにおける RSA キーの使用	「*PKI* の証明書登録の設定」
USB トークンによる RSA 処理：USB トークンを使用するメリット	「*PKI* クレデンシャルの保存」
証明書サーバクライアント証明書の登録、自動登録、および自動ロールオーバー	「*PKI* の証明書登録の設定」
USB トークンの設定および USB トークンへのロギング	「*PKI* クレデンシャルの保存」
Web を使用した証明書登録	「*Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI*」
PEM 形式ファイル内の RSA キー	「*PKI* 内での *RSA* キーの展開」
証明書失効メカニズムの選択	「*PKI* での証明書の許可および失効の設定」
推奨暗号化アルゴリズム	『Next Generation Encryption』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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 展開での Cisco IOS XE 証明書サーバの設定および管理に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *9*：*PKI* 展開での *Cisco IOS XE* 証明書サーバの設定および管理に関する機能情報

機能名	リリース	機能情報
PKI IOS XE 証明書サーバ	Cisco IOS XE Release 3.15S	この機能は、Cisco IOS 証明書サーバをサポートしています。これにより、CA が Cisco ソフトウェアと直接統合され、基本 PKI ネットワークの展開がより簡単になりました。 次のコマンドが導入または変更されました。**auto-rollover**、**cryptokeygeneratersa**、**cryptopkicertificatechain**、**cryptopkiexportpem**、**cryptopkiserverinforequest**、**cryptopkiserver**、**databasearchive**、**databaseurl**、**enrollment url (ca-trustpoint)**、**grantautorollover**、**grantra-auto**、**lifetimeenrollment-requests**、**modera**、**showcryptopkicertificates**、**showcryptopkiserver**、**showcryptopkitrustpoint**

第 8 章

PKI クレデンシャルの保存

Rivest、Shamir、Adelman（RSA）キーと証明書などの公開キー インフラストラクチャ（PKI）は、NVRAM やフラッシュ メモリなどのルータまたは USB eTtoken 64 KB スマート カード上の特定の場所に保存できます。USB トークンを使用すると、セキュアな設定配布や、トークン上のキー生成、署名、認証などの RSA 処理、配置のためのバーチャル プライベート ネットワーク（VPN）クレデンシャルを USB トークンのストレージが提供されます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

PKI クレデンシャルを保存するための前提条件

ローカル証明書の保管場所を指定するための前提条件

ローカル証明書の保管場所を指定するためには、ご使用のシステムが次の要件を満たしている必要があります。

- Cisco IOS Release 12.4(2)T PKI 対応イメージまたはそれ以降のイメージ
- PKI クレデンシャルを個別のファイルとして保存できるプラットフォームであること。
- 設定内に証明書が少なくとも 1 つ含まれていること。
- アクセス可能なローカル ファイル システムがあること。

PKI クレデンシャルの保管場所として USB トークンを指定するための前提条件

USB トークンを使用するためには、ご使用のシステムが次の要件を満たしている必要があります。

- Cisco 871 ルータ、Cisco 1800 シリーズ ルータ、Cisco 2800 シリーズ ルータ、Cisco 3800 シリーズ ルータ、または Cisco 7200VXR NPE-G2 プラットフォームを使用していること。
- サポートされているいずれかのプラットフォーム上で、少なくとも Cisco IOS Release 12.3(14)T イメージが稼働していること。
- シスコのサポート対象 USB トークン（Safenet/Aladdin eToken PRO 32 KB または 64 KB）
- k9 イメージを使用していること。

PKI クレデンシャルの保存に関する制約事項

ローカル証明書の保管場所を指定する場合の制約事項

証明書をローカルな保管場所に保存する場合には、次のような制約事項があります。

- 使用できるのはローカル ファイル システムだけです。リモート ファイル システムを選択すると、エラー メッセージが表示され、コマンドは無効になります。
- ローカル ファイル システムでサポートされていれば、サブディレクトリを指定できます。NVRAM では、サブディレクトリはサポートされていません。

保管場所として USB トークンを指定する場合の制約事項

USB トークンを使用して PKI データを保存する場合には、次のような制約事項があります。

- USB トークンがサポートされるためには、ファイルをセキュアに保存できる 3DES（k9）Cisco IOS ソフトウェア イメージが必要です。

- イメージは USB トークンからは起動できません（ただし、設定は USB トークンからでも起動できます）。
- USB ハブは現在、サポートされていません。そのため、サポートされるデバイスの数は、多くても使用できる USB ポートの数までです。

PKI クレデンシャルの保存について

ローカルな保管場所への証明書の保存

デフォルトでは、証明書は NVRAM に格納されます。ただし、ルータによっては、証明書を正常に保存するために必要なサイズの NVRAM が搭載されていないことがあります。

シスコのプラットフォームはすべて、NVRAM およびフラッシュ ローカル ストレージをサポートしています。ご使用のプラットフォームによっては、ブートフラッシュ、スロット、ディスク、USB フラッシュ、USB トークンなど、サポートされているその他のローカル ストレージを使用できます。

実行時には、証明書を保存するアクティブなローカル ストレージ デバイスを指定できます。

PKI クレデンシャルと USB トークン

ご使用のルータ上でセキュアな USB トークンを使用するためには、次に説明する事柄について十分な知識が必要です。

USB トークンの動作のしくみ

スマート カードはプラスティック製の小型カードで、データの保存や処理を行うためのマイクロプロセッサやメモリが搭載されています。USB インターフェイスを備えたスマート カードが USB トークンです。USB トークンでは、記憶域の容量（32KB）内であれば、どのようなタイプのファイルでもセキュアに保存できます。USB トークンに保存されたコンフィギュレーション ファイルに対する暗号化およびアクセスは、ユーザ PIN を介してだけ行えます。デバイスにコンフィギュレーション ファイルをロードするには、デバイスのコンフィギュレーション ファイルをセキュアに配布できるよう適切な PIN が設定されている必要があります。

USB トークンをデバイスに装着したら、その USB トークンにログインする必要があります。ログイン後は、ユーザ PIN（デフォルトは 1234567890）や、ログインが拒否されるようになるまで許容されるログイン試行の失敗回数（デフォルトは15回）など、さまざまなデフォルト設定を変更できます。USB トークンのアクセス方法および設定方法については、「USB トークンへのログインと USB トークンの設定」を参照してください。

USB トークンへ正常にログインした場合は、**copy** コマンドを使用して、デバイスから USB トークンへファイルをコピーできます。USB トークンの RSA キーおよび関連する IPsec トンネルは、デバイスがリロードされるまで使用できます。キーが削除され IPsec トンネルが切断されるまでの

時間を指定する場合は、**cryptopkitokenremovaltimeout** コマンドを発行します。デフォルト タイムアウトはゼロのため、eTokenがデバイスから削除されるとRSAキーが自動的に削除されるようになります。デフォルト値は、実行中のコンフィギュレーションで次のように表示されます。

```
crypto pki token default removal timeout 0
```

次の表に、USB トークンの機能を示します。

表 *10*：*USB* トークンの主な機能性

機能	USB トークン
アクセシビリティ	デジタル証明書、事前共有キー、およびデバイス設定を USB トークンからデバイスへセキュアに保存したり転送したりするためのものです。
ストレージのサイズ	32 KB または 64 KB
ファイル タイプ	• 通常、IPsec VPN 用のデジタル証明書、事前共有キー、およびデバイス設定を保存する場合には、ファイル タイプを指定します。 • USB トークンには、Cisco IOS イメージは保存できません。
セキュリティ	• ファイルに対する暗号化およびアクセスは、ユーザ PIN を介してだけ行えます。 • ファイルは、ノンセキュアなフォーマットでも保存できます。
ブート設定	• デバイスではブート時に、USB トークンに保存されている設定を使用できます。 • デバイスではブート時に、USB トークンに保存されているセカンダリ設定を使用できます（セカンダリ設定を使用すると、ユーザは各自の IPsec 設定をロードできます）。

USB トークンの応用上の利点

Cisco ルータ上で USB トークンがサポートされていることにより、応用上次のような利点が生まれます。

移動可能な証明書：配置する VPN クレデンシャルを外部デバイスに保存できます。

USB トークンでは、スマート カード テクノロジーにより、IPsec VPN の導入に必要なデジタル証明書や設定を保存できます。これにより、ルータにおいて RSA 公開キーを生成し、少なくとも 1 つの IPsec トンネルを認証できるようになりました（ルータでは複数の IPsec トンネルを開始できるため、USB トークンには、必要に応じて複数の証明書を保存できるようになっています）。

VPN クレデンシャルを外部デバイスに保存すると、機密データが漏洩する危険性は低くなります。

ファイルをセキュアに配置するための PIN 設定

USB トークンには、ユーザが設定した PIN を介してルータにおける暗号化をイネーブルにする際に使用できるコンフィギュレーション ファイルを保存できます（つまり、デジタル証明書、事前共有キー、および VPN は使用されません）。

軽減されるまたは不要になる手動での設定作業

USB トークンを使用すると、リモート ソフトウェアの設定やプロビジョニングの際、手動で行う作業がほとんど（あるいは完全に）必要なくなります。設定は自動プロセスとして構成されます。具体的には、ルータに装着した USB トークンにブートストラップ設定を保存しておくと、そのブートストラップ設定によりルータが起動します。さらにこのルータは、ブートストラップ設定によって TFTP サーバへ接続され、その TFTP サーバに保存されている設定に基づいて、すべてのルータ設定が行われます。

RSA 処理

USB トークンは、ストレージデバイス以外に、暗号化装置として使用できます。USB トークンを暗号化装置として使用すると、トークンでキー生成、署名、認証などの RSA 操作を実行できます。

ご使用のトークンストレージデバイス上に配置されているクレデンシャルからは、モジュラスが 2048 ビット以下の汎用 RSA キー ペア、特殊 RSA キー ペア、暗号化 RSA キー ペア、またはシグニチャ RSA キーペアを生成できます。秘密キーは、デフォルトでは配布されず、トークン上に保存されたままです。ただし、秘密キーの保管場所を設定することは可能です。

USB トークン上に常駐するキーは、生成された段階でトークンの永続的な保管場所に保存されます。キーの削除操作を行うと、トークンに保存されているキーは、永続的な保管場所からただちに削除されます（トークン上に常駐していないキーは、 **writememory** またはそれに類するコマンドが発行されると、トークン以外の保管場所で保存や削除が行われます）。

セキュアデバイスプロビジョニング（SDP）環境におけるリモートデバイスの設定およびプロビジョニング

SDP は USB トークンの設定に使用される場合があります。設定された USB トークンを送付すれば、リモート ロケーションにあるデバイスをプロビジョニングできます。つまり、あるネットワーク デバイスから別のリモート ネットワーク デバイスへ暗号化された情報を送る際に USB トークンを使用することで、USB トークンを段階的に配置できます。

SDP で USB トークンを使用する方法については、「その他の関連資料」に記載されている参照先を参照してください。

PKI データの保管場所の設定方法

証明書のローカル ストレージ場所の指定

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkicertificatestorage***location-name*
4. **exit**
5. **copy***source-urldestination-url*
6. **showcryptopkicertificatesstorage**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**cryptopkicertificatestorage***location-name* 例： `Device(config)# crypto pki certificate storage flash:/certs`	証明書のローカルな保管場所を指定します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `Device(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	**copy***source-urldestination-url*
	
	例：
	
	`Device#`
	`copy system:running-config`
	`nvram:startup-config`
		（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションに保存します。
		
		（注） 設定は、実行コンフィギュレーションがスタートアップ コンフィギュレーションに保存された場合にだけ有効になります。

ステップ 6	**showcryptopkicertificatesstorage** 例： `Device# show crypto pki certificates storage`	（任意）PKI 証明書の保管場所に関する現在の設定を表示します。

例

次に、**showcryptopkicertificatesstorage** コマンドの出力例を示します。ここでは、証明書が disk0 の certs サブディレクトリに保存されています。

```
Device# show crypto pki certificates storage
Certificates will be stored in disk0:/certs/
```

Cisco デバイスにおける USB トークンの設定と使用

USB トークンによる設定の保存

手順の概要

1. **enable**
2. **configureterminal**
3. **boot config***usbtoken[0-9]:filename*

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# 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	**boot config***usbtoken[0-9]:filename* 例： `Device(config)# boot config usbtoken0:file`	スタートアップ コンフィギュレーション ファイルがセキュアな USB トークンに保存されるよう指定します。

USB トークンへのログインと USB トークンの設定

RSA キーと USB トークンの併用方法

- RSA キーは、USB トークンがルータへ正常にログインした後にロードされます。
- デフォルトの場合、新規に生成された RSA キーは、最後に装着された USB トークンに保存されます。再生成されたキーは、元の RSA キーが生成されたのと同じ場所に保存する必要があります。

手動ログイン用のデバイスの設定

自動ログインとは異なり、手動ログインを使用する場合は、ユーザが実際の USB トークン PIN を把握している必要があります。

（注） 手動ログインまたは自動ログインのいずれかを使用する必要があります。

手動ログインは、PIN をデバイス上に保存するのが適していない場合に使用できます。また、初期導入時やハードウェア交換時に、デバイスを現地の業者から調達したり、リモート サイトへ直送したりする場合にも、手動ログインが適しています。手動ログインは、権限の有無にかかわらず実行できます。また、手動ログインを実行すると、USB トークン上のファイルおよびRSA キー

が、Cisco IOS ソフトウェアで使用可能になります。セカンダリ コンフィギュレーション ファイルを設定する場合は、ログインを実行するユーザの権限がある場合にだけ手動ログインを実行できます。そのため、何らかの目的で、手動ログインを実行し、USB トークン上にセカンダリ コンフィギュレーション ファイルを設定する場合は、権限をイネーブルにする必要があります。

手動ログインは、失われたデバイス設定のリカバリを行う場合にも使用できます。通常VPNを使用してコア ネットワークへ接続しているリモート サイトが存在する状況では、設定および RSA キーが失われた場合、USB トークンが備えているアウトオブバンド サービスが必要となります。USB トークンには、ブート設定、セカンダリ設定、および接続を認証するためのRSAキーを保存できます。

手順の概要

1. **enable**
2. **crypto pki token***token-name* [**admin**] **login** [*pin*]
3. **show usbtoken***0-9***:***filename*

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**crypto pki token***token-name* [**admin**] **login** [*pin*] 例： `Device# crypto pki token usbtoken0 admin login 5678`	USB トークンに手動でログインします。 **admin** キーワードを最初に指定していない場合は、このキーワード オプションで **crypto pki token** コマンドを再び入力できます。
ステップ 3	**show usbtoken***0-9***:***filename* 例： `Device# show usbtoken0:usbfile`	（任意）USB トークンがデバイスにログインしているかどうかを確認します。

次の作業

USB トークンへのログインが完了すると、次のような作業が行えます。

- USB トークンを詳細に設定する。「USB トークンの設定」の項を参照してください。

- ユーザ PIN の変更、ルータから USB トークンに設定されたキーの保管場所へのファイルのコピー、USB トークンの変更など、USB トークンの管理作業を行う。「USB トークンにおける管理機能の設定」の項を参照してください。

USB トークンの設定

USB トークンに対しては、自動ログインの設定後、さらに次のような設定を行えます。

PIN およびパスフレーズ

自動ログインにおける PIN のセキュリティをさらに強化するため、NVRAM に保存されている PIN を暗号化し、USB トークンにパスフレーズを設定できます。パスフレーズを設定すると、他のユーザには PIN そのものではなく、そのパスフレーズを周知すればよいため、PIN の安全性を維持できます。

このパスフレーズは、USB トークンをデバイスに装着した後、PIN を復号化する際に必要となります。PIN が復号化されると、デバイスはその PIN を使用して USB トークンにログインします。

（注） ユーザがログインするには特権レベル 1 が必要です。

USB トークンのロック/ロック解除

USB トークン自体をロック（暗号化）またはロック解除（復号化）できます。

USB トークンは、ロック解除すると使用できるようになります。ロック解除した場合、Cisco IOS ソフトウェアでは、その USB トークンは自動ログインされたものと見なされ、その USB トークン上にあるいずれかのキーがロードされます。また、セカンダリ コンフィギュレーション ファイルがトークン上に存在する場合は、ログインしたユーザの権限レベルとは独立したフル ユーザ権限（権限レベル 15）を使用して、そのセカンダリ コンフィギュレーション ファイルが実行されます。

トークンをロックした場合は、トークンからログアウトする場合とは異なり、トークンからロードされた RSA キーがすべて削除され、セカンダリ アンコンフィギュレーション ファイルが（もし設定されていれば）実行されます。

セカンダリ コンフィギュレーション ファイルとセカンダリ アンコンフィギュレーション ファイル

USB トークン上に存在するコンフィギュレーション ファイルは、セカンダリ コンフィギュレーション ファイルと呼ばれます。セカンダリ コンフィギュレーション ファイルを作成および設定する場合、セカンダリ コンフィギュレーション ファイルの有無は、NVRAM に保存された Cisco IOS 設定内のセカンダリ コンフィギュレーション ファイル オプションの存在によって決定されます。ユーザがトークンを取り外した後またはトークンからログアウトした後に、無効タイマーで設定された期間が経過すると、別途用意されているセカンダリ アンコンフィギュレーション ファイルが実行され、セカンダリ コンフィギュレーションのすべての要素が、実行コンフィギュレー

ションから削除されます。セカンダリ コンフィギュレーション ファイルおよびセカンダリ アンコンフィギュレーション ファイルは、ログインしたユーザの権限レベルとは関係なく、権限レベル 15 で実行されます。

手順の概要

1. **enable**
2. **cryptopkitoken***token-name***unlock** [*pin*]
3. **configureterminal**
4. **cryptopkitoken***token-name***encrypted-user-pin** [**write**]
5. **cryptopkitoken***token-name***secondaryunconfig***file*
6. **exit**
7. **cryptopkitoken***token-name***lock** [*pin*]

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**cryptopkitoken***token-name***unlock** [*pin*] 例： `Device# crypto pki token mytoken unlock mypin`	（任意）ロックされている USB トークンを使用できるようにします。 ロック解除した場合、Cisco IOS ソフトウェアでは、その USB トークンは自動ログインされたものと見なされ、その USB トークン上にあるいずれかのキーがロードされます。また、セカンダリ コンフィギュレーション ファイルが存在する場合、このファイルは実行されます。
ステップ 3	**configureterminal** 例： `Device# 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 4	**cryptopkitoken***token-name***encrypted-user-pin** [**write**] 例： `Device(config)# crypto pki token mytoken encrypted-user-pin write`	（任意）NVRAM に保存されている PIN を暗号化します。

	コマンドまたはアクション	目的
ステップ 5	**cryptopkitoken***token-name***secondaryunconfig***file* 例： `Device(config)# crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg`	（任意）セカンダリ コンフィギュレーション ファイルとその保管場所を指定します。
ステップ 6	**exit** 例： `Device(config)# exit`	特権 EXEC モードを開始します。
ステップ 7	**cryptopkitoken***token-name***lock** [*pin*] 例： `Device# crypto pki token mytoken lock mypin`	（任意）トークンからロードされた RSA キーをすべて削除し、セカンダリ アンコンフィギュレーション ファイルが存在する場合は、それを実行します。

例

次の例は、ユーザ PIN の設定、ユーザ PIN の暗号化、デバイスのリロード、およびユーザ PIN のロック解除の各プロセスを順に示したものです。

```
! Configuring the user PIN

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# crypto pki token usbtoken0: userpin

Enter password: mypassword

! Encrypt the user PIN

Device(config)# crypto pki token usbtoken0: encrypted-user-pin

Enter passphrase: mypassphrase

Device(config)# exit

Device#

Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console

Device# show running config

crypto pki token usbtoken0 user-pin *encrypted*

! Reloading the router.

Device> enable

Password:

! Decrypting the user pin.
```

```
Device# crypto pki token usbtoken0: unlock

Token eToken is usbtoken0

Enter passphrase: mypassphrase

Token login to usbtoken0(eToken) successful

Device#

Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken

Login Successful
```

次に示すのは、実行コンフィギュレーションからセカンダリ コンフィギュレーションの要素を削除する際に使用されるセカンダリ アンコンフィギュレーション ファイルの設定例です。セカンダリ コンフィギュレーション ファイルを使用して PKI トラストポイントが設定されている場合を例にとると、それに対応するアンコンフィギュレーション ファイル mysecondaryunconfigfile.cfg には、次のようなコマンド ラインが設定されます。

```
no crypto pki trustpoint token-tp
```

トークンが取り外された後で、次のコマンドが実行されると、デバイスの実行コンフィギュレーションから、トラストポイントおよびそれに関連付けられた証明書が削除されます。

```
 Device# configure terminal
Device(config)# no crypto pki token mytoken secondary unconfig mysecondaryunconfigfile.cfg
```

次の作業

USB トークンへのログインおよび USB トークンの設定が完了すると、次のような作業が行えます。ユーザ PIN の変更、ルータから USB トークンに設定されたキーの保管場所へのファイルのコピー、USB トークンの変更など、USB トークンの管理作業を行う。「USB トークンにおける管理機能の設定」の項を参照してください。

USB トークンにおける管理機能の設定

ここでは、ユーザ PIN、USB トークンに対するログイン試行の失敗回数の上限、クレデンシャルの保管場所など、さまざまなデフォルト設定を変更する手順について説明します。

手順の概要

1. **enable**
2. **crypto pki token***token-name***admin**] **change-pin** [*pin*]
3. **crypto pki token***token-namedevice-name***:label***token-label*
4. **configure terminal**
5. **crypto key storage***device-name***:**
6. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label***key-label*] [**exportable**] [**modulus***modulus-size*] [**storage***device-name***:**] [**redundancy**] [**on***device-name***]:**
7. **crypto key move rsa***keylabel* [**non-exportable** | [**on** | **storage**]] *location*
8. **crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]
9. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
10. **exit**
11. **copy usbflash**[*0-9*]**:***filenamedestination-url*
12. **show usbtoken**[*0-9*]**:***filename*
13. **crypto pki token***token-name***logout**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**crypto pki token***token-name***admin**] **change-pin** [*pin*] 例： `Device# crypto pki token usbtoken0 admin change-pin`	（任意）USB トークン上のユーザ PIN 番号を変更します。 • PIN が変更されない場合は、デフォルトの PIN（1234567890）が使用されます。 （注） PIN の変更後は、ログインの失敗回数を 0 にリセットする必要があります（**cryptopkitokenmax-retries** コマンドを使用）。許容されるログインの失敗回数の上限は、15（デフォルト）に設定されています。
ステップ 3	**crypto pki token***token-namedevice-name***:label***token-label* 例： `Device# crypto pki token mytoken usb0: label newlabel`	（任意）USB トークンの名前を設定または変更します。 • *token-label* 引数には、英数字（ダッシュおよびアンダースコアを含む）からなる 31 文字以下の文字列を指定できます。 ヒント このコマンドは、自動ログインやセカンダリ コンフィギュレーションファイルなどのトークン固有の設定用として複数の USB トークンを設定する場合に有用です。

	コマンドまたはアクション	目的
ステップ 4	**configure terminal** 例： `Device# 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 5	**crypto key storage***device-name***:** 例： `Device(config)# crypto key storage usbtoken0:`	（任意）新規作成した RSA キーに対するデフォルトの保管場所を設定します。 （注） 設定の内容にかかわらず、既存のキーは、ロード元のデバイスに保存されます。
ステップ 6	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label***key-label*] [**exportable**] [**modulus***modulus-size*] [**storage***device-name***:**] [**redundancy**] [**on***device-name*]**:** 例： `Device(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:`	（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、デバイスの完全修飾ドメイン名（FQDN）であるデフォルト値が使用されます。 **noshutdown** 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、**cryptocaexportpkcs12** 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。この装置にはユニバーサル シリアルバス（USB）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。

	コマンドまたはアクション	目的
ステップ 7	**crypto key move rsa***keylabel* [**non-exportable** | [**on** | **storage**]] *location* 例： `Device(config)# crypto key move rsa keypairname non-exportable on token`	（任意）既存の Cisco IOS クレデンシャルを、現在の保管場所から指定した保管場所へ移動します。 デフォルトの場合、RSA キー ペアは現在のデバイス上に保存されたままになります。 デバイス上でキーを生成しそれをトークンに移動するまでの所要時間は 1 分未満です。トークン上でキーを生成する際に **on** キーワードを使用すると、USB トークン上で使用可能なハードウェア キー生成ルーチンに応じて、5 ～ 10 分程度の時間がかかります。 Cisco IOS で生成された既存の RSA キー ペアが USB トークンに保存され、登録に使用される場合は、それら既存の RSA キーペアを代替場所に移動して永続的に保存する必要があります。 このコマンドは、USB トークンと SDP を使用してクレデンシャルを配置する場合に有用です。
ステップ 8	**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*] 例： `Device(config)# crypto pki token usbtoken0 removal timeout 60`	（任意）USB トークンがデバイスから取り外されてから、USB トークンに保存されている RSA キーが削除されるまで、デバイスが待機する時間を秒単位で設定します。 （注） このコマンドが発行されない場合は、USB トークンがデバイスから取り外された直後に、すべての RSA キーが削除されるほか、USB トークンに関連付けられている IPsec トンネルもすべて切断されます。
ステップ 9	**crypto pki token** {*token-name* | **default**} **max-retries** [*number*] 例： `Device(config)# crypto pki token usbtoken0 max-retries 20`	（任意）USB トークンへのアクセスが拒否されるまでに許容されるログイン試行の連続失敗回数の上限を設定します。 • デフォルト値は 15 です。
ステップ 10	**exit** 例： `Device(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 11	**copy usbflash**[*0-9*]**:***filenamedestination-url* 例： `Device# copy usbflash0:file1 nvram:`	USB トークンからデバイスへファイルをコピーします。 • *destination-url*：サポートされているオプションのリストについては、**copy** コマンドに関するセクションを参照してください。

	コマンドまたはアクション	目的
ステップ 12	**show usbtoken**[*0-9*]**:***filename* 例： `Device# show usbtoken:usbfile`	（任意）USB トークンに関する情報を表示します。このコマンドを使用すると、USB トークンがデバイスにログインしているかどうかを確認できます。
ステップ 13	**crypto pki token***token-name***logout** 例： `Device# crypto pki token usbtoken0 logout`	USB トークンからデバイスをログアウトします。 （注） USD トークンに何らかのデータを保存する場合は、再度トークンにログインする必要があります。

USB トークンに関するトラブルシューティング

ここでは、次の各 Cisco IOS コマンドについて説明します。これらのコマンドは、USB トークンの使用中に発生しうる問題についてのトラブルシューティングに使用できます。

USB ポート接続のトラブルシューティング

show file systems コマンドを使用すると、USB モジュールが USB ポートに差し込まれていることをルータが認識しているかどうかを判定できます。差し込まれている USB モジュールは、ファイル システムのリスト上に表示されます。これらのモジュールがリスト上に表示されない場合は、次のいずれかの問題が発生している可能性があります。

- USB モジュールとの接続に問題がある。
- ルータ上で稼働している Cisco IOS イメージによりサポートされていない USB モジュールがある。
- USB モジュールそのものにハードウェア上の問題がある。

次に示すのは、**show file systems** コマンドによる出力例です。この中には USB トークンも表示されています。USB モジュールが現れるのはリストの最下行です。

```
Device# show file systems
File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
      491512      486395     nvram     rw   nvram:
           -           -    opaque     wo   syslog:
           -           -    opaque     rw   xmodem:
           -           -    opaque     rw   ymodem:
           -           -   network     rw   rcp:
```

```
          -           -   network     rw   pram:
          -           -   network     rw   ftp:
          -           -   network     rw   http:
          -           -   network     rw   scp:
          -           -   network     rw   https:
          -           -    opaque     ro   cns:
   63158272    33037312  usbflash     rw   usbflash0:
      32768         858  usbtoken     rw   usbtoken1:
```

シスコによりサポートされている USB トークンの特定

show usb device コマンドを使用すると、USB トークンがシスコによりサポートされているかどうかを判定できます。このコマンドの次の出力例では、太字で記されているのが、モジュールがサポートされているかどうかを示す箇所です。

```
Router# show usb device
Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA
    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

USB トークンのデバイス問題の特定

showusbcontrollers コマンドを使用すると、USB フラッシュ モジュールにハードウェア上の問題があるかどうかを判別できます。**showusbcontrollers** コマンドの出力結果にエラーが表示された場合は、USB モジュールにハードウェア上の問題があると考えられます。

USB フラッシュ モジュールに対するコピー操作が正常に行われていることを確認する場合にも、この **showusbcontrollers** コマンドを使用できます。ファイルのコピーを実行した後で、**showusbcontrollers** コマンドを発行すると、データ転送が正常に行われたことを示す内容が表示されます。

次に示すのは、使用中の USB フラッシュ モジュールの **showusbcontrollers** コマンドによる出力例です。

```
Router# show usb controllers
Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF
Int Level:1
Transfer Completion Codes:
         Success              :920             CRC             :0
         Bit Stuff            :0               Stall           :0
         No Response          :0               Overrun         :0
         Underrun             :0               Other           :0
         Buffer Overrun       :0               Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2               Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0               Bulk Transfer   :0
         Isochronous Transfer :0               Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0               Bulk Transfer   :26
         Isochronous Transfer :0               Control Transfer:894
USBD Failures:
         Enumeration Failures :0               No Class Driver Found:0
         Power Budget Exceeded:0
USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3               Command Fail    :0
         Good Status Timed out:0               Device not Found:0
         Device Never Opened  :0               Drive Init Fail :0
         Illegal App Handle   :0               Bad API Command :0
         Invalid Unit Number  :0               Invalid Argument:0
         Application Overflow :0               Device in use   :0
         Control Pipe Stall   :0               Malloc Error    :0
         Device Stalled       :0               Bad Command Code:0
         Device Detached      :0               Unknown Error   :0
         Invalid Logic Unit Num:0
USB Aladdin Token Driver Counters:
         Token Inserted       :1               Token Removed   :0
         Send Insert Msg Fail :0               Response Txns   :434
         Dev Entry Add Fail   :0               Request Txns    :434
```

```
         Dev Entry Remove Fail:0                Request Txn Fail:0
         Response Txn Fail    :0                Command Txn Fail:0
         Txn Invalid Dev Handle:0
USB Flash File System Counters:
         Flash Disconnected   :0                Flash Connected :1
         Flash Device Fail    :0                Flash Ok        :1
         Flash startstop Fail :0                Flash FS Fail   :0
USB Secure Token File System Counters:
         Token Inserted       :1                Token Detached  :0
         Token FS success     :1                Token FS Fail   :0
         Token Max Inserted   :0                Create Talker Failures:0
         Token Event          :0                Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

USB トークン情報の表示

dir コマンドと **filesystem** キーワード オプション **usbtoken***0-9***:** を使用すると、USB トークン上にあるすべてのファイル、ディレクトリ、およびそれらの権限文字列を表示できます。

次の出力例は、USB トークンに関する情報を表示したものです。

```
Device# dir usbtoken1:
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

次の出力例では、デバイスが認識しているすべてのデバイスのディレクトリ情報を表示します。

```
Device# dir all-filesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles
No space information available
Directory of flash:/
    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T
129880064 bytes total (99753984 bytes free)
Directory of nvram:/
  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data
491512 bytes total (486395 bytes free)
Directory of usbflash0:/
    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
```

```
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

PKI データの保存に関する設定例

例：特定のローカルな保管場所への証明書の保存

次に示すのは、certs サブディレクトリに証明書を保存する場合の設定例です。ここでは、certs サブディレクトリは存在しないため、自動的に作成されています。

```
Router# dir nvram:
 114  -rw-        4687                    <no date>  startup-config
 115  ----        5545                    <no date>  private-config
 116  -rw-        4687                    <no date>  underlying-config
   1  ----          34                    <no date>  persistent-data
   3  -rw-         707                    <no date>  ioscaroot#7401CA.cer
   9  -rw-         863                    <no date>  msca-root#826E.cer
  10  -rw-         759                    <no date>  msca-root#1BA8CA.cer
  11  -rw-         863                    <no date>  msca-root#75B8.cer
  24  -rw-        1149                    <no date>  storagename#6500CA.cer
  26  -rw-         863                    <no date>  msca-root#83EE.cer
129016 bytes total (92108 bytes free)
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
  14  -rw-         707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
  15  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
  16  -rw-         759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
  17  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
  18  -rw-        1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
  19  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)
! The certificate files are now on disk0/certs:
```

例：USB トークンへのログインと USB トークンへの RSA キーの保存

次に示すのは、USB トークンにログインして RSA キーを生成し、その RSA キーを USB トークンに保存する場合の設定例です。

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
```

```
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% 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 subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
 ! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

次に示すのは、USB トークンから正常にロードされた保存済みクレデンシャルの **showcryptokeymypubkeyrsa** コマンドによる出力例です。USB トークン上に保存されているクレデンシャルは、保護領域内に存在します。USB トークン上にクレデンシャルを保存する場合、それらのファイルは /keystore というディレクトリに保存されます。ただし、キー ファイルは、コマンドライン インターフェイス（CLI）では表示されません。

```
Router#
show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
ルータへの USB モジュールの接続	『Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide』
eToken および USB フラッシュのデータ シート	『USB eToken and USB Flash Features Support』
RSA キー	PKI 内での RSA キーの展開
ファイル管理（ファイルのロード、コピー、および再起動）	『*Cisco Configuration Fundamentals Configuration Guide*』（Cisco.com）
USB トークンによる RSA 処理：証明書サーバの設定	「PKI 展開での Cisco IOS 証明書サーバの設定および管理」の機能に関する資料。 「Generating a Certificate Server RSA Key Pair」項、「Configuring a Certificate Server Trustpoint」項、および関連する例を参照してください。
USB トークンの RSA 処理：初期自動登録時における USB トークンを使用した RSA 処理	『Configuring Certificate Enrollment for a PKI』の「Configuring Certificate Enrollment or Autoenrollment」項を参照してください。
SDP のセットアップ、設定、および USB トークンとの使用	PKI クレデンシャルの展開での SDP と USB トークンの使用方法については、「Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI」にある機能名の機能情報の項を参照してください。

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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 クレデンシャルの保存に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *11*：*PKI* クレデンシャルの保存に関する機能情報

機能名	リリース	機能情報
証明書：保管場所の指定	Cisco IOS XE Release 2.1	この機能を使用すると、証明書を個別のファイルとして保存する機能をサポートしているプラットフォームにおいて、ローカル証明書の保管場所を指定できます。シスコのプラットフォームはすべて、デフォルトの保管場所として使用するNVRAM、およびフラッシュローカル ストレージをサポートしています。ご使用のプラットフォームによっては、ブートフラッシュ、スロット、ディスク、USB フラッシュ、USB トークンなど、サポートされているその他のローカルストレージを使用できます。 次のコマンドがこの機能で導入されました。**cryptopkicertificatestorage**、**showcryptopkicertificatesstorage**。
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	**cryptokeygeneratersa** 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。

機能名	リリース	機能情報
USB eToken 64KB スマートカード サポート	Cisco IOS XE Release 3.6S	この機能を使用すると、RSA キーおよび証明書の保管場所で USB eTtoken 64 KB スマートカード サポートを可能にします。USB トークンを使用すると、セキュアな設定配布や、トークン上のキー生成、署名、認証などの RSA 処理、配置のためのバーチャル プライベート ネットワーク（VPN）クレデンシャルを USB トークンのストレージが提供されます。 次のコマンドがこの機能で導入されました。**binary file**、**crypto key move rsa**、**crypto key storage**、**crypto pki token change-pin**、**crypto pki token encrypted-user-pin**、**crypto pki token label**、**crypto pki token lock**、**crypto pki token login**、**crypto pki token logout**、**crypto pki token max-retries**、**crypto pki token removal timeout**、**crypto pki token secondary config**、**crypto pki token unlock**、**crypto pki token user-pin**、**show usb-devices summary**、**show usb driver**、**show usbtoken**、**template file**。

第 9 章

CAにおける発信トラフィックの送信元インターフェイス選択機能

認証局（CA）における発信トラフィックの送信元インターフェイス選択機能により、指定のトラストポイントが設定されたときに、インターフェイスのアドレスをそのトラストポイントと関連付けられたすべての TCP 接続の送信元アドレスとして使用するよう設定できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

CAにおける発信トラフィックの送信元インターフェイス選択機能の詳細

エンティティを識別する証明書

証明書を使用して、エンティティを識別できます。認証局（CA）とも呼ばれるトラステッドサーバにより、エンティティの ID を決定した後にエンティティに証明書が発行されます。Cisco IOS XE ソフトウェアを実行しているルータは、CA にネットワーク接続することでその証明書を取得します。Simple Certificate Enrollment Protocol（SCEP）を使用して、ルータはその証明書要求を CA に送信し、許可された証明書を受信します。ルータは、SCEP を使用した場合と同様に CA の証明書を取得します。リモート デバイスからの証明書を検証する場合、ルータは再度 CA または Lightweight Directory Access Protocol（LDAP）サーバあるいは HTTP サーバに連絡して、リモートデバイスの証明書が失効しているかどうか判断できます（このプロセスは、証明書失効リスト（CRL）のチェックとも呼ばれています）。

（注） Cisco IOS リリースに応じて、LDAP がサポートされます。

設定によっては、有効またはルーティング可能な IP アドレスを持たないインターフェイスを使用して発信 TCP 接続を実行できる場合があります。ユーザは、異なるインターフェイスのアドレスを発信接続の送信元 IP アドレスとして使用するよう指定する必要があります。この要件の具体例としてケーブル モデムがあります。発信ケーブル インターフェイス（RF インターフェイス）には通常、ルーティング可能なアドレスがないためです。ただし、ユーザ インターフェイス（通常は FastEthernet）には有効な IP アドレスはありません。

トラストポイントに関連付けられた発信 TCP 接続の送信元インターフェイス

トラストポイントを指定するには、**crypto pki trustpoint** コマンドを使用します。インターフェイスのアドレスを、そのトラストポイントに関連付けられたすべての発信 TCP 接続の送信元アドレスとして指定する場合は、**source interface** コマンドも **crypto pki trustpoint** コマンドとともに使用します。

（注） インターフェイス アドレスが**source interface** コマンドを使用して指定されていない場合は、発信インターフェイスのアドレスが使用されます。

CAにおける発信トラフィックの送信元インターフェイス選択機能の設定方法

トラストポイントに関連付けられたすべての発信 TCP 接続のインターフェイスの設定

トラストポイントに関連付けられたすべての発信 TCP 接続の送信元アドレスとして使用するインターフェイスを設定するには、次の作業を行います。

手順の概要

1. **enable**
2. **configureterminal**
3. **cryptopkitrustpoint***name*
4. **enrollmenturl***url*
5. **sourceinterface***interface-address*
6. **interface***typeslot*/*port*
7. **description***string*
8. **ipaddress***ip-addressmask*
9. **interface***typeslot*/*port*
10. **description***string*
11. **ipaddress***ip-addressmask*
12. **cryptomap***map-name*

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。 • パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 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。

	コマンドまたはアクション	目的
ステップ 3	**cryptopkitrustpoint***name* 例： `Router (config)# crypto pki trustpoint ms-ca`	ルータが使用する認証局（CA）を宣言し、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	**enrollmenturl***url* 例： `Router (ca-trustpoint)# enrollment url http://yourname:80/certsrv/mscep/mscep.dll`	CA の登録パラメータを指定します。
ステップ 5	**sourceinterface***interface-address* 例： `Router (ca-trustpoint)# interface fastethernet1/0`	そのトラストポイントに関連付けられたすべての発信 TCP 接続の送信元アドレスとして使用するインターフェイス。
ステップ 6	**interface***typeslot/port* 例： `Router (ca-trustpoint)# interface fastethernet1/0`	インターフェイス タイプを設定し、インターフェイスコンフィギュレーションモードを開始します。
ステップ 7	**description***string* 例： `Router (config-if)# description inside interface`	インターフェイスの設定に説明を加えます。
ステップ 8	**ipaddress***ip-addressmask* 例： `Router (config-if)# ip address 10.1.1.1 255.255.255.0`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 9	**interface***typeslot/port* 例： `Router (config-if)# interface fastethernet1/0`	インターフェイス タイプを設定します。

	コマンドまたはアクション	目的
ステップ 10	**description***string* 例： `Router (config-if)# description outside interface 10.1.1.205 255.255.255.0`	インターフェイスの設定に説明を加えます。
ステップ 11	**ipaddress***ip-addressmask* 例： `Router (config-if)# ip address 10.2.2.205 255.255.255.0`	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 12	**cryptomap***map-name* 例： `Router (config-if)# crypto map mymap`	インターフェイスに対して以前に定義されたクリプト マップ セットを適用します。

トラブルシューティングのヒント

コマンドで指定されたインターフェイスのアドレスが有効であることを確認します。指定されたインターフェイスのアドレスを使用して別のデバイス（可能性としてはCRLを処理しているHTTPまたは LDAP サーバ）からルータに ping を実行します。外部デバイスからルータへのトレースルートを使用しても同じことができます。

Cisco IOS XE コマンドライン インターフェイス（CLI）を使用して、ルータと CA または LDAP サーバ間の接続をテストすることもできます。**ping ip** コマンドを入力し、プロンプトに回答します。「Extended commands [n]:」プロンプトに「はい」と回答すると、送信元アドレスまたはインターフェイスが指定できるようになります。

また、Cisco IOS XE CLI を使用して traceroute コマンドを入力できます。**traceroute ip** コマンド（EXEC モード）を入力すると、宛先および送信元アドレスを求めるプロンプトが表示されます。CA または LDAP サーバを、宛先および送信元アドレスの「送信元インターフェイス」として指定されたインターフェイスのアドレスとして指定する必要があります。

CAにおける発信トラフィックの送信元インターフェイス選択機能の設定例

CA における発信トラフィックの送信元インターフェイス選択の例

次に、ルータが支社にある例を示します。ルータは IP セキュリティ（IPSec）を使用して本社と通信します。FastEthernet 1 は、ISP（インターネット サービス プロバイダー）に接続する「外部」インターフェイスです。FastEthernet 0 は、支社の LAN に接続されたインターフェイスです。本社にある CA サーバにアクセスするには、ルータは IPSec トンネルを使用してその IP データグラムを外部インターフェイスである FastEthernet 1（アドレス 10.2.2.205）に送信する必要があります。アドレス 10.2.2.205 は ISP により割り当てられています。アドレス 10.2.2.205 は支社または本社の一部ではありません。

CA は、ファイアウォールがあるため、社外のアドレスにはアクセスできません。CA は 10.2.2.205 から発信されたメッセージを確認しますが、応答はできません（つまり、CA は、ルータが支社の到達可能なアドレス 10.1.1.1 にあることを認識していません）。

source interface コマンドを追加すると、ルータはアドレス 10.1.1.1 を CA に送信される IP データグラムの送信元アドレスとして使用するよう指示されます。CA は 10.1.1.1 に応答できます。

このシナリオは、上記の **source interface** コマンドとインターフェイス アドレスを使用して設定されています。

```
crypto pki trustpoint ms-ca
 enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
 source interface fastethernet0
!
interface fastethernet 0
 description inside interface
 ip address 10.1.1.1 255.255.255.0
!
interface fastethernet 1
 description outside interface
 ip address 10.2.2.205 255.255.255.0
 crypto map main-office
```

その他の参考資料

次に、CA における発信トラフィックの送信元インターフェイスの機能に関する資料を示します。

関連資料

関連項目	マニュアル タイトル
IPsec と認証局の設定	「Security for VPNs with IPsec」
IPsec と認証局に関するコマンド	『*Cisco IOS Security Command Reference*』

標準

標準	タイトル
この機能がサポートする新しい標準または変更された標準はありません。	-

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、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
この機能でサポートが追加または変更された RFC はありません。	-

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/en/US/support/index.html

CAにおける発信トラフィックの送信元インターフェイス選択の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *12*：*CA* における発信トラフィックの送信元インターフェイス選択の機能情報

機能名	リリース	機能情報
CA における発信トラフィックの送信元インターフェイス選択機能	Cisco IOS XE Release 2.1	この機能により、指定のトラストポイントが設定されたときに、インターフェイスのアドレスをそのトラストポイントと関連付けられたすべての TCP 接続の送信元アドレスとして使用するよう設定できます。 次のコマンドが導入されました。**sourceinterface**

用語集

認証：ID の証明書および ID がもたらす秘密を使用してエンティティの ID を証明すること（通常は、秘密キーは証明書の公開キーに対応します）。

CA：認証局（CA）。CA はデジタル証明書を発行するエンティティ（特に X.509 証明書）で、証明書のデータ項目間のバインディングを保証します。

CA 認証：ユーザはルート CA からの証明書を手動で承認します。通常は、証明書のフィンガープリントがユーザに提示され、ユーザはフィンガープリントに基づく証明書を承認するよう求められます。ルート CA の証明書は、通常の証明書確認プロセスで自動的に認証できないよう、自ら署名（自己署名）されます。

CRL：Certificate Revocation List（証明書失効リスト）。CRL は、発行者により無効にされたデジタル証明書をそれらの期限満了予定までに列挙するデータ構造です。

登録：ルータは登録プロセス経由でその証明書を受信します。ルータは、（PKCS #10 と呼ばれる）特定の形式で証明書の要求を生成します。その要求は CA に転送され、CA は要求を許可し、要求と同じ形式に符号化された証明書を生成します。ルータは許可された証明書を受信し、通常操作中に使用するため、内部データベースに保管します。

証明書：エンティティ（マシンまたはユーザ）をそのエンティティの公開キーと関連付けるため国際標準化機構（ISO）規格 X.509 で定義されたデータ構造。証明書には、エンティティの名前など特定のフィールドが含まれています。証明書は通常は、エンティティに代わって CA により発行されます。この場合は、ルータが CA としての役割を果たします。証明書内の共通フィールドには、エンティティの認定者名（DN）、証明書を発行している認証局の DN、およびエンティティの公開キーがあります。

LDAP：ライトウェイト ディレクトリ アクセス プロトコルLDAP は、X.500 ディレクトリに読み書きインタラクティブアクセスできる、管理アプリケーションおよびブラウザアプリケーションにアクセスできるプロトコルです。

第 10 章

PKI トラストプール管理

PKI トラストプール管理機能を使用すると、認証局（CA）と呼ばれる一般的に認識された信頼できるエージェントを使用して、デバイス間で発生する HTTPS などのセッションを認証できます。デフォルトで有効に設定されているこの機能を使用すると、セッションのセキュリティ保護のためにブラウザが提供するサービスと同じ方法で、既知の CA の証明書のプールのプロビジョニング、保管、管理を行うスキーマを作成できます。

（注） Cisco IOS XE Denali 16.3 から、PKI トラストプールが管理される方法が変更されました。このリリースへのアップグレードを計画している場合は、「*PKI* トラストプールの拡張」項に含まれる次の機能に対する変更を確認してください。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモ

ジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

PKI トラストプール管理の前提条件

証明書を使用するには、暗号化サブシステムが Cisco IOS ソフトウェア イメージに含まれている必要があります。

PKI トラストプール管理の制約事項

CA 証明書を使用するデバイス証明書は PKI トラストプールに登録できません。

バンドルに同じ発行者名とサブジェクト名のペアを持つ 2 つの証明書を指定することはできません。

PKI トラストプール管理の情報

PKI トラストプール内の CA 証明書の保管場所

ルータは、PKI トラストプールと呼ばれる特別な証明書ストアに格納された内蔵型 CA 証明書バンドルを使用します。これはシスコから自動的に更新されます。この PKI トラストプールは、シスコおよび他のベンダーにも知られています。CA 証明書バンドルは次の形式で提供されます。

- 公開キー暗号化メッセージ構文標準 7（pkcs7）内にエンベロープ化された、Distinguished Encoding Rules（DER）バイナリ形式の X.509 証明書。PKI でメッセージの署名と暗号化に使用します。X.509 証明書は、PKI と権限管理インフラストラクチャ（PMI）の標準で、特に、公開キー証明書の標準形式、証明書失効リスト、属性証明書、および認証パス検証アルゴリズムを指定します。
- PEM ヘッダー付きプライバシー強化メール（PEM）形式の連結型 X.509 証明書を含むファイル。

（注） また、NVRAM の代わりに、バンドルの保管場所としてフラッシュも使用できます。

PKI トラストプールの更新

PKI トラストプールは、次の条件が発生した場合に更新する必要のある単一エンティティとして処理されます。

- PKI トラストプールの証明書が期限切れまたは再発行されている。
- 公開された CA 証明書のバンドルに、特定のアプリケーションで必要な追加の信頼できる証明書が含まれている。
- 設定が破損している。

（注） PKI トラストプールに組み込まれた証明書は物理的に置き換えることができません。ただし、組み込まれた証明書の X.509 所有者名属性が CA 証明書バンドル内の証明書と一致する場合、組み込まれた証明書は無効と表示されます。

PKI トラストプールは自動または手動で更新できます。PKI トラストプールを使用するアプリケーションによっては、PKI トラストプールが証明書検証で使用される場合があります。詳細については「PKI トラストプール内の証明書の手動更新」と「PKI トラストプール ポリシー パラメータの設定」の項を参照してください。

（注） 自動更新中、ダウンロードした既存のすべてのトラストプールの証明書を削除する必要があります。

PKI トラストプール タイマーは、最初に失効する CA 証明書と一致します。タイマーが作動しても、バンドルのロケーションが設定されておらず、明示的に無効になっていない場合、syslog 警告が発効され、PKI トラストプール ポリシー オプションが設定されていないことが管理者に警告されます。

PKI トラストプールの自動更新では設定済み URL を使用します。

PKI トラストプールが失効すると、ポリシーが読み込まれ、バンドルがロードされ、PKI トラストプールが置き換えられます。PKI トラストプールの自動更新の開始時に問題が発生した場合は、ダウンロードが成功するまで、次のスケジュールで更新が開始されます。20 日、15 日、10 日、5 日、4 日、3 日、2 日、1 日、最後に 1 時間ごとです。

PKI トラストプールとトラストポイントの両方での CA 処理

PKI トラストプールとトラストポイントの両方に CA が格納されている場合があります。たとえば、トラストポイントで CA を使用し、CA バンドルが同じ CA 内で後からダウンロードされたりします。このシナリオでは、PKI トラストプール管理機能がルータに実装されても、現在の動作が変更されないようにするため、トラストポイント内の CA とこのトラストポイントのポリシーが、PKI トラストプールまたは PKI トラストプール ポリシーの CA よりも優先されます。

PKI トラストプールの拡張機能

Cisco IOS XE Denali 16.3 より前のリリースでは、トラストプールは、すべてのシスコ ボックスで展開された内蔵型証明書と、公開されたバンドルからダウンロードした CA 証明書で構成されています。ダウンロードした証明書は、デフォルトでは NVRAM に保存されます。ダウンロードしたトラストプール バンドルの証明書は抽出され、非効率的で多くの領域を使用する実行コンフィギュレーションに保存されていました。

Cisco IOS XE Denali 16.3 以降、PKI トラストプールの拡張機能では、これまでのリリースのような個別の証明書の代わりに、保管場所（デフォルトでは NVRAM）にあるファイルと同じダウンロードしたバンドル形式でバンドルが保存されます。このため、ファイルが圧縮形式の場合は、ストレージ メモリが節約されます。また、証明書は実行コンフィギュレーションでは個別に表示されません。再起動するたびに、バンドルは保存場所から読み取られ、個別の証明書がデータベースにインストールされます。

この機能は、実行コンフィギュレーションから現在のダウンロードした証明書を削除します。これらの証明書は古い NVRAM および新しいイメージと互換性がないため、**crypto pki certificate pool** には DER 形式の証明書を指定できません。アップグレード中、DER 形式のトラストプール証明書が失われたら、バンドルを保管場所に再インストールする必要があります。古い NVRAM ファイルの場合、これは再起動時に syslog に記されます。**show crypto pki trustpool** コマンドは、実行コンフィギュレーションが削除されたことを示します。アップグレード前に、**show crypto pki trustpool** コマンドを使用し、証明書が利用可能かどうかを確認します。

Cisco IOS XE Denali 16.3 へのアップグレード前に、次の手順を実行する必要があります。

- **crypto pki trustpool clean** コマンドを使用して、ダウンロードしたトラストプール証明書を削除します
- **write memory** コマンドを使用します
- デバイスを再起動します
- **crypto pki trustpool import url** コマンドを使用して、トラストプール バンドルをダウンロードします。

SSH へのログインにトラストプールを使用している場合、追加の手順を実行して、特定の証明書をバンドルからトラストポイントに転送する必要があります。詳細については、「例：アップグレード中の *SSH* 接続に *PKI* トラストプールを使用」を参照してください。

PKI トラストプール管理の設定方法

PKI トラストプールの証明書の手動更新

PKI トラストプール管理機能はデフォルトで有効で、PKI トラストプールに組み込まれた CA 証明書バンドルを使用し、シスコから自動更新を受信します。PKI トラストプール内の証明書が最新

のものではない、破損している、または特定の証明書を更新する必要がある場合は、次の作業を実行して手動で更新します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpool import clean** [**terminal** | **url***url*]
4. **crypto pki trustpool import** {**terminal**} {**url***url* | **ca-bundle**} {**vrf***vrf-name* | **source interface***interface-name*}
5. **exit**
6. **show crypto pki trustpool**
7. **show crypto pki trustpool built-in**
8. **show crypto pki trustpool policy**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 trustpool import clean** [**terminal** | **url***url*] 例： `Device(config)# crypto pki trustpool import clean`	（任意）ダウンロードしたすべての PKI CA 証明書を手動で削除します。 • **clean** キーワードは、新しい証明書のダウンロードの前に、ダウンロード済みの PKI トラストプール証明書の削除を指定します。 • **terminal** キーワードは、既存の CA 証明書バンドル端末設定を削除します。 • **url** キーワードおよび *url* 引数は、既存の URL ファイル システム設定を削除します。

	コマンドまたはアクション	目的
ステップ 4	**crypto pki trustpool import** {**terminal**} {**url***url* | **ca-bundle**} {**vrf***vrf-name* | **source interface***interface-name*} 例： `Device(config)# crypto pki trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b`	CA 証明書バンドルを PKI トラストプールに手動でインポート（ダウンロード）したり、既存の CA 証明書バンドルを交換したりします。 • **terminal** キーワードを指定すると、端末（カットアンドペースト）を介して CA 証明書バンドルが PEM 形式でインポートされます。 • **url** キーワードと *url* 引数を指定すると、URL を介して CA 証明書バンドルがインポートされます。この URL は、HTTP などのさまざまな URL ファイルシステムを経由できます。詳細については、「*PKI* トラストプールの更新」の項を参照してください。CA バンドルで、**crypto pki trustpool import** コマンドを使用すると、グローバル VRF を介してトラフィックを転送できます。また、VRF と送信元インターフェイスを指定する **crypto pki trustpool policy** コマンドを設定すると、トラフィックが VRF を介して転送されることはありません。
ステップ 5	**exit** 例： `Device(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 pki trustpool** 例： `Device(config)# show crypto pki trustpool`	冗長形式でルータの PKI トラストプール証明書を表示します。
ステップ 7	**show crypto pki trustpool built-in** 例： `Device(config)# show crypto pki trustpool built-in`	冗長形式でルータに組み込まれた PKI トラストプール証明書を表示します。
ステップ 8	**show crypto pki trustpool policy** 例： `Device(config)# show crypto pki trustpool policy`	

オプション PKI トラストプール ポリシー パラメータの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpool policy**
4. **cabundle url** {*url* | **none**}
5. **chain-validation**
6. **crl** {**cache** {**delete-after** {*minutes* | **none**} | **query***url*}
7. **default***command-name*
8. **match certificate***certificate-map-name* [**allow expired-certificate** | **override** {**cdp directory***ldap-location* | **ocsp** {*number***url***url* | **trustpool***name number***url***url*} | **sia***number url*} | **skip** [**revocation-check** | **authorization-check**]]
9. **ocsp** {**disable-nonce** | **url***url*}
10. **revocation-check***method1* [*method2* [*method3*]]
11. **source interface***name number*
12. **storage***location*
13. **vrf***vrf-name*
14. **show**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 trustpool policy**
	
	例：
	
	`Device(config)# crypto pki trustpool policy`
	`Device(ca-trustpool)#`
		CA PKI トラストプール ポリシー パラメータを設定するコマンドにアクセスできる、ca-trustpool コンフィギュレーション モードを入力します。トラストプール ポリシーは crl 検索プロセスにのみ影響し、トラストプール インポート プロセスには影響しません。

	コマンドまたはアクション	目的
ステップ 4	**cabundle url** {*url* | **none**} 例： `Device(ca-trustpool)# cabundle url http://www.cisco.com/security/pki/crl/crca2048.crl`	PKI トラストプール認証局の CA 証明書バンドルのダウンロード元となる URL を指定します。 • *url* 引数は CA 証明書バンドルの URL です。 • **none** 引数を指定すると、PKI トラストプール CA の自動更新が許可されません。
ステップ 5	**chain-validation** 例： `Device(ca-trustpool)# chain-validation`	ピアの証明書から PKI トラストプールのルート CA 証明書までチェーン検証を有効にします。デフォルトの検証はピア証明書の発行者で停止します。
ステップ 6	**crl** {**cache** {**delete-after** {*minutes* | **none**} | **query***url*} 例： `Device(ca-trustpool)# crl query http://www.cisco.com/security/pki/crl/crca2048.crl`	PKI トラストプールの証明書失効リスト（CRL）クエリおよび CRL キャッシュ オプションを指定します。 • **cache** キーワードは CRL キャッシュ オプションを指定します。 • **delete-after** キーワードは、タイムアウト後にキャッシュから CRL を削除します。 • *minutes* 引数は、キャッシュから CRL が削除されるまで待機する分数（1 ～ 43,200）です。 • **none** キーワードを指定すると、CRL がキャッシュ化されます。 • *url* 引数の **query** キーワードは、CRL を照会するために CA サーバによって公開される URL を指定します。
ステップ 7	**default***command-name* 例： `Device(ca-trustpool)# default crl query http://www.cisco.com/security/pki/crl/crca2048.crl`	ca-trustpool コンフィギュレーション サブコマンドの値をデフォルト値にリセットします。 • *command-name* 引数は、その適用可能なキーワードを設定した ca-trustpool コンフィギュレーション モード コマンドです。
ステップ 8	**match certificate***certificate-map-name* [**allow expired-certificate** | **override** {**cdp directory***ldap-location* | **ocsp** {*number***url***url* | **trustpool***name number***url***url*} | **sia***number url*} | **skip** [**revocation-check** | **authorization-check**]]	PKI トラストプールの証明書マップを使用できるようにします。 • *certifcate-map-name* 引数は証明書マップ名と一致します。

	コマンドまたはアクション	目的
	例： `match certificate mycert override ocsp 1 url http://ocspts.identrust.com`	• オプションの **allow expired-certificate** キーワードは、失効した証明書を無視します。 （注） このキーワードを設定しないと、ルータは失効した証明書を無視しません。 • **override** キーワードは、PKI トラストプール内にある証明書の Online Certificate Status Protocol（OCSP）または SubjectInfoAccess（SIA）属性フィールドを上書きします。 • **cdp** キーワードは、証明書の証明書分散ポイント（CDP）を上書きします。 • **directory** キーワードおよび *ldap-location* は、証明書内で上書きする http: または ldap: URL の CDP、あるいは LDAP ディレクトリを指定します。 • **ocsp** キーワードと *number* 引数および **url** キーワードと *url* 引数は、証明書内で上書きする OCSP シーケンス番号（0 ～ 10000）および URL を指定します。 • **trustpool** キーワードと *name* や *number* 引数および **url** キーワードと *url* 引数は、PKI トラストプール名、シーケンス番号、URL を指定することで、OCSP 証明書を確認するための PKI トラストプールを上書きします。 • **sia** キーワードと *number* や *url* 引数は、SIA シーケンス番号と URL を指定することで、証明書内の SIA URL を上書きします。 • オプションの **skip revocation-check** キーワードを組み合わせると、PKI トラストプールが特定の証明書を除いた証明書失効リスト（CRL）を適用できます。 （注） このキーワードの組み合わせを設定しないと、PKI トラストプールはすべての証明書に CRL を適用します。 • オプションの **skip authorization-check** キーワードを組み合わせると、公開キー インフラストラクチャ（PKI）と AAA サーバとの統合を設定した場合に、証明書の認証、許可、アカウンティング（AAA）の確認をスキップします。

	コマンドまたはアクション	目的
		（注） このキーワードの組み合わせを設定せずに、PKI と AAA サーバとの統合を設定すると、証明書の AAA の確認が行われます。
ステップ 9	**ocsp** {**disable-nonce** | **url***url*} 例： `Device(ca-trustpool)# ocsp url http://ocspts.identrust.com`	PKI トラストプールの OCSP 設定を指定します。 • **disable-nonce** キーワードは OCSP ナンス拡張部を無効にします。 • **url** キーワードと *url* 引数は、証明書の Authority Info Access（AIA）拡張部で上書きする（存在する場合）OCSP サーバの URL を指定します。設定した PKI トラストプールに関連するすべての証明書は、指定した HTTP URL の OCSP サーバによって確認されます。使用可能な URL は、ホスト名、IPv4 アドレス、または IPv6 アドレスです。
ステップ 10	**revocation-check***method1* [*method2* [*method3*]] 例： `Device(ca-trustpool)# revocation-check ocsp crl none`	PKI トラストプールポリシー使用時の失効確認を無効にします。*method* 引数は、ルータが証明書の失効ステータスを確認するために使用されます。使用可能なキーワードは次のとおりです。 • **crl** キーワードは、証明書失効リスト（CRL）で証明書の確認を行います。これはデフォルトの動作です。 • **none** キーワードでは、証明書の確認が必要ありません。 • **ocsp** キーワードは、Online Certificate Status Protocol（OCSP）サーバによって証明書の確認を行います。 2 番目と 3 番目の方法を指定した場合、各方法はその直前の方法でエラーが返された場合（サーバがダウンしている場合など）にだけ使用されます。
ステップ 11	**source interface***name number* 例： `Device(ca-trustpool)# source interface tunnel 1`	CRL の取得、OCSP ステータス、または PKI トラストプールの CA 証明書バンドルのダウンロードに使用する送信元インターフェイスを指定します。 • *name* および *number* 引数は、PKI トラストプールの送信元アドレスとして使用されるインターフェイスのタイプと数値です。

	コマンドまたはアクション	目的
ステップ 12	**storage***location* 例： `Device(ca-trustpool)# storage storage disk0:crca2048.crl`	PKI トラストプール証明書がルータ上で保存される場合のファイル システム ロケーションを指定します。 • *location* は、PKI トラストプール証明書が保存されるファイル システム ロケーションです。ファイル システム ロケーションのタイプには、**disk0:**、**disk1:**、**nvram:**、**unix:**、または名前付きファイル システムがあります。
ステップ 13	**vrf***vrf-name* 例： `Device(ca-trustpool)# vrf myvrf`	登録、CRL の取得、および OCSP ステータスに使用される VPN ルーティングおよび転送（VRF）インスタンスを指定します。

ステップ 14	**show**
	
	例：
	`Device(ca-trustpool)# show`
	
	`Chain validation will stop at the first CA certificate in the pool`
	`  Trustpool CA certificates will expire 12:58:31 PST Apr 5 2012`
	`  Trustpool policy revocation order:  crl`
	`  Certficate matching is disabled`
	`  Policy Overrides:`
		ルータの PKI トラストプール ポリシーを表示します。

PKI トラストプール管理の設定例

例：PKI トラストプール管理の設定

次の **show crypto pki trustpool** コマンド出力は、PKI トラストプールの証明書を表示します。

（注） この例のコマンド出力は、デバッグのためなので省略されています。

```
Device# show crypto pki trustpool

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 00D01E474000000111C38A964400000002
  Certificate Usage: Signature
  Issuer:
    cn=DST Root CA X3
```

```
    o=Digital Signature Trust Co.
  Subject:
    cn=Cisco SSCA
    o=Cisco Systems
  CRL Distribution Points:
    http://crl.identrust.com/DSTROOTCAX3.crl
  Validity Date:
    start date: 12:58:31 PST Apr 5 2007
    end   date: 12:58:31 PST Apr 5 2012

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 6A6967B3000000000003
  Certificate Usage: Signature
  Issuer:
    cn=Cisco Root CA 2048
    o=Cisco Systems
  Subject:
    cn=Cisco Manufacturing CA
    o=Cisco Systems
  CRL Distribution Points:
    http://www.cisco.com/security/pki/crl/crca2048.crl
  Validity Date:
    start date: 14:16:01 PST Jun 10 2005
    end   date: 12:25:42 PST May 14 2029
```

次の **show crypto pki trustpool verbose** コマンド出力は、PKI トラストプールの証明書を表示します。

```
Device# show crypto pki trustpool verbose

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=Licensing Root - DEV
    o=Cisco
  Subject:
    cn=Licensing Root - DEV
    o=Cisco
  Validity Date:
    start date: 03:25:43 IST Apr 25 2013
    end   date: 03:25:43 IST Apr 25 2033
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: 432CBFA0 32D2983A 8A56A319 FD28C6F9
  Fingerprint SHA1: 6341FCAF 19CE9FEE 961D92A5 D47390B5 2DD6D94D
  X509v3 extensions:
    X509v3 Key Usage: 6000000
      Key Cert Sign
      CRL Signature
    X509v3 Subject Key ID: 43214521 B5FB217A 1A4D1BB7 0236E664 CBEC8B65
    X509v3 Basic Constraints:
        CA: TRUE
    Authority Info Access:
  Associated Trustpoints: Trustpool
  Trustpool: Built-In
```

次の **show crypto pki trustpool built-in** コマンド出力は、PKI トラストプールに組み込まれた証明書を表示します。

（注） この例のコマンド出力は、デバッグのためなので省略されています。

```
Device# show crypto pki trustpool built-in

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 02
  Certificate Usage: Signature
  Issuer:
    cn=Cisco Root CA M2
    o=Cisco
  Subject:
    cn=Cisco Manufacturing CA SHA2
    o=Cisco
  CRL Distribution Points:
    http://www.cisco.com/security/pki/crl/crcam2.crl
  Validity Date:
    start date: 19:20:58 IST Nov 12 2012
    end   date: 12:02:01 IST Oct 7 1901
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
  Signature Algorithm: SHA256 with RSA Encryption
  Fingerprint MD5: AC14F08F C3780F8F D9EEE6C9 39111280
  Fingerprint SHA1: 90B2E06B 7AD5DAFF CFD43187 2909F381 37471BF8
    X509v3 Key Usage: 6000000
      Key Cert Sign
      CRL Signature
    X509v3 Subject Key ID: 7AD77995 CABB482B B85514FD A3C00FBC A70F9619
    X509v3 Basic Constraints:
        CA: TRUE
  X509v3 Authority Key ID: C900F91F 8A1FC266 BDA5D26D 650E222E 34C305A0
    Authority Info Access:
    X509v3 CertificatePolicies:
        Policy: 1.3.6.1.4.1.9.21.1.18.0
            Qualifier ID: 1.3.6.1.5.5.7.2.1
            Qualifier Info: http://www.cisco.com/security/pki/policies/index.html
  Associated Trustpoints: Trustpool
  Trustpool: Built-In
```

次の **show crypto pki trustpool policy** コマンド出力は、PKI トラストプールの証明書を表示します。

```
Device# show crypto pki trustpool policy

Trustpool Policy

   Chain validation will stop at the first CA certificate in the pool
   Trustpool CA certificates will expire 05:29:59 IST Aug 3 2028
   Trustpool policy revocation order:      crl
   Certficate matching is disabled
   Policy Overrides:
```

次の **show crypto pki certificates pool** コマンド出力は、PKI トラストプールの証明書を表示します。

```
Device# show crypto pki certificate pool

! ('certificate ca' cmd has been deprecated. Downloaded
! Trustpool certificates should be re-downloaded
! using 'crypro pki trustpool import url')
cabundle nvram:ios_core.p7b
```

例：アップグレード中の SSH 接続に PKI トラストプールを使用

Cisco IOS XE Denali 16.3 へアップグレードの前に、トラストプールから新しいトラストポイントに証明書をコピーします。

```
Device # show run | sec pool
crypto pki trustpool policy
 revocation-check none
 source interface GigabitEthernet0/0/0
crypto pki certificate pool
 certificate ca 01
  308204FA 308202E2 A0030201 02020101 300D0609 2A864886 F70D0101 0C050030
  0E310C30 0A060355 04031303 61626330 1E170D31 36303730 35303435 3935335A
  170D3136 30373035 30353535 35335A30 0E310C30 0A060355 04031303 61626330
  82022230 0D06092A 864886F7 0D010101 05000382 020F0030 82020A02 82020100
  C78AA144 8EC1D18A 4EECC3E8 81450CC7 A85A4C57 AF59E584 5C1EA888 6EF70DA8
  33327D93 E1F6CED7 32BB4FCF 693F60E0 37000225 40F6F9C5 0462C4AD 899E5BDD
  ED779180 D6C75E1B FBE97D42 E2A7B35D DDC18C4D 4CCDE401 68F67A6D E40FD744
  904EE49F 40820640 C6E0B072 510BC40E A0883F6C E8DF5128 EFF3B5F4 B31E5C16
  217652FF AFC30EBF 593CB19C 56C0E793 2814D504 0E079E0C 8E9E856A BCADB19C
  F2376994 A0A040C1 7BC1E88F CF80F218 9C48B4D9 F84ED5C0 79827BD1 32448478
  8F1F82F2 C91A9479 692B6456 C53CF937 777D0C31 1B8A1F5E 24B33553 047C2448
  855CF974 DFA21665 8AD8A0E5 81ED8068 81688997 FF05118C 93A59CA0 7FD594F6
  B7B1898C 272E089A 3392A2C4 22A22625 2BC1E16F 95B2FC15 207CCA49 378AD3A6
  0C574197 C5E94D8C E6736271 CE0BA9AB ACB380E3 A8084243 4E038DD1 8E86E206
  E2269290 F1AFB29A D28CFB3A 5ABADE4A 21A59728 7174E7A3 2FF59C90 E6100C6E
  E2E8CB4C 91BD574D 57B5E18A 78F9CE75 624C4A2E 1A6EFCC3 7D1BB20B 1CC79024
  CD2FBC4D 46BE1B7A 6EFD8F05 6FD84E91 51215E9B E5E952A4 6E2D1388 10075706
  7D6FAF9B 3F7F8994 F39B9B5D 0C7CD5BC 40738877 5D9985AC 5AB6363D 811BA440
  41A1639F 352F4F01 1994300A A4B85B75 01486CA0 4C4B3175 82038B26 BEFE1D2A
  4AC0D577 7784FACF A6877D68 5D73DD04 DC8D942B DE3FC9FE 4C1FF715 A2E7A5AB
  02030100 01A36330 61300F06 03551D13 0101FF04 05300301 01FF300E 0603551D
  0F0101FF 04040302 0186301F 0603551D 23041830 168014CA 195EDBF1 51753A92
  71342CA8 36DDABA9 63A93130 1D060355 1D0E0416 0414CA19 5EDBF151 753A9271
  342CA836 DDABA963 A931300D 06092A86 4886F70D 01010C05 00038202 0100553B
  FB77A348 C4447C40 BEB2DDFD 63C82441 3CBDC198 B5D5B1AB DF17C4E2 98AEAF2F
  CD570939 BCC116E0 33CFF471 E91EE308 8B29B5BD 11DFACF9 A3AC3135 8BE81B22
  ED205587 5DE04654 A051CC14 CA8D2A6E 81F924DA 001BB1C4 7F85F177 4E75D8EA
  797CCAEF 1502492D 17627CD1 E39E295B 44C55884 8E6DFF68 2129B222 18E3187D
  AB97B4A7 6F838E75 A8908566 AD9E6687 35B150DE 0C8C1B37 6F17FDAC 7A7C53A4
  434F5CF3 6EB71957 E65EC5D2 7685B05B A9D8C0D3 2DB8F97E E6B37E11 C9E26F4F
  BFB97745 83E1A214 461B0E49 0FFDEF21 A7CA5364 44416002 03A01F0C 2BC098D3
  B50A4071 AC4D2234 4E55C5D4 0FD9C308 63F2A8D4 24D34613 B73EAA1B B407D56F
  90EEF5C7 AE61C0D8 13FB493D 0E1C8F9B 1D2D6DEA 458CDE18 8753FF14 F8C75213
  35557FCC 50405056 D9790AF0 EAC21646 2D9AF88D 59C05434 45F21248 0BB72191
  74D951DD 9D23997E 1134611E 837137E6 C40C694E 7AB4A05F E8470E87 E0F6D924
  A69A98A8 5AA2B9B3 B7446883 94A7230D EE3C6EDA 4A348351 FC40C16D 6FDC91EC
  CEFF580B F7826DD1 1D1D07DB 17CA3298 8C510826 D2712E04 EB669909 3D8106EB
  5391A5BA 80B7E981 B41AAEB9 CE4A5236 20E30AE7 01D5FDB3 604C5505 0F8C96DC
  8F5CF569 5D90C1FB F5679221 B7B922C0 5F11C379 9EBA283C 45A209F7 132B8DA2
  EAF4751B 290A1CAC C3E7978B 760FB05A 185991FE 4884FA1A D3EEDD7C 63
  3B
        quit
```

新しいトラストポイントを作成し、設定モードで証明書を貼り付けます。

```
Device(config)#cry pki trust abc
Device(ca-trustpoint)#cry pki cert chain abc
Device(config-cert-chain)#certificate ca 01
```

16 進数で証明書を入力します

```
Device(config-pki-hexmode)#  308204FA 308202E2 A0030201 02020101 300D0609 2A864886 F70D0101
```

```
 0C050030
Device(config-pki-hexmode)#  0E310C30 0A060355 04031303 61626330 1E170D31 36303730 35303435
 3935335A
Device(config-pki-hexmode)#  170D3136 30373035 30353535 35335A30 0E310C30 0A060355 04031303
 61626330
Device(config-pki-hexmode)#  82022230 0D06092A 864886F7 0D010101 05000382 020F0030 82020A02
 82020100
Device(config-pki-hexmode)#  C78AA144 8EC1D18A 4EECC3E8 81450CC7 A85A4C57 AF59E584 5C1EA888
 6EF70DA8
Device(config-pki-hexmode)#  33327D93 E1F6CED7 32BB4FCF 693F60E0 37000225 40F6F9C5 0462C4AD
 899E5BDD
Device(config-pki-hexmode)#  ED779180 D6C75E1B FBE97D42 E2A7B35D DDC18C4D 4CCDE401 68F67A6D
 E40FD744
Device(config-pki-hexmode)#  904EE49F 40820640 C6E0B072 510BC40E A0883F6C E8DF5128 EFF3B5F4
 B31E5C16
Device(config-pki-hexmode)#  217652FF AFC30EBF 593CB19C 56C0E793 2814D504 0E079E0C 8E9E856A
 BCADB19C
Device(config-pki-hexmode)#  F2376994 A0A040C1 7BC1E88F CF80F218 9C48B4D9 F84ED5C0 79827BD1
 32448478
Device(config-pki-hexmode)#  8F1F82F2 C91A9479 692B6456 C53CF937 777D0C31 1B8A1F5E 24B33553
 047C2448
Device(config-pki-hexmode)#  855CF974 DFA21665 8AD8A0E5 81ED8068 81688997 FF05118C 93A59CA0
 7FD594F6
Device(config-pki-hexmode)#  B7B1898C 272E089A 3392A2C4 22A22625 2BC1E16F 95B2FC15 207CCA49
 378AD3A6
Device(config-pki-hexmode)#  0C574197 C5E94D8C E6736271 CE0BA9AB ACB380E3 A8084243 4E038DD1
 8E86E206
Device(config-pki-hexmode)#  E2269290 F1AFB29A D28CFB3A 5ABADE4A 21A59728 7174E7A3 2FF59C90
 E6100C6E
Device(config-pki-hexmode)#  E2E8CB4C 91BD574D 57B5E18A 78F9CE75 624C4A2E 1A6EFCC3 7D1BB20B
 1CC79024
Device(config-pki-hexmode)#  CD2FBC4D 46BE1B7A 6EFD8F05 6FD84E91 51215E9B E5E952A4 6E2D1388
 10075706
Device(config-pki-hexmode)#  7D6FAF9B 3F7F8994 F39B9B5D 0C7CD5BC 40738877 5D9985AC 5AB6363D
 811BA440
Device(config-pki-hexmode)#  41A1639F 352F4F01 1994300A A4B85B75 01486CA0 4C4B3175 82038B26
 BEFE1D2A
Device(config-pki-hexmode)#  4AC0D577 7784FACF A6877D68 5D73DD04 DC8D942B DE3FC9FE 4C1FF715
 A2E7A5AB
Device(config-pki-hexmode)#  02030100 01A36330 61300F06 03551D13 0101FF04 05300301 01FF300E
 0603551D
Device(config-pki-hexmode)#  0F0101FF 04040302 0186301F 0603551D 23041830 168014CA 195EDBF1
 51753A92
Device(config-pki-hexmode)#  71342CA8 36DDABA9 63A93130 1D060355 1D0E0416 0414CA19 5EDBF151
 753A9271
Device(config-pki-hexmode)#  342CA836 DDABA963 A931300D 06092A86 4886F70D 01010C05 00038202
 0100553B
Device(config-pki-hexmode)#  FB77A348 C4447C40 BEB2DDFD 63C82441 3CBDC198 B5D5B1AB DF17C4E2
 98AEAF2F
Device(config-pki-hexmode)#  CD570939 BCC116E0 33CFF471 E91EE308 8B29B5BD 11DFACF9 A3AC3135
 8BE81B22
Device(config-pki-hexmode)#  ED205587 5DE04654 A051CC14 CA8D2A6E 81F924DA 001BB1C4 7F85F177
 4E75D8EA
Device(config-pki-hexmode)#  797CCAEF 1502492D 17627CD1 E39E295B 44C55884 8E6DFF68 2129B222
 18E3187D
Device(config-pki-hexmode)#  AB97B4A7 6F838E75 A8908566 AD9E6687 35B150DE 0C8C1B37 6F17FDAC
 7A7C53A4
Device(config-pki-hexmode)#  434F5CF3 6EB71957 E65EC5D2 7685B05B A9D8C0D3 2DB8F97E E6B37E11
 C9E26F4F
Device(config-pki-hexmode)#  BFB97745 83E1A214 461B0E49 0FFDEF21 A7CA5364 44416002 03A01F0C
 2BC098D3
Device(config-pki-hexmode)#  B50A4071 AC4D2234 4E55C5D4 0FD9C308 63F2A8D4 24D34613 B73EAA1B
 B407D56F
Device(config-pki-hexmode)#  90EEF5C7 AE61C0D8 13FB493D 0E1C8F9B 1D2D6DEA 458CDE18 8753FF14
 F8C75213
Device(config-pki-hexmode)#  35557FCC 50405056 D9790AF0 EAC21646 2D9AF88D 59C05434 45F21248
 0BB72191
Device(config-pki-hexmode)#  74D951DD 9D23997E 1134611E 837137E6 C40C694E 7AB4A05F E8470E87
 E0F6D924
Device(config-pki-hexmode)#  A69A98A8 5AA2B9B3 B7446883 94A7230D EE3C6EDA 4A348351 FC40C16D
 6FDC91EC
Device(config-pki-hexmode)#  CEFF580B F7826DD1 1D1D07DB 17CA3298 8C510826 D2712E04 EB669909
 3D8106EB
```

```
Device(config-pki-hexmode)#  5391A5BA 80B7E981 B41AAEB9 CE4A5236 20E30AE7 01D5FDB3 604C5505
 0F8C96DC
Device(config-pki-hexmode)#  8F5CF569 5D90C1FB F5679221 B7B922C0 5F11C379 9EBA283C 45A209F7
 132B8DA2
Device(config-pki-hexmode)#  EAF4751B 290A1CAC C3E7978B 760FB05A 185991FE 4884FA1A D3EEDD7C
 63
Device(config-pki-hexmode)#  3B
Device(config-pki-hexmode)#        quit
```

これで Cisco IOS XE Denali 16.3 にアップグレードできるようになりました。トラストプールの証明書は消えていますが、トラストポイントにはまだ保管されています。アップグレード後にトラストプールに証明書をインストールします。

PKI トラストプール管理の追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
セキュリティ コマンド	• 『Cisco IOS Security Command Reference: Commands A to C』 • 『Cisco IOS Security Command Reference: Commands D to L』 • 『Cisco IOS Security Command Reference: Commands M to R』 • 『Cisco IOS Security Command Reference: Commands S to Z』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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 トラストプール管理の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *13* : *PKI* トラストプール管理の機能情報

機能名	リリース	機能情報
PKI トラストプール管理		PKI トラストプール管理機能を使用すると、認証局（CA）と呼ばれる一般的に認識された信頼できるエージェントを使用して、デバイス間で発生する HTTPS などのセッションを認証できます。デフォルトで有効に設定されているこの機能を使用すると、セッションのセキュリティ保護のためにブラウザが提供するサービスと同じ方法で、既知の CA の証明書のプールのプロビジョニング、保管、管理を行うスキーマを作成できます。 次のコマンドが導入または変更されました。**cabundle url**、**chain-validation (ca-trustpool)**、**crypto pki trustpool import**、**crypto pki trustpool policy**、**crl**、**default (ca-trustpool)**、**match certificate (ca-trustpool)**、**ocsp**、**show (ca-trustpool)**、**show crypto pki trustpool**、**source interface (ca-trustpool)**、**storage**、**vrf (ca-trustpool)**、**show crypto pki trustpool built-in**、**crypto pki trustpool import clean ca-bundle**。
PKI トラストプールの拡張機能	Cisco IOS XE Denali 16.3.1	PKI トラストプールの拡張機能は、ルータで構築された HTTPS 接続の認証に使用されます。 次のコマンドが導入または変更されました。**show crypto pki trustpool built-in**、**crypto pki trustpool import clean ca-bundle**。

第 11 章

トラストポイントの PKI 分割 VRF

トラストポイントの PKI 分割 VRF 機能を使用すると、証明書登録と失効で VPN ルーティングおよび転送（VRF）を設定できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

トラストポイントの PKI 分割 VRF に関する情報

トラストポイントの PKI 分割 VRF の概要

トラストポイントの PKI 分割 VRF 機能を使用すると、証明書登録と証明書失効リスト（CRL）の確認で VPN ルーティングおよび転送（VRF）を設定できます。VRF は、**crypto pki profile enrollment** コマンドの後に **enrollment url** コマンドを使用して登録プロファイルに設定し、この登録プロファイルをトラストポイントに添付します。登録および CRL に同じ VRF を設定したり、異なる VRF を設定したりできます。設定（登録または失効）に基づいて、対応する VRF が選択され、Simple Certificate Enrollment Protocol（SCEP）要求が各 VRF を介して送信されます。

さまざまなルーティングパスを介して登録および CRL を設定するには、**crypto pki profile enrollment** コマンドを使用して登録 url コマンドを設定する必要があります。ここで設定した VRF は登録 VRF として動作し、登録要求はこの VRF を介して送信されます。ただし、CRL はトラストポイントで設定したグローバル VRF を使用します。

enrollment url コマンドで設定した VRF がない場合は、登録が **crypto pki trustpoint** コマンドで設定されるグローバル登録に変わります。

トラストポイントの PKI 分割 VRF の設定方法

分割 VRF の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki profile enrollment***label*
4. **enrollment url***url* [**vrf***vrf-name*]
5. **exit**
6. **show crypto pki profile**
7. **show crypto pki trustpoint**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# 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 profile enrollment***label* 例： `Device(config)# crypto pki profile enrollment pki_profile`	登録プロファイルを定義し、ca-profile-enroll コンフィギュレーション モードを開始します。 • *label*：登録プロファイルの名前。登録プロファイル名は、**enrollmentprofile** コマンドで指定された名前と同じである必要があります。
ステップ 4	**enrollment url***url* [**vrf***vrf-name*] 例： `Device(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe vrf vrf1`	証明書登録要求を HTTP または TFTP によって送信する CA サーバの URL および VPN ルーティングおよび転送（VRF）を指定します。
ステップ 5	**exit** 例： `Device(ca-profile-enroll)# exit`	ca-profile-enroll コンフィギュレーション モードを終了します。 • グローバル コンフィギュレーション モードを終了するため、このコマンドをもう一度入力します。
ステップ 6	**show crypto pki profile** 例： `Device# show crypto pki profile`	（任意）PKI プロファイルの情報を表示します。
ステップ 7	**show crypto pki trustpoint** 例： `Device# show crypto pki trustpoint`	（任意）PKI トラストポイントの情報を表示します。

トラストポイントの PKI 分割 VRF の設定例

例：トラストポイントの PKI 分割 VRF の設定

同一 VRF を介した登録と証明書失効リスト

次の例では、同一 VRF を介した登録と証明書失効リスト（CRL）の設定方法について示します。

```
crypto pki trustpoint trustpoint1
 enrollment url http://10.10.10.10:80
 vrf vrf1
 revocation-check crl
```

異なる VRF を介した登録と証明書失効リスト

次の例では、異なる VRF を介した登録と証明書失効リスト（CRL）の設定方法について示します。

```
crypto pki profile enrollment pki_profile
 enrollment url http://10.10.10.10:80 vrf vrf2

crypto pki trustpoint trustpoint1
 enrollment profile pki_profile
 vrf vrf1
 revocation-check crl
```

トラストポイントの PKI 分割 VRF の追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
セキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』
推奨暗号化アルゴリズム	『Next Generation Encryption』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

トラストポイントの PKI 分割 VRF の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *14*：トラストポイントの *PKI* 分割 *VRF* の機能情報

機能名	リリース	機能情報
トラストポイントの PKI 分割 VRF	Cisco IOS XE 3.11S	トラストポイントの PKI 分割 VRF 機能を使用すると、証明書登録と失効で VPN ルーティングおよび転送（VRF）を設定できます。 次のコマンドが導入または変更されました。**enrollment url (ca-profile-enroll).**

第 12 章

EST クライアント サポート

EST クライアント サポート機能を使用すると、SSL または TLS を使用して転送の安全性を保護しながら、すべてのトラストポイントの EST（Enrollment Over Secure Transport）を有効にできます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

EST クライアント サポートの前提条件

- **ip http authentication fore-close** コマンドを有効にする必要があります。
- TLS 認証に使用するのは RSA 証明書のみです。

EST クライアント サポートの制約事項

- EST クライアントでサポートされるのは TLS 1.0 のみです。
- 証明書属性要求はサポートされていません。
- CA 証明書のロールオーバーはサポートされていません。
- 証明書のない TLS 認証はサポートされていません。

EST クライアント サポートの情報

EST クライアント サポートの概要

EST クライアント サポート機能を使用すると、証明書をプロビジョニングするための証明書管理プロトコルとして Enrollment over Secure Transport（EST）を使用できます。PKI コンポーネント内に統合された既存の SCEP 登録では、EST を追加すると、転送を保護する SSL または TLS を使用する新しいコンポーネントが導入されます。PKI にはすべての証明書が格納されます。

EST サポートを有効にするには、EST クライアントが、TLS 接続の確立中にサーバを認証する必要があります。この認証では、TLS サーバがクライアントのクレデンシャルを要求する場合があります。

EST クライアント サポートの設定方法

EST を使用するためのトラストポイントの設定

ユーザが登録プロファイルを使用できるようにすることで、EST（Enrolment Over Secure Transport）を使用するトラストポイントを設定するには、この作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki profile enrollment** *label*
4. **method-est**
5. **enrollment url** *url* [**vrf** *name*]
6. **enrollment credential** *label*
7. **exit**
8. **show crypto pki profile**
9. **show crypto pki trustpoint**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 profile enrollment** *label* 例： `Device(config)# crypto pki profile enrollment pki_profile`	登録プロファイルを定義し、ca-profile-enroll コンフィギュレーション モードを開始します。 • *label*：登録プロファイルの名前。登録プロファイル名は、**enrollment profile** コマンドで指定された名前と同じである必要があります。
ステップ 4	**method-est** 例： `Device(ca-profile-enroll)# method-est`	登録プロファイルで EST の使用を選択できるようにします。
ステップ 5	**enrollment url** *url* [**vrf** *name*] 例： `Device(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe vrf vrf1`	登録プロファイルが証明書認証および登録用に使用されるように指定します。

	コマンドまたはアクション	目的
ステップ 6	**enrollment credential** *label* 例： `Device(ca-profile-enroll)# enrollment credential test_label`	TLS クライアント認証にプロファイルで現在利用可能なサーバ トラストポイントを提供します。
ステップ 7	**exit** 例： `Device(ca-profile-enroll)# exit`	ca-profile-enroll コンフィギュレーション モードを終了します。
ステップ 8	**show crypto pki profile** 例： `Device# show crypto pki profile`	（任意）PKI プロファイルの情報を表示します。
ステップ 9	**show crypto pki trustpoint** 例： `Device# show crypto pki trustpoint`	（任意）PKI トラストポイントの情報を表示します。

EST クライアント サポートの設定例

EST を使用するためのトラストポイントの設定例

次の例では、Enrollment over Secure Transport（EST）を使用するためにトラストポイントを設定する方法について示します。

```
crypto pki profile enrollment pki_profile
 method-est
 enrollment url http://www.example.com/BigCA/est/simpleenroll.dll
 enrollment credential test_label
```

EST クライアント サポートの追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』

関連項目	マニュアル タイトル
セキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』

標準および RFC

標準/RFC	タイトル
RFC 7030	『*Enrollment over Secure Transport*』
RFC 2818	『*HTTP Over TLS*』
RFC 6125	『*Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)*』
RFC 2510	『*Internet X.509 Public Key Infrastructure Certificate Management Protocols*』
RFC 4210	『*Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*』

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

EST クライアント サポートの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *15*：*EST* クライアント サポートの機能情報

機能名	リリース	機能情報
EST クライアント サポート	Cisco IOS XE リリース 3.14S	EST クライアント サポート機能を使用すると、SSL または TLS を使用して転送の安全性を保護しながら、すべてのトラストポイントの EST（Enrollment Over Secure Transport）を有効にできます。 **method-est** コマンドが導入されました。

第 13 章

OCSP 応答ステープリング

OCSP 応答ステープリング機能では、Online Certificate Status Protocol（OCSP）を使用してデジタル証明書に含まれるピアのユーザまたはデバイス クレデンシャルの有効期間を確認できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュール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。最新の機能情報および警告については、Bug Search Tool およびご使用のプラットフォームおよびソフトウェア リリースのリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

OCSP 応答ステープリングの情報

OCSP 応答ステープリングの概要

ピアが失効情報を取得し、この情報を検証して証明書失効のステータスを確認する場合、Online Certificate Status Protocol（OCSP）は証明書失効を確認するための方式になります。この方式では、

証明書失効のステータスは、クラウドを介して OCSP 応答者に到達するピアの能力、または証明書失効情報を検索する際の証明書送信者の能力によって制限されます。

OCSP 応答ステープリングは、デバイスの独自の証明書で OCSP 応答を取得する新しい方式をサポートします。この機能を使用すると、OCSP サーバに接続し、この結果とその証明書をピアに直接送信して、その独自の証明書失効情報を入手できます。その結果、ピアが OCSP 応答者に接続する必要はありません。

OCSP 応答ステープリングの設定方法

EKU 属性を要求するための PKI クライアントの設定

次の作業を実行し、OCSP（Online Certificate Status Protocol）応答ステープリングを設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **ocsp url** *url*
5. **eku request** *attribute*
6. **match eku** *attribute*
7. **revocation-check** *method1* [*method2* [*method3*]]
8. **exit**
9. **exit**
10. **show cry pki counters**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 **1** パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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。

	コマンドまたはアクション	目的
ステップ 3	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint msca`	トラストポイントおよび設定された名前を宣言して、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	**ocsp url** *url*
	
	例：
	
	`Device(ca-trustpoint)# ocsp url http://ocsp-server`
	
	例：
	
	`Device(ca-trustpoint)# ocsp url http://10.10.10.1:80`
	
	例：
	
	`Device(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80`
		url 引数は、トラストポイントが証明書ステータスをチェックできるように OCSP サーバの URL を指定します。この URL は、証明書の AIA 拡張部に指定されている OCSP サーバの URL（存在する場合）を上書きします。設定したトラストポイントに関連するすべての証明書は、OCSP サーバによって確認されます。使用可能な URL は、ホスト名、IPv4 アドレス、または IPv6 アドレスです。

ステップ 5	**eku request** *attribute* 例： `Device(ca-trustpoint)# eku request ssh-client`	証明書に指定した eku *attribute* を含めるように要求します。この要求は、PKI クライアントで設定した場合、登録時に CA サーバに送信されます。 *attribute* 引数には次のいずれかを指定できます。 • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client

	コマンドまたはアクション	目的
ステップ 6	**match eku** *attribute* 例： `Device(ca-trustpoint)# match eku client-auth`	指定した属性が証明書内に存在し、他の検証が失敗した場合のみ、PKI はピア証明書を検証できます。 *attribute* 引数には次のいずれかを指定できます。 • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client
ステップ 7	**revocation-check** *method1* [*method2* [*method3*]] 例： `Device(ca-trustpoint)# revocation-check ocsp none`	（任意）証明書の失効ステータスをチェックします。 • crl：CRL によって証明書をチェックします。これがデフォルトのオプションです。 • none：証明書のチェックを無視します。 • ocsp：OCSP サーバによって証明書をチェックします。 2 番目と 3 番目の方法を指定した場合、各方法はその直前の方法でエラーが返された場合（サーバがダウンしている場合など）にだけ使用されます。
ステップ 8	**exit** 例： `Device(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、グローバル コンフィギュレーション モードに戻ります。
ステップ 9	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**show cry pki counters** 例： `Device# show cry pki counters`	（任意）デバイスの PKI カウンタを表示します。

EKU 属性を追加するための PKI サーバの設定

次の作業を実行し、OCSP（Online Certificate Status Protocol）応答ステープリングを設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*
5. **eku request** *attribute*
6. **exit**
7. **exit**
8. **show crypto pki counters**

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 **1** パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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	**ip http server** 例： `Device(config)# ip http server`	ご使用のシステムの HTTP サーバをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**crypto pki server** *cs-label* 例： `Device(config)# crypto pki server server-pki`	証明書サーバのラベルを定義し、証明書サーバコンフィギュレーション モードを開始します。 （注） 手動で RSA キー ペアを生成した場合、*cs-label* 引数はキーペアの名前と一致する必要があります。
ステップ 5	**eku request** *attribute* 例： `Device(cs-server)# eku request ssh-server`	証明書に指定した eku *attribute* を含めるように要求します。 *attribute* 引数には次のいずれかを指定できます。 • client-auth • code-signing • email-protection • ipsec-end-system • ipsec-tunnel • ipsec-user • ocsp-signing • server-auth • time-stamping • ssh-server • ssh-client
ステップ 6	**exit** 例： `Device(cs-server)# exit`	cs-server コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**exit** 例： `Device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 8	**show crypto pki counters** 例： `Device# show crypto pki counters`	（任意）デバイスの PKI カウンタを表示します。

次に、**show crypto pki counters** の出力例を示します。

```
Device# show crypto pki counters

PKI Sessions Started: 0
```

```
PKI Sessions Ended: 0
PKI Sessions Active: 0
Successful Validations: 0
Failed Validations: 0
Bypassed Validations: 0
Pending Validations: 0
CRLs checked: 0
CRL - fetch attempts: 0
CRL - failed attempts: 0
CRL - rejected busy fetching: 0
OCSP - fetch requests: 0
OCSP - received responses: 0
OCSP - failed attempts: 0
OCSP - staple requests: 0
AAA authorizations: 0
```

OCSP 応答ステープリングの追加資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Master Command List, All Releases』
セキュリティ コマンド	• 『Cisco IOS Security Command Reference Commands A to C』 • 『Cisco IOS Security Command Reference Commands D to L』 • 『Cisco IOS Security Command Reference Commands M to R』 • 『Cisco IOS Security Command Reference Commands S to Z』

標準および RFC

標準/RFC	タイトル
RFC 2560	『*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*』
RFC 4806	『*Online Certificate Status Protocol (OCSP) Extensions to IKEv2*』
RFC 5280	『*Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*』

標準/RFC	タイトル
RFC 6187	『*X.509v3 Certificates for Secure Shell Authentication*』
RFC 6066	『*Transport Layer Security (TLS) Extensions: Extension Definitions*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

OCSP 応答ステープリングの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを

示しています。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。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *16*：*OCSP* 応答ステープリングの機能情報

機能名	リリース	機能情報
OCSP 応答ステープリング	Cisco IOS XE リリース 3.14S	この機能では、Online Certificate Status Protocol（OCSP）を使用してデジタル証明書に含まれるピアのユーザまたはデバイスクレデンシャルの有効期間を確認できます。