PKI展開での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） セキュリティに対する脅威は、脅威からの保護に役立つ暗号化技術と同様に絶え間なく変化しています。最新のシスコ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。

コピー中に、running-config に CA 証明書と ID 証明書の両方が含まれている場合、CA 証明書が running-config と同じであれば、CA と ID は置き換えられません。一方、CA 証明書が異なる場合は、ID 証明書と CA 証明書の両方がクリアされ、新しい CA が再挿入されます。

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

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 サーバー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、を参照してください。

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。
-

- サードパーティの OpenSSL を使用して HTTP 接続を検証するために、完全な ISE 証明書チェーンがデバイスに送信されます。これらの証明書には、ISE 証明書とその発行元 CA 証明書が含まれます。環境データには、これらの証明書がリストされます。

  バージョン 2.7.0.310 以前を実行している Cisco ISE は、環境データの一部として着信証明書リストに証明書チェーンを入れます。Cisco IOS XE リリース 17.1.1 以前のリリースでは、Ciscoルータは、ISEからのマルチチェーン証明書のダウンロードをサポートしていません。そのため、デバイスは、ISE 証明書を受信せず、TLS ハンドシェイクエラーが表示されます。

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、「証明書サーバーの *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 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、「証明書サーバーの *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

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 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

証明書サーバーは、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、**database level** 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 *1*：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

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 *2:* 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、証明書サーバーデバイス上に設定された追加のトラストポイントを指定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、証明書サーバートラストポイントを自動的に登録しないでください。証明書サーバーの初期登録は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。

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、**lifetime crl** 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、証明書サーバー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、証明書サーバーから CRL を取得するために非 SCEP 要求を指定できます。この *cs-addr* は証明書サーバーの場所です。

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

（注） また、 CA が HTTP CDP サーバーとしても設定されている場合、**cdp-url** http://*cs-addr*/cgi-bin/pkiclient.exe?operation=GetCRL コマンドシンタックスを使用して 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、証明書サーバーが実行されてから、**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。証明書サーバーは、**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、**crypto pki server request pkcs10** コマンドを使用します。

表 *3:* 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

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、**auto-rollover**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、証明書サーバー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、ユーザーは Cisco IOSCisco IOS XE 証明書サーバー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、「下位証明書サーバーの設定」の作業を参照してください。

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） セキュリティに対する脅威は、脅威からの保護に役立つ暗号化技術と同様に絶え間なく変化しています。最新のシスコ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 key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
5. **crypto key import rsa** *key-label* **pem** [**usage-keys** | **signature** | **encryption**] {**terminal** | **url** *url*} [**exportable**] [**on** *devicename:*] *passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 key generate rsa** [**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、**crypto pki server** *cs-label* コマンドによって証明書サーバー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

	コマンドまたはアクション	目的
		ドメイン名（FQDN）であるデフォルト値が使用されます。 **no shutdown** コマンドを発行する前に、CA 証明書が生成されるまで待ってからエクスポート可能な RSA キーペアを手動で生成する場合、**crypto ca export pkcs12** コマンドを使用して、証明書サーバー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。 • **on** キーワードは、指定したデバイス上で RSA キーペアが作成されることを指定します。このデバイスには Universal Serial Bus（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	**crypto key export rsa** *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	**crypto key import rsa** *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、**on** キーワードを使用して、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ー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：	グローバル コンフィギュレーションを終了します。

	コマンドまたはアクション	目的
	`Device(config)# exit`	
ステップ 7	**show crypto key mypubkey rsa** 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

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. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*
5. **no shutdown**
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 モードを有効にします。 • 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。

	コマンドまたはアクション	目的
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。
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	**no shutdown** 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。
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、サーバー 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ー 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） セキュリティに対する脅威は、脅威からの保護に役立つ暗号化技術と同様に絶え間なく変化しています。最新のシスコ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。

始める前に

- ルート証明書サーバーは、Cisco IOS XE 証明書サーバー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. **configure terminal**

3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
6. **exit**
7. **crypto pki server** *cs-label*
8. **issuer name** *DN-string*
9. **mode sub-cs**
10. **auto-rollover** [*time-period*]
11. **grant auto rollover** {**ca-cert** | **ra-cert**}
12. **hash** {**md5** | **sha1** | **sha256** | **sha384** | **sha512**}
13. **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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 trustpoint** *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 RA モードを指定します。デフォルトでは、RA モードは無効です。
		• （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。

	コマンドまたはアクション	目的
		（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、証明書要求にプライバシー強化メール（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`	（任意）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。 • **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（非推奨）。 • **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（非推奨）。 • **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。

	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	**crypto pki server** *cs-label* 例： `Device(config)# crypto pki server sub`	Cisco IOS XE 証明書サーバーをイネーブルにし、CS サーバー 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	**issuer name** *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 サーバー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。
ステップ 11	**grant auto rollover** {**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。

	コマンドまたはアクション	目的
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）はサーバー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（非推奨）。 • **sha1**：SHA-1 ハッシュ関数が使用されるように指定します（非推奨）。 • **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	**no shutdown** 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

証明書サーバーがイネーブルにならない、あるいは証明書サーバー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、**debug crypto pki server** 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
```

Clock Not Set

```
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
```

Trustpoint Not Configured

```
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
```

証明書サーバーが署名証明書をルート証明書サーバーから取得できない場合は、次の例に示すように、**debug crypto pki transactions** 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.
```

証明書サーバーがイネーブルにならない、あるいは証明書サーバー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、**debug crypto pki server** 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

証明書サーバは、 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. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **subject-name** *x.500-name*
6. **exit**
7. **crypto pki server** *cs-label*
8. **mode ra** [**transparent**]
9. **auto-rollover** [*time-period*]
10. **grant auto rollover** {**ca-cert** | **ra-cert**}
11. **no shutdown**
12. **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	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 trustpoint** *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	**enrollment url** *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）。
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。

	コマンドまたはアクション	目的
ステップ 7	**crypto pki server** *cs-label* 例： `Device(config)# crypto pki server ra-server`	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	**mode ra** [**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 サーバーが複数のタイプの CA サーバー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	**grant auto rollover** {**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	**no shutdown** 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、 **no shutdown** コマンドを再度発行する必要があります。これにより、証明書サーバー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	**no shutdown** 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 **crypto pki server info-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. **crypto pki server** *cs-label* **info requests**
3. **crypto pki server** *cs-label* **grant** *req-id*
4. **configure terminal**
5. **crypto pki server** *cs-label*
6. **grant ra-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 モードを有効にします。 • 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 server** *cs-label* **info requests** 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。
ステップ 3	**crypto pki server** *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。

	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 pki server** *cs-label* 例： `Device(config)# crypto pki server root-server`	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	**grant ra-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、「証明書サーバー機能の設定」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）、証明書サーバーは 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. **database url** *root-url*
2. **database url** {**cnm** | **crl** | **crt** | **p12** | **pem** | **ser**} *root-url*
3. **database url** {**cnm** | **crl** | **crt**} **publish** *root-url*
4. **database level** {**minimal** | **names** | **complete**}
5. **database username** *username* [**password** [*encr-type*] *password*]
6. **database archive** {**pkcs12** | **pem**}[**password** *encr-type*] *password*]
7. **issuer-name** *DN-string*
8. **lifetime** {**ca-certificate** | **certificate**} *time*
9. **lifetime crl** *time*
10. **lifetime enrollment-request** *time*
11. **cdp-url** *url*
12. **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	**database url** *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	**database url** {**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。
ステップ 3	**database url** {**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。

	コマンドまたはアクション	目的
ステップ 4	**database level** {**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、**database url** コマンドを使用して、データを保管する外部 TFTP サーバーも指定する必要があります。
ステップ 5	**database username** *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	**database archive** {**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-name cn**={*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	**lifetime crl** *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	**lifetime enrollment-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。

	コマンドまたはアクション	目的
		（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。
ステップ 12	**no shutdown** 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
```

証明書サーバーがルータ上でイネーブルになってから、 **show crypto pki server** 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. **configure terminal**
3. **crypto pki server** *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 モードを有効にします。 • 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 server** *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. **configure terminal**
3. **crypto pki server** *cs-label* **rollover request pkcs10 terminal**

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 モードを有効にします。 • 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 server** *cs-label* **rollover request pkcs10 terminal** 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. **configure terminal**
3. **crypto pki export** *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 モードを有効にします。 • 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 export** *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. **crypto pki server** *cs-label* **grant** {**all** | *req-id*}
3. **crypto pki server** *cs-label* **reject** {**all** | *req-id*}
4. **crypto pki server** *cs-label* **password generate** *minutes*
5. **crypto pki server** *cs-label* **revoke** *certificate-serial-number*
6. **crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**base64**| **pem**
7. **show crypto pki server** *cs-label* **crl**
8. **show crypto pki server** *cs-label* **requests**

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 モードを有効にします。 • 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 server** *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	**crypto pki server** *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	**crypto pki server** *cs-label* **password generate** *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。

	コマンドまたはアクション	目的
		（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	**crypto pki server** *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*：次のオプションのいずれかを指定します。 • 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	**crypto pki server** *cs-label* **request pkcs10** {*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。 • **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、証明書をプライバシー強化メール（PEM）ヘッダーなしで返すように指定します。
ステップ 7	**show crypto pki server** *cs-label* **crl** 例： `Device# show crypto pki server mycs 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	**show crypto pki server** *cs-label* **requests** 例： `Device# show crypto pki server mycs 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. **crypto pki server** *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 モードを有効にします。 • 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 server** *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. **configure terminal**
3. **no crypto pki server** *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 モードを有効にします。 • 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	**no crypto pki server** *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. **debug crypto pki server**
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 モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debug crypto pki server** 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 **database url** コマンドを入力した場合は、このコマンドを使用して、証明書サーバー自動

	コマンドまたはアクション	目的
		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. **crypto pki certificate chain**
2. **crypto pki server info requests**
3. **show crypto pki certificates**
4. **show crypto pki server**
5. **show crypto pki trustpoints**

手順の詳細

ステップ 1 **crypto pki certificate chain**

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 **crypto pki server info requests**

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 show crypto pki certificates

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 show crypto pki server

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 show crypto pki trustpoints

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 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、**crypto pki server info requests** 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、**crypto pki server remove** 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、**database archive** 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 **database archive** 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 **database archive** 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。**no shutdown** 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） **no shutdown** 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
```

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

```
! 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
3Fn1DcBmlF7O86XUkiSHP4EV0cI6n5ZMzVLx0XAUtdAl1gD94y1V+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 **show crypto pki server command** 出力は、下位の証明書サーバー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、**crypto pki server** コマンドを使用して、サーバー 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。**show crypto pki server** コマンドを実行すると、mycs サーバーの現在の状態と、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 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」
推奨される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 証明書サーバの設定および管理に関する機能情報

翻訳について
このドキュメントは、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。