PKI への登録のための Secure Device Provisioning の設定

この章では、公開キー インフラストラクチャ（PKI）で Secure Device Provisioning（SDP）を使用する方法を説明します。SDP は、Cisco IOS 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。エンドデバイスは、配置やプロビジョニングの時点ではネットワークに直接接続されていたり、されていない場合があります。SDP は、多数のピア デバイスを導入するユーザに（証明書および設定を含む）ソリューションを提供します。

（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。

PKI への登録のための Secure Device Provisioning（SDP）の設定の前提条件

PKI への登録のための SDP の設定

SDP を設定する前に、次の要件を満たす必要があります。

- ペティショナのデバイスとサーバは、互いに IP 接続されている必要があります。
- イントロデューサには、JavaScript をサポートする Web ブラウザが必要です。
- イントロデューサは、クライアントデバイスで特権をイネーブルにしておく必要があります。
- Cisco IOS リリース 12.3(8)T PKI 対応イメージまたは以降のイメージ。

USB トークンを使用した PKI への登録のための SDP の設定

USB トークンを活用してデバイスを SDP にプロビジョニングするには、ご使用の環境が次の要件を満たしている必要があります。

- ペティショナのデバイスとサーバの両方とも、互いにIP接続されている必要があります。
- イントロデューサには、JavaScript をサポートする Web ブラウザが必要です。
- イントロデューサは、クライアントデバイスで特権をイネーブルにしておく必要があります。
- イントロデューサは、ペティショナのデバイスにアクセスできなければなりません。
- イントロデューサは、設定されている場合は、USB トークンと PIN にアクセスできなければなりません。
- Cisco IOS リリース 12.4(15)T PKI 対応イメージまたは以降のイメージ。

（注） Cisco IOS リリース 12.4(15)T 以降のリリースは、USB トークンに保管されたログイン情報を移動できる柔軟性を備えています。ただし、USB トークンの設定に使用したデバイスは 任意の Cisco IOS リリース 12.3(14)T PKI 対応イメージまたは以降のイメージを実行できます。

サービス プロバイダー経由のインターネット接続に対する SDP を使用したデバイスの設定

SDP を活用してインターネットに接続されていないデバイスを設定するには、ご使用の環境が次の要件を満たしている必要があります。

- イントロデューサには、JavaScript をサポートする Web ブラウザが必要です。
- イントロデューサは、クライアントデバイスで特権をイネーブルにしておく必要があります。
- DHCP クライアントおよびPPPoE クライアントをサポートし、LAN または WAN インターフェイスが設定されている Cisco ルータ。
- Cisco IOS リリース 12.4(20)T PKI 対応イメージまたは以降のイメージ。前回の Cisco IOS リリースがいずれかのデバイスで使用されている場合、SDP 機能はデフォルトで以前の Cisco IOS バージョンになります。

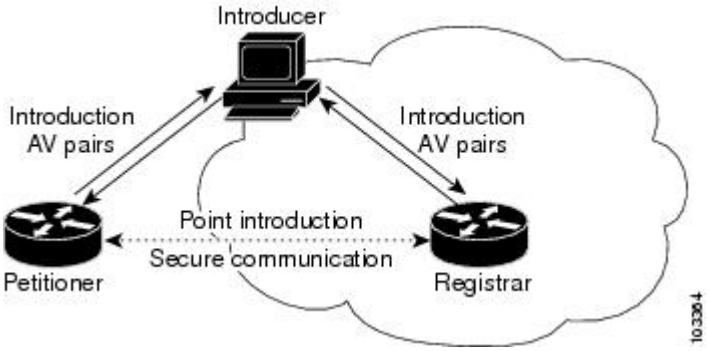
PKI への登録のための Secure Device Provisioning（SDP）の設定に関する情報

SDP の概要

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仮想プライベートネットワーク（VPN）の間といった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 には次の3つのエンティティが必要です（次の図を参照）。

- 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。
  - イントロデューサは、管理イントロデューサとして設定できます。これにより紹介を行っている管理者は、紹介中のデバイスの名前を提供できます。提供されたデバイス名は、通常の SDP メカニズムにおいてイントロデューサの名前のように使用され、SDP 設定の既存機能を保持します。管理イントロデューサの機能の詳細については、「管理イントロデューサの認証リストと認可リスト （15 ページ）」を参照してください。
- ペティショナ：セキュアネットワークに紹介されるクライアント、あるいは新しいデバイス。
- レジストラ：ペティショナを認証するサーバー。レジストラは、証明書サーバの場合があります。

図 *1*：紹介後のセキュア通信



Cisco IOS リリース 12.4(20)T 以降のリリースの時点では、ペティショナにあらかじめインターネット接続を確立しなくても、SDP プロセスを起動できます。予備接続段階と接続段階を利用することで、サービスプロバイダー経由のインターネット接続に対してペティショナを設定できます。予備接続段階と接続段階の詳細については、SDP の機能 （4 ページ）を参照してください。

レジストラは、外部認証、許可、アカウンティング（AAA）サーバと直接通信し、ペティショナのクレデンシャルを確認し、登録を許可または拒否して、特定のペティショナ設定情報を取得します。ペティショナとレジストラは、エンドユーザであるイントロデューサへ Web ページを配信します。ペティショナは、イントロデューサの Web ブラウザを使用してリモート管理システムからブートストラップ設定を受信します。

SDP は、予備接続（任意）、接続、開始（任意）、ようこそ、紹介、および完了の可能な 6 つの段階により Web ブラウザ上に実装されます。各段階は、Web ページを通してユーザに表示されます。各段階の詳細については、SDP の機能 （4 ページ）を参照してください。

SDP の機能

ここでは、SDP が 2 つのデバイス間で PKI を展開する方法について説明します。

SDP プロセスは、イントロデューサにより Web ブラウザにロードされている 3 つの入口ページのいずれかで起動します。3 つの入口ページは、管理者から受信した SDP 予備接続段階、レジストラからロードされた開始段階、ペティショナからロードされたようこそ段階です。

サンプル図では、ローカル デバイス（ペティショナ）をレジストラのセキュア ドメインに紹介する方法を示しています。「イントロデューサ」は、エンドユーザーとも呼ばれます。

SDP 予備接続段階

予備接続ページはオプションです。予備接続ページがない場合、ペティショナは IP 接続を確立しておく必要があります。

管理者は予備接続テンプレートを設定し、予備接続ページをイントロデューサに送信する必要があります。詳細については、デフォルトの予備接続テンプレート （21 ページ） を参照してください。

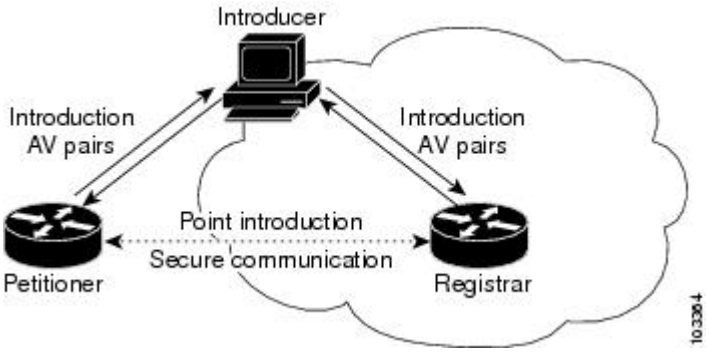
また管理者は、電話、E メール、セキュア E メール、CD、または USB トークンでセキュア ネットワークのユーザ名とパスワードを取得し、イントロデューサに連絡する必要があります。レジストラは、既存の AAA インフラストラクチャ（たとえば、既存の企業ドメインの一部である既存のユーザ名とパスワードのデータベース）を使用してイントロデューサを認証するよう設定できます。SDP 予備接続段階では、一般的な AAA インフラストラクチャで使用されているようなチャレンジパスワードメカニズムがサポートされています。詳細については、SDP による外部 AAA データベースの使用方法 （14 ページ） を参照してください。

予備接続ページを受信後、イントロデューサはそのページを HTTP ブラウザが動作するコンピュータにロードする必要があります。イントロデューサが予備接続ページをローカルファイルとしてHTTPブラウザにロードすると、予備接続ページが表示されます（次の図を参照）。

図 *2 : SDP* 予備接続ページのサンプル

イントロデューサが [Ciscoデバイスにログオン（Log onto Cisco Device）] ボタンをクリックすると、ログインダイアログボックスが表示されます（次の図を参照）。イントロデューサは、シスコデバイスの出荷時デフォルトのユーザ名（cisco）とパスワード（cisco）を入力します。

図 *3 :* ペティショナ ログイン ダイアログボックスのサンプル



イントロデューサはペティショナを認証し、既知の URL にアクセスすることでインターネット接続をテストします。www.cisco.com（198.133.219.25）へのアクセスがデフォルトでテストされます。管理者は、デフォルト予備接続テンプレートを変更することで、URL をテスト接続用に変更できます。デフォルトテスト URL および管理者が予備接続ページに対して設定できるその他フィールドの詳細については、デフォルトの予備接続テンプレート （21 ページ）の項を参照してください。

（注） 予備接続ページに信頼できないレジストラの IP アドレスが含まれるよう変更されたり、予備接続ページが信頼できない発信元から E メール送信される可能性を減らすため、セキュア E メールなどのセキュアな方法を使用して予備接続ページを送信してください。

インターネット接続が確立されると、管理者により定義された予備接続テンプレート設定によって、開始ページまたはようこそページのいずれかが表示されます。インターネット接続が確立されていない場合は、接続ページが表示されます。

SDP 接続段階

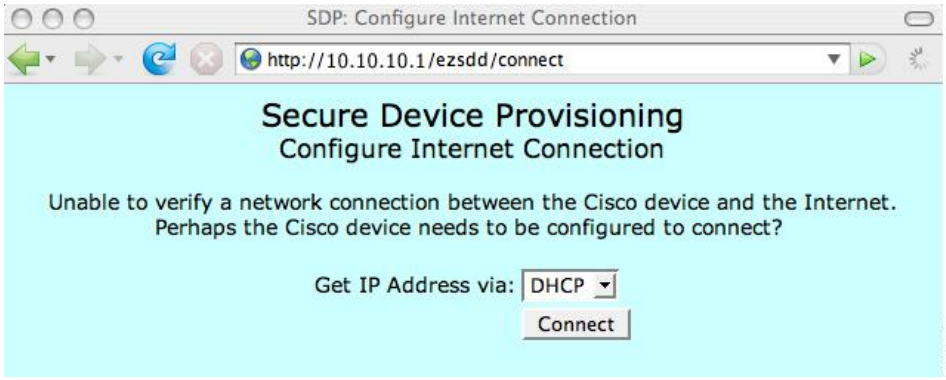
接続ページは、予備接続ページが使用され、予備接続ページの完了時にペティショナの IP 接続がない場合だけ表示されます。接続ページには、Cisco IOS プラットフォームに柔軟性をもたらすため Dynamic Host Configuration Protocol（DHCP）、Point to Point Protocol over Ethernet（PPPoE）、またはスタティック IP アドレス割り当ての 3 つの IP アドレス割り当て方法があります。

（注） インターネット接続を確立する場合、Cisco IOS 設定では SDP 機能は使用されません。SDP 機能には Cisco IOS 設定にシグニチャがあり、送信中の値が変更されないようにします。

DHCP IP アドレス割り当て方法

イントロデューサが IP アドレス割り当て方法オプションとしてデフォルト方法である DHCP を選択すると（次の図を参照）、[接続（Connect）]ボタンをクリックするとペティショナのインターネット接続が設定されます。

図 *4 : DHCP IP* アドレス割り当て方法のサンプル接続ページ



PPPoE IP アドレス割り当て方法

イントロデューサが PPPoE を選択すると、PPPoE ユーザー名およびパスワードの入力フィールドが表示されます（次の図を参照）。イントロデューサは、インターネット サービス プロバイダー（ISP）により提供されたユーザ名とパスワードを入力し、[Connect] ボタンをクリックする必要があります。これによりペティショナのインターネット接続が設定されます。

図 *5 : PPPoE IP* アドレス割り当て方法のサンプル接続ページ

スタティック IP アドレス割り当て方法

イントロデューサがスタティックを選択すると、IP アドレス、ネットマスク、およびデフォルトゲートウェイの入力フィールドが表示されます（次の図を参照）。イントロデューサは、ISP により提供された設定値を入力し、[Connect] ボタンをクリックする必要があります。これによりペティショナのインターネット接続が設定されます。

図 *6 :* スタティック *IP* アドレス割り当て方法の接続ページ

接続ページ IP アドレス設定

IP アドレス設定後、予備接続テンプレートで管理者により設定された既知の URL（デフォルトで www.cisco.com）にアクセスすることで、インターネット接続を再度テストします。これでインターネット接続が確立されると、管理者により定義された予備接続テンプレート設定によって、開始ページまたはようこそページのいずれかが表示されます。インターネット接続が確立されない場合、イントロデューサは入力された設定を確認するか、管理者に連絡します。

SDP スタティック段階

開始ページはオプションです。SDP 交換中に開始ページがない場合、ようこそページで [次へ（Next）]ボタンをクリックすると、ユーザーはレジストラの紹介ページに送信されます。ユーザはまだレジストラに接続していないので、使用可能な資格情報を使用して（レジストラを設定するたびに）レジストラにログインする必要があります。ユーザがログインデータを入力した後では、レジストラに再接続できないブラウザもあります。Cisco IOS Release 12.4(4)T の時点では、ユーザーは、開始ページからレジストラの紹介 URL に連絡することで SDP 交換を開始するようブラウザを設定できます。その後、レジストラはペティショナデバイスにあるようこそページにユーザを送信できます。SDP トランザクションは、このマニュアルに記載されているように、ようこそ段階から紹介段階を経て、完了段階へと続きます。

レジストラから SDP トランザクションを開始するには、**template http start** コマンドを使用してブラウザを設定する必要があります。それ以外の場合、SDP トランザクションはペティショナのようこそページから始まる必要があります。カスタム テンプレートの SDP での動作 （16 ページ）を参照してください。

ようこそページが表示される前に、ユーザは自分のブラウザの開始ページが URL http://registrar/ezsdd/intro を指すように設定する必要があります。ログインダイアログボックスが表示されると、エンドユーザーは、管理者により提供されたユーザー名とパスワードを使用してレジストラにログインし、セキュアネットワークにアクセスできます（次の図を参照）。

図 *7:* レジストラ リモート ログイン ダイアログボックス

有効なユーザー名とパスワードを入力すると、開始ページが表示されます（次の図を参照）。

図 *8:* サンプル *SDP* 開始ページ

ユーザは URL http://10.10.10.1/ezsdd/welcome からペティショナにログインする必要があります。ようこそ段階は、開始ページでユーザが [Next] ボタンをクリックすると開始されます。

SDP ようこそ段階

ローカル ログイン ダイアログボックスが表示されたら（次の図を参照）、エンドユーザーは出荷時デフォルトのユーザー名（cisco）とパスワード（cisco）を使用してローカルデバイスにログインできます。ようこそページが表示されます。

図 *9*：ペティショナ ローカル ログイン ダイアログボックス

パスワードの入力に成功すると、ペティショナにより処理されるようこそ Web ページが表示されます（次の図を参照）。

図 *10*：サンプル *SDP* ようこそページ

ようこそ Web ページでレジストラの URL（例：http://192.0.2.155/ezsdd/intro）を入力し、[Next] ボタンをクリックすると、SDP紹介段階が始まり、レジストラにより処理される紹介ページが表示されます。

SDP 紹介段階

紹介ページを表示する前に、開始ページからまだログインしていない場合、エンドユーザーはレジストラにログインする必要があります（SDP スタティック段階 （8 ページ） を参照）。ここで外部 AAA データベースを利用します。

外部 AAA データベースがある場合、レジストラのイネーブル パスワードを知らなくても、イントロデューサはデータベースのアカウントを使用して紹介を行うことができます。外部AAA データベースがない場合、イントロデューサは認証のためレジストラのイネーブルパスワードを使用できます。

（注） レジストラのイネーブル パスワードを使用すると、パスワードがエンド ユーザに公開されます。したがって、イネーブル パスワードは管理テストの目的でだけ使用することを推奨します。

管理イントロデューサは、紹介ページ（または開始ページ）の HTTP 認証で識別され、AAA データベース クエリーによりユーザの管理特権が戻されます。イントロデューサに管理特権がある場合、デバイス名は管理紹介ページに入力された名前になります。イントロデューサに管理特権がない場合、デバイス名はイントロデューサ名になります。既存のデバイス証明書はペティショナの現在の証明書で、製造識別証明書（MIC）の場合があります。この証明書は存在する場合も、しない場合もあります。外部 AAA データベースの機能の詳細については、SDP による外部 AAA データベースの使用方法 （14 ページ） を参照してください。

エンドユーザーがパスワードの入力に成功したら、紹介 Web ページが表示されます（次の図を参照）。

図 *11* : サンプル *SDP* 紹介ページ

この時点で、レジストラはデバイス情報を外部管理システムに渡し、ブートストラップ設定ファイルを取得します。カスタマイズされたブートストラップ設定ファイルを識別するのに利用可能なオプションの詳細については、カスタム HTML テンプレートの展開ルール （17 ページ）を参照してください。

紹介ページで [Next] ボタンをクリックすると、エンド ユーザは完了段階に入り、自動的に自分のデバイスに戻ります。

SDP 完了段階

エンドユーザーがペティショナをレジストラに登録したので、ペティショナは完了ページを処理します（次の図を参照）。

図 *12* : サンプル *SDP* 完了ページ

これで SDP 交換が完了しました。ペティショナはレジストラから設定情報を受信したため、まもなくレジストラから証明書を受信するはずです。

USB トークンを活用している SDP

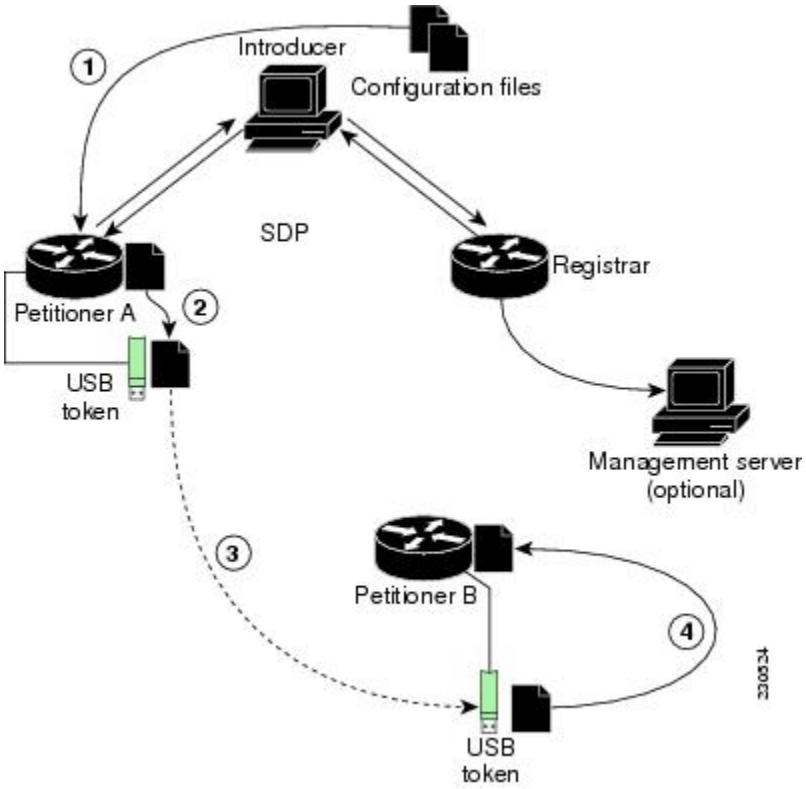
SDP により極めてスケーラブルな配置が実現され、個々のデバイスまたは複数デバイスの配置が簡略化されます。USB トークンによりセキュアな保管と設定の配信が行われます。

Cisco IOS リリース 12.4(15)T 以降の時点では、USB トークンは SDP を使用して PKI ログイン情報を転送する場合に利用でき、SDP は USB トークンの設定に使用できます。USB トークンを使用して、同じ位置にあるデバイスをプロビジョニングしたり、リモートデバイスのプロビジョニングとして使用できる別の場所に USB トークンを転送できます。

USB トークンを使用して PKI ログイン情報を転送する SDP 展開の例を次の図に示します。必要なデバイスとして、USB トークンとデバイスのプロビジョニングに必要な SDP エンティティがあります。これらの SDP エンティティは、イントロデューサ、レジストラ、ローカル位置のペティショナであるペティショナ A、リモート位置でのペティショナであるペティショナ B になります。オプションとして、管理サーバーが使用できます。

（注） オプション設定は、1 台のデバイスをレジストラおよびペティショナ両方として設定することです。これは、USB トークンがリモート位置に転送される場合に利点があります。リモート位置では、個別のペティショナ デバイスは必要ありません。

図 *13 : USB* トークンを使用したクレデンシャル転送の *SDP* 環境例



SDP を使用した USB トークンの設定

SDP 導入の開始前に、USB トークンがペティショナ デバイスに挿入されます。図の設定例では、USB トークンはペティショナ A に挿入されます。ペティショナは、USB トークンにある既存の情報を無視するように設定できます。通常の SDP 操作の場合のように、USB トークンのスケーラブル設定では、テンプレートの初期設定を作成し、適切なターゲット設定情報を備えた各 SDP デバイスに配置する必要があります。

デバイスのプロビジョニングに使用するファイルは、次の順番で移動します。

1. ペティショナの 1 つ、ペティショナ A はローカル位置にあります。ペティショナ A は SDP 交換に直接関わり、USB トークンの初期設定を行います。USB トークン、バイナリ ファイル、テンプレート ファイルの設定に使用するファイルは、レジストラから取得され、ペティショナ A に移動します。

バイナリ ファイル位置の URL は、レジストラで展開されます。バイナリ ファイルは、テンプレート展開機能では処理されません。テンプレート展開はレジストラで、発信元 URL と宛先 URL の両方に対して行われます。

デフォルトでは、バイナリ ファイルとテンプレート ファイルは NVRAM から取得され、それぞれレジストラとペティショナに保管されます。レジストラのバイナリファイル位置とペティショナ A の宛先バイナリファイル位置は、**binary file** コマンドで指定できます。レジストラの

テンプレートファイル位置とペティショナ A の宛先テンプレートファイル位置は、**template file** コマンドで指定できます。

1. Rivest、Shamir、Adelman（RSA）キーおよび証明書チェーン情報は、ペティショナ A から USB トークンに移動します。
2. USB トークンはリモート位置に転送され、ペティショナ B に挿入されます。
3. USB トークンの設定ファイルは、ローカルデバイスのプロビジョニングに使用されます。USB トークンのファイルは、**crypto key move rsa** コマンドでペティショナ B の保管位置に移動できます。

SDP 段階と USB トークン

「SDP の概要」で紹介された同じ SDP フェーズ概念が使用されます。SDP ようこそ段階、SDP 紹介段階、および SDP 完了段階には次のような違いがあります。

SDP ようこそページと USB トークン

ようこそユーザ インターフェイスに接続して紹介が開始される場合、SDP ようこそ段階は通常どおり開始します。USB トークンに既存の証明書がある場合、SDP 交換に署名する場合に使用されます。ローカルな RSA キー ペアではなく、トークンの新しい RSA キー ペアが使用されます。

（注） RSA キー ペアは、キーがトークンで生成される場合、どの場所からでも実質的に 5 分～10 分掛かります。時間の長さは、USB トークンで使用できるハードウェア キー生成により異なります。イントロデューサには、RSA キー ペアが生成されていることを示す情報 Web ページが表示されます。

ペティショナ A で生成された新しいキー ペアは、既存の RSA キー ペアを削除しなくても、USB トークンに追加されます。SDP AV ペアは、トークンが使用中であり、またトークンのセカンダリ設定情報があるかどうか両方を示します。オプションの管理サーバが使用中の場合、AV ペア情報を使用して、特殊なコンフィギュレーション コマンドが必要かどうかを判断します。

SDP 紹介段階と USB トークン

SDP 紹介段階は、レジストラに転送中の AV ペアから開始します。レジストラにより USB トークン関連の AV ペアが検出されると、レジストラがすでに設定されている場合、レジストラは USB トークン宛ての設定情報を作成できます。現在、コンフィギュレーション コマンドは特定の設定ファイルとして送信され、引き続き実行コンフィギュレーションとマージされます。

管理者は通常の SDP コンフィギュレーション コマンドを活用して、USB トークンを設定できます。設定する必要がある USB トークン情報には、証明書、ブートストラップ設定、および PIN 番号設定があります。

SDP 完了段階と USB トークン

完了段階の始めに、紹介はペティショナに転送中の AV ペアに移ります。指定のファイル システム位置には各種ファイルが保管されており、既存の設定ファイル処理が行われます。この順序により、転送された新しいファイルを設定で利用できます。

設定された USB トークンの使用

USB トークンがペティショナ A により設定されたら、その現在位置からリモート位置へと転送されます。リモート位置には、2 番目のペティショナであるペティショナ B が配置されています。USB トークンはターゲットデバイスであるペティショナ B に挿入されます。ペティショナ B では USB トークンの設定と USB トークンの暗号素材が継承されます。リモート位置のエンドユーザには、USB トークンの PIN 番号がなければなりません。PIN 番号は、出荷時デフォルトの PN 番号、または紹介段階中に管理者が設定した PIN 番号のいずれかになります。

SDP による外部 AAA データベースの使用方法

外部 AAA データベースは、SDP 交換中に 2 回アクセスされます。AAA データベースへの最初のアクセスでは、イントロデューサが認証されます。つまりレジストラで、セキュア HTTP（HTTPS）サーバー経由で紹介要求が受信されると、イントロデューサのユーザー名とパスワードに基づいて AAA 検索が行われ、要求が許可されます。AAA データベースへの 2 番目のアクセスでは、認証情報が取得され、ペティショナデバイスに発行された設定および証明書に適用されます。つまり、レジストラはペティショナが署名している証明書を使用して要求シグニチャが完全であることを確認します。証明書の題名は AAA データベースで指定でき、最大 9 つの設定テンプレート型変数を指定し、テンプレート設定にまで展開できます。

自己署名証明書と別の CA サーバにより発行された証明書の使用

デフォルトでは、SDP 交換の実施結果では、ペティショナ デバイスに証明書が 1 枚だけ発行されます。発行される証明書は 1 枚だけですが、イントロデューサでは複数デバイスを紹介し、複数の証明書を取得する際の制限はありません。発行されている証明書の題名を指定することで、イントロデューサに関連しているすべての証明書がこのように発行されていることを保証できます。PKI AAA 統合により、さらにこれらの証明書の使用を制限できます。さらに、ユーザごとに 1 つだけ認証および認可の要求を許可するよう、AAA データベースを設定できます。

ペティショナ証明書は自己署名されているため、ペティショナの公開キーを伝送するためだけに使用されます。証明書に対する確認チェックや認可チェックは行われません。したがって、認可はユーザごとに行われ、デバイス単位の情報は使用されません。

デバイス単位の認可を使用した方が好ましい場合もあります。したがって、ペティショナが SDP トランザクションのために他の認証機関（CA）サーバにより発行された証明書を使用できる場合、既存の PKI を使用でき、その証明書属性に対して認可を受けることができます。

証明書を使用して認可を受けるためにペティショナとレジストラを設定すると、展開中の特定のデバイスの認可が受けられます。以前は、イントロデューサとペティショナ間の通信は、イントロデューサとペティショナデバイス間の物理的なセキュリティだけでその安全が確保され

ていました。SDPの証明書を使用した認可では、レジストラは紹介を受け入れる前に、現在のデバイス ID を確認できる機会があります。

SDP の認証および認可リスト

SDP レジストラを設定している場合に認証リストと認可リストを指定すると、レジストラではイントロデューサのすべての要求に対して、指定のリストが使用されます。認証リストは、イントロデューサを認証する場合に使用されます（AAA サーバでユーザ名とパスワードを確認して、アカウントが有効かどうか確認されます）。認可リストは、証明書題名およびペティショナに返信される Cisco IOS コマンドライン インターフェイス（CLI）スニペットに展開されるテンプレート型変数のリストの該当認可フィールドを受信する場合に使用されます。認証リストと認可リストは通常、同じ AAA サーバ リストを指しますが、認証と認可に異なるデータベースを使用できます（異なるデータベースへのファイルの保管は推奨しません）。

ペティショナが紹介要求をする場合、複数の照会が RADIUS サーバまたは TACACS+ サーバ上の AAA リスト データベースに送信されます。照会により、次の形式のエントリが検索されます。

```
user Password <userpassword>
   cisco-avpair="ttti:subjectname=<<DN subjectname>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#=<<value>>"
```

（注） 有効な AAA ユーザ名レコードさえあれば、認証チェックを通過できます。「cisco-avpair=tti」情報は、認可チェックの場合だけ必要です。

認可応答で題名を受信した場合、SDP レジストラによりその題名は登録データベースに保管され、「subjectname」は、ペティショナデバイスからの以降の証明書要求（PKCS10）で提供される題名より優先されます。

番号が付けられた「tti:iosconfig」値は、ペティショナに送信される SDP Cisco IOS スニペットに展開されます。設定により、あらゆる番号付き（$1 ～ $9）のテンプレート型変数が置き換えられます。デフォルト Cisco IOS スニペットテンプレートには変数 $1 ～ $9 が含まれていないため、外部 Cisco IOS スニペットテンプレートを設定しない限り、これらの変数は無視されます。外部設定を指定するには、**template config** コマンドを使用します。

（注） テンプレート設定位置には、変数「$n」が含まれている場合があります。この変数はユーザーがログインに使用した名前に展開されます。

管理イントロデューサの認証リストと認可リスト

SDP メカニズムでは、イントロデューサとデバイス間に永続的関係があることを前提としています。その結果、イントロデューサのユーザ名はデバイス名の定義に使用されます。

SDP配置シナリオの中には、イントロデューサが多数のデバイスの紹介を行う、管理者の場合があります。ただし、イントロデューサ（管理者）名を使用してデバイス名を定義すると、複数のデバイスのデバイス名が同じになり、正しく配置されなくなります。代わりに、管理イントロデューサを使用すれば、管理者は紹介中に正しいデバイス名を指定できます。

一般的に言えば、イントロデューサのユーザ名がデータベース レコード ロケータとして使用され、Cisco IOS 設定テンプレート、（AAA データベースから取り出され、テンプレートに展開される）各種テンプレート型変数、およびデバイスに発行されたPKI証明書の該当する題名など、デバイスに関する他のすべての情報が決定されます。簡単にするため、データベース レコード ロケータはユーザ名またはデバイス名と呼びます。

管理イントロデューサは、デバイス名を提供します。そのようにして、管理者は紹介を行う場合に適切なレコード ロケータを提供できます。たとえば、管理者がユーザー名「user1」のデバイスを紹介しようとしている場合、管理者はそのデバイスをPKIネットワークに紹介し、管理者自身のログイン情報を使用してレジストラにログインした後に、user1 をレコードロケータとして提供します。レコード ロケータ user1 がデバイス名になります。紹介に固有の他のすべてのテンプレートおよびPKI 証明書の題名に関する情報が、管理者のレコードではなく、user1 ユーザー名レコードにより提供されます。

レジストラ デバイスでは、ユーザ イントロデューサ名とともに、提供されたユーザ名情報が使用されます。ユーザー名により既存のメカニズムで、変更なくサポートする必要があるユーザーの認可、テンプレート、およびPKI 証明書の情報が判断できます。

カスタム テンプレートの SDP での動作

カスタム テンプレートを使用して、SDP プロセスを簡略化できます。

- カスタム テンプレートにより、Web ページに必要な開始情報を記入できるため、イントロデューサはレジストラに連絡する必要がなくなり、SDP トランザクションを即座に開始できます。
- カスタム テンプレートにより、カスタマイズされた展開情報を Web ページに表示できるため、ユーザに合せてユーザ エクスペリエンスを調整できます。

デフォルト テンプレートを変更すると、カスタム テンプレートを簡単に定義できます。カスタム テンプレートがない場合、イントロデューサは SDP トランザクションが開始できるための情報をレジストラに問い合わせる必要があります。デフォルトテンプレートのリストについては、SDP トランザクション Web ページのデフォルト テンプレート （21 ページ） の項を参照してください。

（注） カスタムテンプレートを設定するのは、上級の SDP ユーザーだけに限定することを推奨します。テンプレートがイントロデューサのブラウザに表示される前に、テンプレートを誤って変更してしまった場合に問題が発生するおそれがあるためです。

カスタム テンプレート型変数の展開

テンプレートには、Cisco IOS SDP レジストラまたはペティショナにより置き換えられる展開変数があります。これらの変数は、次のように展開されます。

- $$：「$」
- $a：属性と値（AV）のペア
- $c：信頼できる証明書
- $d：ブラウザのダンプ AV ペア
- $h：ホスト名
- $k：キーラベルまたは「tti」
- $l：トラストポイントラベル＝「tti」
- $n：HTTP クライアントのユーザー名
- $s：TTI キーのデフォルトサイズ
- $t：トラストポイント設定
- $u：完了 URL
- $1 ～ $9：ユーザー認証中に AAA サーバーから取得された変数

カスタム テンプレート型変数の展開ルール

設定とテンプレートはSDP交換中に使用されます。使用前および配布後、これらのテンプレートは、SDP 通信段階に基づき、次のルールで展開されます。

カスタム HTML テンプレートの展開ルール

HTML テンプレートは HTTP クライアントに送信される前に、即座に展開されます。HTTP テンプレートは次のように展開されます。

- $u：SDP 完了 URL（例：http://10.10.10.1/ezsdd/completion）が入力される完了 URL。この変数は、内部「ウィザード」状態として SDP により内部的に使用されます。通常のウィザード処理のため、SDP 紹介ページには「<FORM action=\“$u\”method=\“post\”>」といったようなテキストが含まれている場合があります。
- $n：管理イントロデューサにより入力されたイントロデューサ名またはデバイス名。
- $$：$
- $h：ホスト名
- $a：指定のテンプレート文字があるないにかかわらずすべての AV ペアは、次の HTML フォーム形式に書き出されます（これらの AV ペアは「INPUT type=hidden」でないため、テンプレートまたはSDPプロセスのデバッグのためにWebページに直接表示されます）。

<INPUT type=hidden NAME="attribute string here"

value="variable string here">

すべての HTML テンプレートに以下のラインがなければなりません。

$d = dump all av pairs in: attribute = value

URL テンプレートの展開ルール

設定テンプレートの発信元、ファイル テンプレートの発信元、およびファイル宛先には URL が存在します。これらの変数は、レジストラが URL を作成するとき、つまり設定またはファイルを取得する直前に展開されます。ファイル宛先については、これらの変数は、ペティショナによりファイルがファイル宛先にコピーされる直前に展開されます。

- $$: $
- $h : ホスト名

iPhone の導入に関する URL テンプレートの展開ルール

iPhone を導入するために、次のテンプレート展開変数が導入されました。

- $o : チャレンジパスワード。このテンプレート文字は、SDP レジストラが Simple Certificate Enrollment Protocol（SCEP）サーバからチャレンジ パスワードを取得した後、開始段階で設定プロファイルが iPhone に送信される前に、SDP レジストラによって展開されます。
- $i : iPhone の固有デバイス識別子（UDID）。このテンプレート文字は、紹介段階で設定プロファイルが iPhone に送信される前に、SDP レジストラによって所有者名の CN フィールドに展開されます。
- $p : 所有者名の差別化要因。このテンプレート文字は、CLI によって設定された値を使用して SDP レジストラによって展開されます。詳細については、Apple iPhone を導入するための SDP レジストラの設定 （39 ページ）を参照してください。この値は、SCEP サーバが iPhone に対して発行する 2 つの証明書を区別するために使用されます。1 つの証明書は完了段階で発行され、もう 1 つの証明書は VPN 確立段階で発行されます。この値を挿入する所有者名の部分またはフィールドを決定します。

詳細については、PKI で SDP が Apple iPhone を導入する方法 （24 ページ） を参照してください。

カスタム設定およびファイルのテンプレート型変数の展開ルール

カスタム設定とファイルのテンプレート型変数は両方とも、レジストラが設定またはファイルのテンプレートを作成する場合、またペティショナが設定またはファイルのテンプレートを受信する場合に展開されます。

カスタム設定とファイルのテンプレート型変数のレジストラでの展開ルール

レジストラが設定またはファイルのテンプレートを展開する場合、Cisco IOS CA により次の変数が使用されます。これらの変数は、SDP ウィザードで送信前に展開されます。

- $$：$
- $h：ホスト名
- $t：クライアントで展開されるよう $l、$k、および $ を組み込んだ単純なトラストポイントデフォルト設定
- $1 ～ $9：ユーザー認証中に AAA サーバーから取得された変数（ファイルテンプレートには適用されない）

カスタム設定とファイルのテンプレート型変数のペティショナでの展開ルール

ペティショナが設定またはファイルのテンプレートを展開する場合、次の変数が展開されます。

- $$：$
- $h：ホスト名
- $k：キーラベル
- $l：トラストポイントラベル
- $s：キーのサイズ
- $c：証明書チェーンに展開
- $n：ユーザー名に展開（ファイルテンプレートには適用されない）

カスタム設定 HTTP テンプレート型変数の展開ルール

カスタム設定 HTTP テンプレートにより、バックエンド コモン ゲートウェイ インターフェイス（CGI）スクリプトに柔軟性が与えられ、外部管理システムと統合されます。テンプレート URL は、レジストラが外部管理システムからブートストラップ設定を受信する前に、HTTP テンプレートを展開することで実行されます。デバイス情報に基づいて特定のブートストラップ設定ファイルが見つかるようにするため、デバイス名（$n）は URL に展開され、外部管理システムへと渡されます。

（注） 表示される HTML テキストの変更だけ行う必要があります。既存の展開変数、Javascript、およびデフォルトテンプレートの形式は、テンプレートのカスタマイズ時には削除しないでください。これらのは SDP が正しく動作するために必要な情報です。

HTTP テンプレートの展開と **template config** コマンドにより、次のいずれかのファイルタイプを指定して、カスタマイズ ブートストラップ構成ファイルを取得できます。

- デバイス名を使用した構成ファイル（例：template config http://myserver/$n-config-file.conf）
- デバイス名を使用した CGI スクリプト（例：template config http://myserver/cgi-bin/mysdpcgi post）

Cisco IOS リリース 12.4(6)T の時点で、ブートストラップ設定がデバイス名だけでなく、タイプ、Cisco IOS 現行バージョン情報、および現行の設定で識別できるよう CGI サポートが拡張されました。この機能では、**post** キーワードにより **template config** コマンドが拡張されています。このキーワードはレジストラに、HTTP または HTTPS プロトコルだけを使用した CGI スクリプトによってこの追加デバイス情報を外部管理システムに送信するよう指示します。

レジストラにより、AVペア（$a）を使用してデバイス情報が外部管理システムに渡されます。AV ペア情報を使用して、管理システムは適切なブートストラップ設定ファイルを識別し、レジストラに返信します。カスタマイズ ブートストラップ構成ファイルを識別するため、拡張 CGI サポートにより送信される追加 AV ペアを次の表に示します。

表 *1: HTTP* ポスト中に外部管理システムに送信される *AV* ペア

AV ペア	**説明**
TTIFixSubjectName	AAA_AT_TTI_SUBJECTNAME（レルム認証ユーザがレジストラでルート ユーザでない場合だけ送信）
TTIIosRunningConfig	**show running-config brief** の出力
TTIKeyHash	デバイス公開キー上で計算されるダイジェスト
TTIPrivilege	AAA_AT_TTI_PRIVILEGE：ユーザーが管理者の場合は「admin」、ユーザーが管理者でない場合は「user」が送信されます（レルム認証ユーザーが管理者で AAA サーバーから情報が利用できる場合だけ送信）
TTISignature	UserDeviceName および TTISignCert を除く AV ペアすべてで計算されるダイジェスト
TTISignCert	デバイスの現在の証明書（デバイスに現在証明書がある場合だけ送信）
TTITemplateVar	AAA_AT_TTI_IOSCONFIG（1-9）（レルム認証ユーザがレジストラでルート ユーザでない場合だけ送信）
TTIUserName	デバイス名
TTIVersion	TTI バージョンのレジストラ
UserDeviceName	管理イントロデューサにより入力されたデバイス名（レルム認証ユーザが管理者の場合だけ送信）

（注） レジストラでは Cisco IOS リリース 12.4（6）T が実行され、**template config** コマンドは **post** キーワードを指定して発行する必要があります。また、*url* 引数には HTTP または HTTPS のいずれかが含まれている必要があります。拡張CGIテンプレート機能にはその他のプロトコルはサポートされていません（例：FTP）。

SDP トランザクション Web ページのデフォルト テンプレート

SDP トランザクション Web ページにはそれぞれ、次のデフォルトテンプレートが存在します。

デフォルトの予備接続テンプレート

予備接続テンプレートは、ユーザの環境に応じた値を含めるよう、管理者により変更できます。予備接続ページのフォーマットも、テンプレートに含まれている設定により変更できます。

管理者がカスタマイズする必要があるレジストラの IP アドレスを除き、予備接続テンプレートは次に示すように使用できます。

```
<html><head><title>
SDP: Test Internet Connection</title></head>
<noscript><b>
If you see this message, your browser is not running JavaScript,<br>
which is required by Cisco Secure Device Provisioning.<br>
If you cannot enable JavaScript, please contact your system administrator.
<br><br></b></noscript>
<body style="background-color: rgb(204, 255, 255);">
<div style="text-align: center;"><big><big>
Secure Device Provisioning</big><br>
Test Internet Connection</big><br><br>
<form action="http://10.10.10.1/ezsdd/connect" method="post">
<input type="submit" value="Log onto Cisco Device"><br><br>
Default username/password is cisco/cisco.
<input type="hidden" name="TTIAfterConnectURL" value="http://10.10.10.1/ezsdd/welcome">
<!-- Note, that for the below, 198.133.219.25 = www.cisco.com. -->
<input type="hidden" name="TTIConnectTestURL" value="http://198.133.219.25">
<input type="hidden" name="TTIInsideAddr" value="10.10.10.1">
<input type="hidden" name="TTIlanport" value="Vlan1">
<input type="hidden" name="TTIwanport" value="FastEthernet4">
</form></div></body></html>
```

非表示 HTML 形式フィールド

非表示 HTML 形式フィールドにより、初期設定情報が管理者により設定されたとおりにブラウザに送信されますが、署名はされていません。

（注） 「非表示」という用語は、イントロデューサができるだけ混乱しないよう、これらの HTML 形式フィールドが予備接続ページに表示されないことを示します。

管理者は、次の表に示すように、予備接続テンプレートの非表示 HTML 形式フィールドを設定できます。

表 *2:* 予備接続段階中に送信される管理者が定義した *AV* ペア

AV ペア	説明
TTIAfterConnectURL	管理者は、TTIAfterConnectURL フィールドをようこそページの URL または開始ページの URL のいずれかに設定できます。ようこそページの URL は、ペティショナの出荷時デフォルト IP アドレスに指定されています。接続後 URL は、インターネット接続確立後に SDP が使用されない場合に任意の有効な URL になることができます。
TTIConnectTestURL	管理者は、TTIConnectTestURL フィールドを、インターネット接続確立時にアクセスできるはずの有効な URL に設定できます。予備接続テンプレートのデフォルト値は、www.cisco.com（198.133.219.25）です。
TTIInsideAddr	管理者は、TTIInsideAddr フィールドをペティショナの出荷時デフォルト IP アドレスに設定できます。Cisco 871 ISR の場合は、IP アドレスは 10.10.10.1 です。
TTIlanportx	管理者は、TTIlanportx フィールドをペティショナ プラットフォームの LAN インターフェイスに設定できます。このフィールドは、Cisco IOS 接続設定の適用に使用できます。Cisco 871 の場合は、フィールド値は「Vlan1」になります。
TTIwanport	管理者は、TTIwanport フィールドをペティショナの WAN インターフェイス名に設定できます。このフィールドは、Cisco IOS 接続設定の適用に使用できます。Cisco 871 の場合は、フィールド値は「FastEthernet4」になります。

（注） 接続テンプレートはカスタマイズできません。

デフォルト開始ページ テンプレート

```
<html><head><title>EZ-Secure Device Deployment Start page on $h</title></head>
<NOSCRIPT><B>
If you see this message, your browser is not running JavaScript.<BR>
Cisco Secure Device Deployment requires JavaScript.<BR> Please contact
your system adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function submit_to_url(form){
form.action=form.TTIWelcomeURL.value;return true;}</SCRIPT>
<B>Welcome to Cisco Secure Device Deployment Server $h</B> <FORM  action="" method="post"
 onSubmit="return submit_to_url(this)"> Your
device:<BR> <INPUT type="text" name="TTIWelcomeURL" size=80  value=\"\"><BR><BR> <INPUT
 type="submit" value="Next"><BR>
$a</FORM></html>
```

デフォルトようこそページ テンプレート

```
<html><head><title>EZ-Secure Device Deployment WELCOME to $h</title></head>
<NOSCRIPT><B>
If you see this message, your browser is not running JavaScript.<BR>
Cisco Secure Device Deployment requires JavaScript.<BR> Please contact
your system adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE=\"JavaScript\">
function submit_to_url(form){
natURL=location.href.split(\"/\");
localURL=form.TTICompletionURL.value.split(\"/\");
if(natURL[2]!=localURL[2]){
form.TTICompletionURL.value=localURL[0]+\"//\"+natURL[2]+\"/
\"+localURL[3]+
\"/\"+localURL[4];}
form.action=form.vpnserviceurl.value;
return true;}</SCRIPT>
<B>Welcome to Cisco Secure Device Deployment for $h</B> <FORM  action=\"\" method=\"post\"
 onSubmit=\"return submit_to_url (this)\">
To join a Virtual Private Network (VPN) enter the web<BR> site URL
provided by your network administrator:<BR> <INPUT type=\"text\"  name=\"vpnserviceurl\"
 size=80 value=\"\"><BR><BR><INPUT  type=\"submit\" value=\"Next>\"><BR> $a</FORM></html>
```

デフォルト紹介ページ テンプレート

```
<html><head><title>EZ-Secure Device Deployment INTRODUCTION to $h</title>
</head><B>Welcome to the VPN network gateway on $h</B> <FORM  action=\"$u\"
method=\"post\"> Your 'username' and 'password' entered
have been accepted.<BR> Your device will now be allowed to
automatically join the VPN network.<BR> <BR>Press Next to complete
automatic configuration of your VPN Device.<BR> <BR><INPUT  type=\"submit\"
value=\"Next>\"><BR> $a</P></FORM></html>
```

デフォルト管理紹介ページ テンプレート

```
<html><head><title>EZ-Secure Device Deployment ADMINISTRATIVE
INTRODUCTION to $h</title></head> <NOSCRIPT><B> If you see this
message, your browser is not running JavaScript.<BR> Cisco Secure
Device Deployment requires JavaScript.<BR> Please contact your system
adminstrator.<BR><BR></B></NOSCRIPT>
<SCRIPT LANGUAGE=\"JavaScript\">
function submit_to_url(form){
form.introadminurl.value=location.href+\"/admin\";
form.action=form.introadminurl.value;
return true;}</SCRIPT>
<B>Welcome to the VPN network gateway on $h</B> <FORM action=\"\"  method=\"post\"
onSubmit=\"return submit_to_url (this)\"> Your
administrator 'username' and 'password' entered have been
accepted.<BR> Please provide the name to be associated with this
device:<BR> <INPUT type=\"text\" name=\"userdevicename\" size=64  value=\"\"><BR><BR>
<INPUT type=\"submit\" value=\"Next>\"><BR> <INPUT  type=\"hidden\" name=\"introadminurl\"
 value=\"\"><BR>
$a</FORM></html>
```

デフォルト完了ページ テンプレート

```
<html><head><title>EZ-Secure Device Deployment COMPLETE on $h</title></head>
<B>Now enrolling $h with the VPN network...</B><BR> Full network VPN
access should be available in a moment.<BR><BR> $d<BR></html>
```

設定ファイルのデフォルト テンプレート

デフォルト設定のテンプレートを示します。このデフォルト設定ファイルは、設定テンプレートが指定されていない、または **template config** コマンドが **post** キーワードを指定せずに発行されている場合に使用されます。デフォルト設定テンプレートの使用方法の詳細については、UsingaConfigurationTemplateFile の例 （55 ページ）を参照してください。

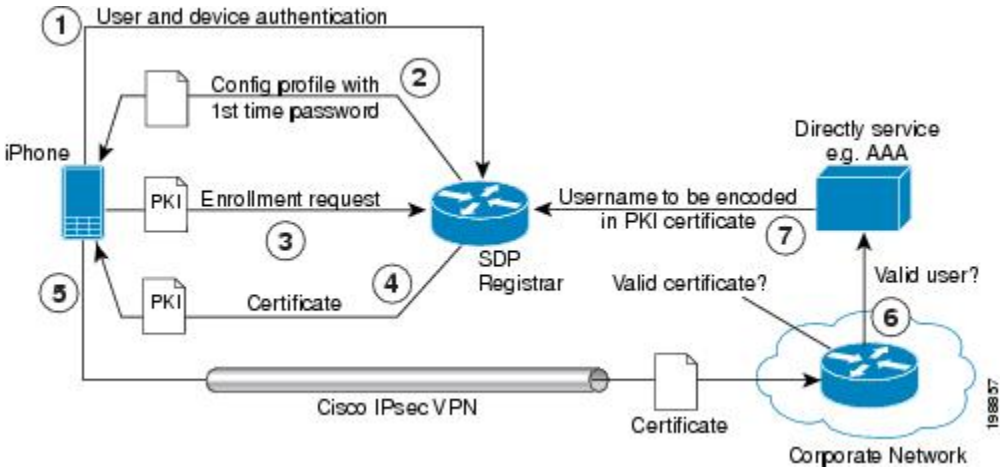
```
$t
!
$c
!
end
```

PKI で SDP が Apple iPhone を導入する方法

Cisco IOS 15.1(2)T および Apple iPhone OS 3.0 リリースが導入されたため、Cisco IOS ネットワーク デバイスで Apple iPhone がサポートされるようになりました。Cisco IOS ルータは SDP レジストラを使用して iPhone を導入し、IPSec VPN、SCEP サーバ、および PKI 証明書の導入テクノロジーを使用してネットワーク アプリケーションに安全にアクセスできるようにします。

Apple iPhone では、XML ベースの「設定プロファイル」の配布と証明書の初期導入を組み合わせて実行します。SDP はこれらの初期の証明書を使用してエンタープライズ アプリケーションへのアクセスを認証し、その後のプロファイルの配布を暗号化します。SDP は、iPhone にデジタル証明書を配布する際に、この登録ソリューションを使用します。

図 *14 : PKI* での *SDP* レジストラによる *iPhone* の導入



PKI での SDP レジストラによる Apple iPhone の導入段階

ここでは、PKI で SDP レジストラが iPhone を導入する場合の各段階について説明します。

SDP 導入開始段階

次のステップでは、SDP 導入開始段階について説明します。

（注） SDP 導入開始段階は、『Apple iPhone Enterprise Deployment Guide』（http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf）で説明する「Begin Enrollment」段階（またはフェーズ 1）と同じです。

手順の概要

1. iPhone ユーザは Safari ブラウザを開き、開始ページの HTTPS URL を入力します。たとえばこの HTTPS URL は、社内のネットワーク アドレスなどです。SDP レジストラの HTTPS ページによってプロセスが開始されます。
2. ユーザは、ユーザ名とパスワードを入力して Cisco ルータとの認証を開始します。Cisco ルータは SDP レジストラとして動作します。
3. SDP レジストラは SCEP サーバに接続し、チャレンジ パスワードを取得します。
4. SDP レジストラは、チャレンジ パスワード、SCEP サーバの URL、および iPhone 属性の要求で構成される設定プロファイルを XML 形式で作成します。SCEP サーバの URL は登録要求の送信に使用され、iPhone デバイスの属性は RSA キーを生成する際に iPhone によって使用されます。
5. iPhone ユーザは、設定ファイルを iPhone にインストールして、SDP 開始段階を終了します。

手順の詳細

ステップ 1 iPhone ユーザは Safari ブラウザを開き、開始ページの HTTPS URL を入力します。たとえばこの HTTPS URL は、社内のネットワーク アドレスなどです。SDP レジストラの HTTPS ページによってプロセスが開始されます。

ステップ 2 ユーザは、ユーザ名とパスワードを入力して Cisco ルータとの認証を開始します。Cisco ルータは SDP レジストラとして動作します。

ステップ 3 SDP レジストラは SCEP サーバに接続し、チャレンジ パスワードを取得します。

ステップ 4 SDP レジストラは、チャレンジ パスワード、SCEP サーバの URL、および iPhone 属性の要求で構成される設定プロファイルを XML 形式で作成します。SCEP サーバの URL は登録要求の送信に使用され、iPhone デバイスの属性は RSA キーを生成する際に iPhone によって使用されます。

次の例は、SDP 導入開始段階で SDP レジストラが iPhone に送信する設定プロファイルを示しています。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>PayloadContent</key>
<dict>
<key>URL</key>
<string>https://profileserver.example.com/iphone</string>
<key>DeviceAttributes</key>
```

```
<array>
<string>UDID</string>
<string>IMEI</string>
<string>ICCID</string>
<string>VERSION</string>
<string>PRODUCT</string>
</array>
<key>Challenge</key>
<string>optional challenge</string>
```

ステップ 5 iPhone ユーザは、設定ファイルを iPhone にインストールして、SDP 開始段階を終了します。

SDP 導入ようこそ段階

SDP 導入ようこそ段階は iPhone には適用されません。これは、イントロデューサ（Safari Web ブラウザなど）が SDP ペティショナ（iPhone）で実行されるためです。

SDP 導入紹介段階

次のステップでは、SDP 導入紹介段階について説明します。

（注） SDP 導入紹介段階は、「デバイス認証」段階に相当します。

手順の概要

1. iPhone は、要求されたデバイス属性情報とチャレンジ パスワードを含む HTTPS POST を設定プロファイルとしてトリガーします。HTTPS POST は、SDP 導入開始段階で取得した設定プロファイルに指定されている HTTPS の URL に送信されます。これは、SDP 導入紹介段階の URL である必要があります。POST データは Apple 社が発行した証明書（組み込みの ID）を使用して iPhone によって署名されます。そしてこの署名が確認され、ID が確認され、デバイス属性が確認されます。
2. iPhone によって送信された UDID は SDP レジストラによって取得され、所有者名に追加されます。その後、SDP レジストラによって取得されたデバイス属性は、これが本当に受け入れられるデバイスのタイプであるかどうかを判断するために使用されます。たとえばネットワーク管理者は、3GS の iPhone のみをネットワークで使用できるように許可します。これは、iPhone 3GS にはハードウェアに暗号化された保管場所があるためです。取得されたデバイス属性によって、SDP レジストラは 3GS の iPhone と 3G の iPhone を区別できます。
3. SDP レジストラは、SCEP サーバの HTTP URL、登録要求で送信される所有者名（UDID を含む）、キーのサイズ、キーのタイプ、キーの使用状況、およびチャレンジパスワードで構成された設定プロファイルを作成して応答します。開始段階がスキップされた場合、SDP レジストラは SCEP サーバに接続し、チャレンジパスワードを取得します。SDP レジストラが所有者名とチャレンジパスワードを取得する方法の詳細については、iPhone の導入に関する URL テンプレートの展開ルール （18 ページ）を参照してください。

手順の詳細

ステップ 1 iPhone は、要求されたデバイス属性情報とチャレンジパスワードを含む HTTPS POST を設定プロファイルとしてトリガーします。HTTPS POST は、SDP 導入開始段階で取得した設定プロファイルに指定されている HTTPS の URL に送信されます。これは、SDP 導入紹介段階の URL である必要があります。POST データは Apple 社が発行した証明書（組み込みの ID）を使用して iPhone によって署名されます。そしてこの署名が確認され、ID が確認され、デバイス属性が確認されます。

ステップ 2 iPhone によって送信された UDID は SDP レジストラによって取得され、所有者名に追加されます。その後、SDP レジストラによって取得されたデバイス属性は、これが本当に受け入れられるデバイスのタイプであるかどうかを判断するために使用されます。たとえばネットワーク管理者は、3GS の iPhone のみをネットワークで使用できるように許可します。これは、iPhone 3GS にはハードウェアに暗号化された保管場所があるためです。取得されたデバイス属性によって、SDP レジストラは 3GS の iPhone と 3G の iPhone を区別できます。

次の例は、SDP 導入紹介段階で iPhone が送信する設定プロファイルを示しています。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
  DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>UDID</key>
<string></string>
<key>VERSION</key>
<string>7A182</string>
<key>MAC_ADDRESS_EN0</key>
<string>00:00:00:00:00:00</string>
<key>CHALLENGE</key>
either:
    <string>String</string>
or:
    <data>"base64 encoded data"</data>
</dict>
</plist>
```

ステップ 3 SDP レジストラは、SCEP サーバの HTTP URL、登録要求で送信される所有者名（UDID を含む）、キーのサイズ、キーのタイプ、キーの使用状況、およびチャレンジパスワードで構成された設定プロファイルを作成して応答します。開始段階がスキップされた場合、SDP レジストラは SCEP サーバに接続し、チャレンジパスワードを取得します。SDP レジストラが所有者名とチャレンジパスワードを取得する方法の詳細については、iPhone の導入に関する URL テンプレートの展開ルール （18 ページ）を参照してください。

（注） SDP レジストラでは RSA のキー タイプのみをサポートしています。

次の例は、SDP 導入紹介段階で SDP レジストラが送信する設定プロファイルを示しています。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
```

```
<dict>
<key>PayloadContent</key>
<dict>
<key>URL</key>
<string>https://iphone.vpn.apple.com/pkifoobar.exe</string>
<key>Name</key>
<string>instance_for_getcacert_call</string>
<key>Subject</key>
<array>
<array>
<array>
<string>O</string>
<string>Apple Inc.</string>
</array>
</array>
<array>
<array>
<string>CN</string>
<string>Foo</string>
</array>
</array>
</array>
<key>Challenge</key>
<string>CHALLENGE</string>
<key>Keysize</key>
<integer>1024</integer>
<key>Key Type</key>
<string>RSA</string>
<key>Key Usage</key>
<integer>5</integer>
</dict>
<key>PayloadDescription</key>
<string>Provides device encryption identity</string>
<key>PayloadUUID</key>
<string>fd8a6b9e-0fed-406f-9571-8ec98722b713</string>
<key>PayloadType</key>
<string>com.apple.security.scep</string>
<key>PayloadDisplayName</key>
<string>Encryption Identity</string>
<key>PayloadVersion</key>
<integer>1</integer>
<key>PayloadOrganization</key>
<string>Apple Inc.</string>
<key>PayloadIdentifier</key>
<string>com.apple.encrypted-profile-service</string>
</dict>
</plist>
```

SDP 導入ポスト紹介段階

次のステップでは、SDP 導入ポスト紹介段階について説明します。

（注） SDP 導入ポスト紹介段階は、『http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdfApple iPhone Enterprise Deployment Guide』で説明する「Certificate Installation」段階（またはフェーズ 3）と同じです。

手順の概要

1. iPhone は、SDP 導入紹介段階で SDP レジストラから取得した SCEP 情報を含む設定プロファイルの指定をインストールします。
2. iPhone はプロファイルに指定された指示に従ってキーを生成し、HTTP URL がプロファイルに指定されている SCEP サーバに登録要求とチャレンジ パスワードを送信します。
3. SCEP サーバはチャレンジ パスワードを確認し、iPhone にデジタル証明書を発行します。
4. ユーザはこの証明書を iPhone にインストールし、Cisco IPsec VPN を使用して会社のネットワークに接続できます。

手順の詳細

ステップ 1 iPhone は、SDP 導入紹介段階で SDP レジストラから取得した SCEP 情報を含む設定プロファイルの指定をインストールします。

ステップ 2 iPhone はプロファイルに指定された指示に従ってキーを生成し、HTTP URL がプロファイルに指定されている SCEP サーバに登録要求とチャレンジ パスワードを送信します。

ステップ 3 SCEP サーバはチャレンジ パスワードを確認し、iPhone にデジタル証明書を発行します。

ステップ 4 ユーザはこの証明書を iPhone にインストールし、Cisco IPsec VPN を使用して会社のネットワークに接続できます。

（注） この証明書は、VPN の設定などの会社のその他の設定や Wi-Fi の設定のダウンロードに使用することもできます。

SDP 導入第二紹介段階

次のステップでは、SDP 導入第二紹介段階について説明します。

（注） SDP 展開第2導入段階は、『http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf Apple iPhone Enterprise Deployment guide』で説明する「Device Configuration」段階（またはフェーズ 4）と同じです。

手順の概要

1. iPhone は、次の場合を除き SDP 導入紹介段階を繰り返します。
2. SDP レジストラは、VPN の設定、Wi-Fi の設定、および電子メールの設定などの会社の一般的な設定を含む設定プロファイルを使用して応答します。また、VPN の確立に使用する別の証明書の SCEP の設定も含みます。

手順の詳細

ステップ 1 iPhone は、次の場合を除き SDP 導入紹介段階を繰り返します。

- iPhone の POST データにチャレンジ パスワードが含まれていない。
- SDP 導入ポスト紹介段階で、SCEP サーバから取得した証明書を使用して、iPhone が POST データに署名している。

ステップ 2 SDP レジストラは、VPN の設定、Wi-Fi の設定、および電子メールの設定などの会社の一般的な設定を含む設定プロファイルを使用して応答します。また、VPN の確立に使用する別の証明書の SCEP の設定も含みます。

2 回目の SDP 導入ポスト紹介段階

2 回目の SDP 導入ポスト紹介段階は、SDP 導入ポスト紹介段階と同じです。iPhone は、2 回目の SDP 導入紹介段階で SDP レジストラが提供する SCEP の設定に基づいて証明書要求を生成し、SCEP サーバに登録します。

SDP 導入完了段階

SDP 導入完了段階は iPhone には適用されません。これは、イントロデューサ（Safari Web ブラウザなど）が SDP ペティショナ（iPhone）で実行されるためです。

PKI への登録のための Secure Device Provisioning（SDP）の設定方法

ここでは、ご使用の PKI に対して SDP を設定する場合に従う次の手順について説明します。レジストラは、レジストラ設定作業のいずれかだけにしたがって設定できます。

SDP ペティショナのイネーブル化

ペティショナをイネーブルまたはディセーブルにし、トラストポイントを SDP 交換に関連付ける場合にこの作業を行います。

またこの作業で、証明書および特定のトラストポイントに関連付けられた RSA キーを使用するようペティショナを設定できます。

（注） ペティショナは、暗号イメージを含むシスコデバイスではデフォルトでイネーブルにされています。したがって、以前にペティショナをディセーブルにしたことがあったり、自動生成されたトラストポイントではなく、既存のトラストポイントを使用する場合は、**crypto provisioning petitioner** コマンドを発行するだけです。

（注） デフォルトでは、SDP ペティショナ デバイスでは既存の証明書が使用されます。複数の証明書および特定の証明書が 1 つ存在する場合は、どちらか選択するためにこの作業を行います。ただし、デフォルト動作をイネーブルにする場合にはこの作業は必要ありません。

始める前に

- **ip http server** コマンドを使用して HTTP サーバーをイネーブルにする必要があります（HTTP サーバは通常、多数の Cisco IOS 設定ではデフォルトでイネーブルにされています）。
- 証明書および RSA キーを使用するようペティショナを設定している場合、SDP ペティショナデバイスには既存の製造業者の証明書または第三者の証明書がなければなりません。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto provisioning petitioner**
4. 次のいずれかを実行します。
   - **trustpoint** *trustpoint-label*
5. **end**

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： `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 provisioning petitioner** 例： `Router(config)# crypto provisioning petitioner`	SDP ペティショナデバイスの動作を変更できるようにし、tti-petitioner コンフィギュレーション モードを開始します。 （注） Cisco IOS リリース 12.3(14)T では、**crypto provisioning petitioner** コマンドが **crypto wui tti petitioner** コマンドに置き換えられました。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかを実行します。
	• **trustpoint** *trustpoint-label*
	例：
	`Router(tti-petitioner)# trustpoint mytrust`
	例：
	例：
	例：
	trustpoint signing
	trustpoint-label
	例：
	`Router(tti-petitioner)# trustpoint signing mytrust`
		（任意）ペティショナとレジストラ間でSDP交換と関連付けるトラストポイントを指定します。
		（注） このコマンドが発行されないと、*trustpoint-label* 引数には自動的に「tti」のラベルが付きます。
		（任意）SDP交換中にすべての紹介データに署名する場合に使用されるトラストポイントと関連証明書を指定します。

ステップ 5	**end** 例： `Router(tti-petitioner)# end`	（任意）tti-petitioner コンフィギュレーション モードを終了します。

トラブルシューティングのヒント

SDP交換が完了したら、「tti」という新しいトラストポイントラベルができあがります。トラストポイントは、自動的に証明書サーバ（レジストラ）に登録されます。トラストポイントが実際に存在することを確認するには、**show running-config** コマンドを使用します。

次の作業

証明書と特定のトラストポイントに関連付けられた RSA キーを使用するようペティショナを設定する場合、「証明書を使用した認可のための SDP レジストラのイネーブル化」の作業で示されている方法で、レジストラを設定する必要があります。

SDP レジストラのイネーブル化と AAA リストのサーバへの追加

レジストラをイネーブルにし、証明書サーバを SDP 交換と関連付ける場合にこの作業を行います。

また、認証リストと認可リストを RADIUS サーバまたは TACACS+ サーバに追加する場合にもこの作業を行うことができます。

前提条件

レジストラを設定する前に、次の作業を実行します。

- HTTP サーバまたは HTTPS サーバをイネーブルにします。

（注） HTTPS サーバをイネーブルにする前に、標準の HTTP サーバが設定されている場合は、それをディセーブルにるす必要があります。HTTP サーバーをディセーブルにするには、**no ip http server** コマンドを使用します。HTTPS サーバーをイネーブルにするには、**ip http secure-server** コマンドの後に **ip http secure-trustpoint** コマンドを発行する必要があります。指定のトラストポイントは、レジストラとユーザーのブラウザ間の HTTPS 通信に適切なレジストラ ローカル トラストポイントです。

- **crypto pki server** コマンドを使用して、Cisco IOS 証明書サーバーを設定します。

AAA リストを設定する場合、次の作業を完了するだけでなく、レジストラに必要な前提条件を完了する必要があります。

- ユーザ情報を AAA サーバ データベースに追加します。RADIUS サーバーまたは TACACS+ AAA サーバーを設定するには、『*Cisco IOS Security Configuration Guide*』の「Configuring RADIUS」および「Configuring TACACS+」の章を参照してください。
- 新しい AAA リストを設定します。AAA リストを設定するには、『*Cisco IOS Security Configuration Guide*』の「Configuring RADIUS」、「Configuring TACACS+」、「Configuring Authentication」、および「Configuring Authorization」を参照してください。

機能制限

Cisco IOS CA デバイスの要件

SDP プロセス中、Cisco IOS CA 証明書はピア デバイスに自動的に発行されます。SDP レジストラが第三者のベンダーの CA デバイスで設定されている場合、SDP プロセスは動作しません。

template config コマンド

Cisco IOS 設定変数は 9 つあります。設定でさらに柔軟性が必要な場合、**template config** コマンドを使用して、イントロデューサに固有の設定テンプレートを参照できます。設定の柔軟性の詳細については、「カスタム設定およびファイルのテンプレート型変数の展開ルール（18 ページ）」を参照してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **pki-server** *label*
5. **authentication list** *list-name*
6. **authorization list** *list-name*
7. **template username** *name* **password** *password*
8. **template config** *url* [**post**]
9. **end**

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： `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 provisioning registrar** 例： `Router(config)# crypto provisioning registrar`	デバイスを SDP 交換のレジストラになるよう設定し、tti-registrar コンフィギュレーション モードを開始します。 （注） Cisco IOS リリース 12.3(14)T では、**crypto provisioning registrar** コマンドが **crypto wui tti registrar** コマンドに置き換えられました。
ステップ 4	**pki-server** *label* 例： `Router(tti-registrar)# pki-server mycs`	ペティショナとレジストラ間でSDP交換と関連付ける証明書サーバを指定します。
ステップ 5	**authentication list** *list-name* 例： `Router (tti-registrar)# authentication list authen-tac`	（任意）SDP 交換でイントロデューサを認証します。
ステップ 6	**authorization list** *list-name* 例：	（任意）証明書の題名およびペティショナに返信される Cisco IOS CLI スニペットに展開されるテンプ

	コマンドまたはアクション	目的
	`Router (tti-registrar)# authorization list author-rad`	レート型変数のリストに該当する認証フィールドを受信します。
ステップ 7	**template username** *name* **password** *password* 例： `Router(tti-registrar)# template username ftpuser password ftppwd`	（任意）ファイルシステムの設定テンプレートにアクセスするためのユーザ名およびパスワードを確立します。
ステップ 8	**template config** *url* [**post**] 例： `Router(tti-registrar)# template config http://myserver/cgi-bin/mycgi post`	（任意）Cisco IOS CLI 設定テンプレートのリモート URL を指定します。 *url* 引数は設定ファイルを参照し、デバイス名（$n）を指定してブートストラップ設定を識別できます。CGI サポートにより HTTP または HTTPS のいずれかを使用して CGI スクリプトを参照でき、デバイス名だけでなく、タイプ、Cisco IOS 現行バージョン情報、および現行の設定でブートストラップ設定を識別できます。 CGI サポートでは **post** キーワードを使用する必要があります。 （注） 拡張 CGI サポートを利用するには、レジストラは Cisco IOS リリース 12.4(6)T 以降を実行している必要があります。レジストラがそれ以前のバージョンの Cisco IOS を実行している場合は、追加デバイス ID 情報は無視されます。
ステップ 9	**end** 例： `Router(tti-registrar)# end`	（任意）tti-registrar コンフィギュレーション モードを終了します。

例

SDP トランザクションのトラブルシューティングに役立てるため、**debug crypto provisioning** コマンドを発行できます。このコマンドにより、ペティショナデバイスとレジストラデバイスからの出力が表示されます。

次に、**debug crypto provisioning** コマンドの出力を示します。次に、ペティショナ デバイスとレジストラ デバイスからの出力を示します。

```
Petitioner device
! The user starts the Welcome phase.
```

```
Nov  7 03:15:48.171: CRYPTO_PROVISIONING: received welcome get request.
! The router generates a Rivest, Shamir, and Adelman (RSA) keypair for future enrollment.
Nov  7 03:15:48.279: CRYPTO_PROVISIONING: keyhash 'A506BE3B83C6F4B4A6EFCEB3D584AACA'
! The TTI transaction is completed.
Nov  7 03:16:10.607: CRYPTO_PROVISIONING: received completion post request.
Registrar device
!. During the introduction phase, the browser prompts for login information.
06:39:18: CRYPTO_PROVISIONING: received introduction post request.
06:39:18: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist,
ttiuser)
! This happens if the user types in the wrong username or password.
06:39:19: CRYPTO_PROVISIONING: authentication declined by AAA, or AAA server not found
- 0x3
06:39:19: CRYPTO_PROVISIONING: aaa query fails!
! The user re-enters login information.
06:39:19: CRYPTO_PROVISIONING: received introduction post request.
06:39:19: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist,
ttiuser)
06:39:20: CRYPTO_PROVISIONING: checking AAA authorization (ipsecca_script_aaalist,
ttiuser)
! The login attempt succeeds and authorization information is retrieved from the AAA
database.
06:39:21: CRYPTO_PROVISIONING: aaa query ok!
! These attributes are inserted into the configuration template.
06:39:21: CRYPTO_PROVISIONING: building TTI av pairs from AAA attributes
06:39:21: CRYPTO_PROVISIONING: "subjectname" = "CN=user1, O=company, C=US"
06:39:21: CRYPTO_PROVISIONING: "$1" = "ntp server 10.3.0.1"
06:39:21: CRYPTO_PROVISIONING: "$2" = "hostname user1-vpn"
! The registrar stores this subject name and overrides the subject name in the subsequent
 enrollment request.
06:39:21: CRYPTO_PROVISIONING: subjectname=CN=user1, O=company, C=US
! The registrar stores this key information so that it may be used to automatically grant
 the subsequent enrollment request.
06:39:21: CRYPTO_PROVISIONING: key_hash=A506BE3B83C6F4B4A6EFCEB3D584AACA
```

証明書を使用した認可のための SDP レジストラのイネーブル化

SDP レジストラをイネーブルにし、指定されたトラストポイントまたは設定済みのトラストポイントを使用してペティショナ署名証明書を確認し、イントロデューサのユーザ名と証明書名フィールドを使用して認可検索を開始する場合は、この作業を実行します。

始める前に

証明書および特定のトラストポイントに関連付けられた RSA キーを使用するには、SDP ペティショナも設定する必要があります。この作業を完了するには、「SDP ペティショナのイネーブル化 （30ページ）」の作業の項で示すように、トラストポイント署名コマンドを使用します。

（注） RADIUS では認証と認可の区別がされていないため、証明書の認可にはデフォルトパスワードの cisco を使用する必要があります。

>

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **template file** *sourceURL destinationURL*
5. **binary file** *sourceURL destinationURL*
6. **authentication trustpoint** {**trustpoint-label| use-any }**
7. **authorization {login | certificate | login certificate}**
8. **authorization username subjectname** *subjectname*
9. **end**

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： `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 provisioning registrar** 例： `Router(config)# crypto provisioning registrar`	SDP レジストラになるようデバイスを設定し、tti-registrar 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	**template file** *sourceURL destinationURL* 例： `Router(tti-registrar)# template file http://myserver/registrar_file_r1 http://myserver/petitioner_file_p1`	（任意）レジストラの発信元テンプレート ファイル位置とペティショナの宛先テンプレート ファイル位置を指定します。 （注） このコマンドは、USB トークンを使用してデバイスをプロビジョニングする場合に便利です。 テンプレート展開は、レジストラで発信元 URL とファイル コンテンツの両方について行われます。宛先 URL はペティショナで展開されます。
ステップ 5	**binary file** *sourceURL destinationURL* 例： `Router(tti-registrar)# binary file`	（任意）レジストラのバイナリ ファイル位置とペティショナの宛先バイナリ ファイル位置を指定します。

	コマンドまたはアクション	目的
	`http://myserver/registrar_file_a1` `http://myserver/petitioner_file_b1`	（注） このコマンドは、USB トークンを使用してデバイスをプロビジョニングする場合に便利です。 発信元と宛先両方の URL はレジストラで展開されます。また、宛先 URL とファイル コンテンツはペティショナで展開されます。バイナリ ファイルは、テンプレート展開機能では処理されません。
ステップ 6	**authentication trustpoint** {**trustpoint-label| use-any }** 例： `Router(tti-registrar)# authentication trustpoint mytrust`	（任意）SDP ペティショナデバイスの現在の証明書の認証に使用するトラストポイントを指定します。 • *trustpoint-label*：特定のトラストポイントを指定します。 • **use-any**：任意の設定済みトラストポイントを指定します。 （注） トラストポイントを指定するのにこのコマンドを使用しない場合、既存のペティショナ証明書は検証されません（この機能は、自己署名ペティショナ証明書と互換性があります）。
ステップ 7	**authorization {login | certificate | login certificate}** 例： `Router(tti-registrar)# authorization login certificate`	（任意）イントロデューサまたは証明書の AAA 認可をイネーブルにします。 • イントロデューサのユーザー名を使用した認可には、**login** キーワードを使用します。 • ペティショナの証明書を使用した認可には、**certificate** キーワードを使用します。 • イントロデューサのユーザー名およびペティショナの証明書を使用した認可には、**login certificate** キーワードを使用します。
ステップ 8	**authorization username subjectname** *subjectname* 例： `Router(tti-registrar)# authorization username subjectname all`	AAA ユーザー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。 • **all** キーワードは、証明書を認可ユーザー名として使用する場合に、所有者名全体を指定します。
ステップ 9	**end** 例：	（任意）tti-registrar 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
	`Router(tti-registrar)# end`	

Apple iPhone を導入するための SDP レジストラの設定

会社のネットワークに Apple iPhone を導入するために HTTPS を実行するように SDP レジストラを設定する場合は、この作業を実行します。

始める前に

HTTPS を実行するために SDP レジストラがイネーブルであることを確認します。詳細については、「SDP レジストラのイネーブル化と AAA リストのサーバーへの追加」を参照してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip http secure-server**
4. **crypto provisioning registrar**
5. **url-profile start** *profile-name*
6. **url-profile intro** *profile-name*
7. **match url** *url*
8. **match authentication trustpoint** *trustpoint-name*
9. **match certificate** *certificate-map*
10. **mime-type** *mime-type*
11. **template location** *location*
12. **template variable p** *value*

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： `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	**ip http secure-server** 例：	HTTPS Web 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
	`Router(config)# ip http secure-server`	
ステップ 4	**crypto provisioning registrar** 例： `Router(config)# crypto provisioning registrar`	デバイスを SDP 交換のレジストラになるよう設定し、tti-registrar コンフィギュレーション モードを開始します。 （注） Cisco IOS リリース 12.3(14)T では、**crypto provisioning registrar** コマンドが **crypto wui tti registrar** コマンドに置き換えられました。
ステップ 5	**url-profile start** *profile-name* 例： `Router(tti-registrar)# url-profile start START`	**start** キーワードを指定して、URL プロファイルが SDP 導入開始段階と関連付けられることを示します。*profile-name* 引数には、一意の URL プロファイルの名前を指定します。 （注） SDP 導入紹介段階と SDP 導入開始段階では、いずれも異なるプロファイルを使用したり、同じ URL プロファイルを使用したりすることができます。
ステップ 6	**url-profile intro** *profile-name* 例： `Router(tti-registrar)# url-profile intro INTRO`	**intro** キーワードを指定して、URL プロファイルが SDP 導入紹介段階と関連付けられることを示します。*profile-name* 引数には、一意の URL プロファイルの名前を指定します。 （注） SDP 導入紹介段階と SDP 導入開始段階では、いずれも異なるプロファイルを使用したり、同じ URL プロファイルを使用したりすることができます。
ステップ 7	**match url** *url* 例： `Router(tti-registrar)# match url /sdp/intro`	URL プロファイルに関連付ける URL を指定します。
ステップ 8	**match authentication trustpoint** *trustpoint-name* 例： `Router(tti-registrar)# match authentication trustpoint apple-tp`	（任意）ピアの証明書の認証に使用するトラストポイントの名前を指定します。トラストポイントの名前が指定されていない場合、ピアの証明書の認証には tti-registrar コンフィギュレーション モードで **authentication trustpoint command** を使用して設定されたトラストポイントが使用されます。詳細については、「証明書を使用した認可のための SDP レジストラのイネーブル化」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 9	**match certificate** *certificate-map* 例： `Router(tti-registrar)# match certificate cat 10`	（任意）ピアの証明書の許可に使用される証明書マップの名前を指定します。
ステップ 10	**mime-type** *mime-type* 例： `Router(tti-registrar)# mime-type application/x-apple-aspen-config`	SDP レジストラが URL プロファイルを通して受信した要求への応答に使用する多目的インターネットメール拡張（MIME）タイプを指定します。
ステップ 11	**template location** *location* 例： `Router(tti-registrar)# template location flash:intro.mobileconfig`	SDP レジストラが URL プロファイルを通して受信した要求に応答するときに使用するテンプレートの場所を指定します。
ステップ 12	**template variable p** *value* 例： `Router(tti-registrar)# template variable p iphone-vpn`	（任意）SDP レジストラによって発行されるトラストポイント証明書の所有者名の組織ユニット（OU）フィールドに入力する値を指定します。以下の「Apple CA サーバーのトラストポイント証明書の設定例」に示されている証明書のこのフィールドを参照してください。

Apple CA サーバーのトラストポイント証明書の設定

SDP レジストラは、Apple CA サーバーの証明書を信頼するために、iPhone のトラストポイント証明書から生成された署名を確認する必要があります。iPhone はトラストポイント証明書を使用してメッセージに署名します。このトラストポイント証明書は、SDP 導入紹介段階で Apple 社の CA サーバーによって発行されます。

次の例では、Apple 社の CA 証明書をカットアンドペーストして手動で登録する方式を使用して、証明書登録を設定する方法を示します。

（注） トラストポイント証明書の設定の詳細については、『Configuring Certificate Enrollment for a PKI』フィーチャモジュールの「How to Configure Certificate Enrollment for a PKI」の項も参照してください。

手順の概要

1. グローバル コンフィギュレーション モードで **crypto pki trustpoint** コマンドを入力してトラストポイントおよび設定された名前を宣言し、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。

2. カットアンドペーストして手動で証明書を登録するように指定するには、 **enrollment terminal** コマンドを入力します。
3. **crypto pki authenticate** コマンドを使用して、指定された TFTP サーバーから CA 証明書を取得して認証します。
4. Base 64 符号化の信頼できる Apple 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。
5. **exit** コマンドを使用して CA トラストポイント コンフィギュレーション モードを終了し、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。
6. グローバル コンフィギュレーション モードで **crypto provisioning registrar** コマンドを入力し、SDP 交換用のレジストラになるルータを指定して、tti-registrar コンフィギュレーション モードを開始します。
7. tti-registrar コンフィギュレーション モードで **url-profile command with the intro** キーワードを入力し、SDP 導入紹介段階に関連付けられる一意の URL プロファイルの名前を指定します。
8. tti-registrar コンフィギュレーション モードで **match authentication trustpoint** コマンドを入力し、ピアの証明書の認証に使用するトラストポイントの名前を指定します。

手順の詳細

ステップ 1 グローバル コンフィギュレーション モードで **crypto pki trustpoint** コマンドを入力してトラストポイントおよび設定された名前を宣言し、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。

例：

```
Router(config)# crypto pki trustpoint apple-tp
```

ステップ 2 カットアンドペーストして手動で証明書を登録するように指定するには、 **enrollment terminal** コマンドを入力します。

例：

```
Router(ca-trustpoint)# enrollment terminal
```

ステップ 3 **crypto pki authenticate** コマンドを使用して、指定された TFTP サーバーから CA 証明書を取得して認証します。

例：

```
Router(ca-trustpoint)# crypto pki authenticate apple-tp
```

ステップ 4 Base 64 符号化の信頼できる Apple 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。

例：

```
I Bag Attributes
    localKeyID: 7C 29 15 15 12 C9 CF F6 15 2B 5B 25 70 3D A7 9A 98 14 36 06
subject=/C=US/O=Apple Inc./OU=Apple iPhone/CN=Apple iPhone Device CA
issuer=/C=US/O=Apple Inc./OU=Apple Certification Authority/CN=Apple iPhone Certification Authority
```

```
-----BEGIN CERTIFICATE-----
MIIDaTCCAlGgAwIBAgIBATANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzET
MBEGA1UEChMKQXBwbGUgSW5jLjEmMCQGA1UECxMdQXBwbGUgQ2VydGlmaWNhdGlv
biBBdXRob3JpdHkxLTArBgNVBAMTJEFwcGxlIGlQaG9uZSBDZXJ0aWZpY2F0aW9u
IEF1dGhvcml0eTAeFw0wNzA0MTYyMjU0NDZaFw0xNDA0MTYyMjU0NDZaMFoxCzAJ
BgNVBAYTAlVTMRMwEQYDVQQKEwpBcHBsZSBJbmMuMRUwEwYDVQQLEwxBcHBsZSBp
UGhvbmUxHzAdBgNVBAMTFkFwcGxlIGlQaG9uZSBEZXZpY2UgQ0EwgZ8wDQYJKoZI
hvcNAQEBBQADgY0AMIGJAoGBAPGUSsnquloYYK3Lok1NTlQZaRdZB2bLl+hmmkdf
Rq5nerVKc1SxywT2vTa4DFU4ioSDMVJl+TPhl3ecK0wmsCU/6TKqewh0lOzBSzgd
Z04IUpRai1mjXNeT9KD+VYW7TEaXXm6yd0UvZ1y8Cxi/WblshvcqdXbSGXH0KWO5
JQuvAgMBAAGjgZ4wgZswDgYDVR0PAQH/BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8w
HQYDVR0OBBYEFLL+ISNEhpVqedWBJo5zENinTI50MB8GA1UdIwQYMBaAFOc0Ki4i
3jlga7SUzneDYS8xoHw1MDgGA1UdHwQxMC8wLaAroCmGJ2h0dHA6Ly93d3cuYXBw
bGUuY29tL2FwcGxlY2EvaXBob25lLmNybDANBgkqhkiG9w0BAQUFAAOCAQEAd13P
Z3pMViukVHe9WUg8Hum+0I/0kHKvjhwVd/IMwGlXyU7DhUYWdja2X/zqj7W24Aq5
7dEKm3fqqxK5XCFVGY5HI0cRsdENyTP7lxSiiTRYj2mlPedheCn+k6T5y0U4Xr40
FXwWb2nWqCF1AgIudhgvVbxlvqcxUm8Zz7yDeJ0JFovXQhyO5fLUHRLCQFssAbf8
B4i8rYYsBUhYTspVJcxVpIIltkYpdIRSIARA49HNvKK4hzjzMS/OhKQpVKw+OCEZ
xptCVeN2pjbdt9uzi175oVo/u6B2ArKAW17u6XEHIdDMOe7cb33peVI6TD15W4MI
pyQPbp8orlXe+tA8JA==
-----END CERTIFICATE-----
```

ステップ 5 **exit** コマンドを使用して CA トラストポイント コンフィギュレーション モードを終了し、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。

例：

```
Router(ca-trustpoint)# exit
```

ステップ 6 グローバル コンフィギュレーション モードで **crypto provisioning registrar** コマンドを入力し、SDP 交換用のレジストラになるルータを指定して、tti-registrar コンフィギュレーション モードを開始します。

例：

```
Router(config)# crypto provisioning registrar
```

ステップ 7 tti-registrar コンフィギュレーション モードで **url-profile command with the intro** キーワードを入力し、SDP 導入紹介段階に関連付けられる一意の URL プロファイルの名前を指定します。

例：

```
Router(tti-registrar)# url-profile intro INTRO
```

ステップ 8 tti-registrar コンフィギュレーション モードで **match authentication trustpoint** コマンドを入力し、ピアの証明書の認証に使用するトラストポイントの名前を指定します。

例：

```
Router(tti-registrar)# match authentication trustpoint apple-tp
```

これで、SDP レジストラは iPhone の署名を確認する際に、「apple-tp」という名前の Apple CA トラストポイント証明書を使用できます。

管理イントロデューサの設定

管理者の認証リストと認可リストを使用して、管理イントロデューサを設定するには、次の作業を行います。

始める前に

管理イントロデューサは、クライアントデバイスの特権およびサーバの管理者特権をイネーブルにしておく必要があります。

（注） RADIUS を使用する場合、管理イントロデューサにより紹介される必要があるユーザまたはデバイスのパスワードとして常に cisco を使用する必要があります。TACACS+ にはこの制限はありません。ユーザまたはデバイスはどのパスワードを使用しても、管理イントロデューサにより紹介されます。

>

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **administrator authentication list** *list-name*
5. **administrator authorization list** *list-name*
6. **end**

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： `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 provisioning registrar** 例： `Router(config)# crypto provisioning registrar`	SDP レジストラになるようデバイスを設定し、tti-registrar 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	**administrator authentication list** *list-name* 例： `Router(tti-registrar)# administrator authentication list authen-tac`	紹介中に管理者を認証場合に使用する AAA リストを設定します。
ステップ 5	**administrator authorization list** *list-name* 例： `Router(tti-registrar)# administrator authorization list author-tac`	紹介中に管理者の認可情報を取得する場合に使用する AAA リストを設定します。取得できる情報として、証明書の題名またはペティショナに返信される Cisco IOS CLI スニペットに展開されるテンプレート型変数のリストがあります。
ステップ 6	**end** 例： `Router(tti-registrar)# end`	（任意）tti-registrar コンフィギュレーション モードを終了します。

例

show running-config コマンドの次の例では、認証リストと認可リストを使用した管理イントロデューサが作成されたことを確認できます。

```
Router# show running-config
Building configuration...
Current configuration : 2700 bytes
!
! Last configuration change at 01:22:26 GMT Fri Feb 4 2005
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname router
!
boot-start-marker
boot-end-marker
!
memory-size iomem 5
enable secret 5 $1$tpBS$PXnBDTIDXfX5pWa//1JX20
enable password lab
!
aaa new-model
!
!
!
aaa session-id common
!
resource manager
!
clock timezone GMT 0
```

```
ip subnet-zero
no ip routing
!
!
no ip dhcp use vrf connected
!
!
no ip cef
no ip domain lookup
ip domain name company.com
ip host router 10.3.0.6
ip host router.company.com 10.3.0.6
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
crypto pki server mycs
!
crypto pki trustpoint mycs
 revocation-check crl
 rsakeypair mycs
!
crypto pki trustpoint tti
 revocation-check crl
 rsakeypair tti
!
crypto pki trustpoint mic
 enrollment url http://router:80
 revocation-check crl
!
crypto pki trustpoint cat
 revocation-check crl
!
!
!
crypto pki certificate map cat 10
!
crypto pki certificate chain mycs
 certificate ca 01
crypto pki certificate chain tti
crypto pki certificate chain mic
 certificate 02
 certificate ca 01
crypto pki certificate chain cat
!
crypto provisioning registrar <---------- !SDP registrar device parameters!
 administrator authentication list authen-tac
 administrator authorization list author-tac
!
no crypto engine onboard 0
username qa privilege 15 password 0 lab
```

カスタム テンプレートの設定

カスタム テンプレートを作成および設定するには、次の作業を行います。

手順の概要

1. **enable**
2. **configure terminal**

3. **crypto provisioning registrar**
4. **template http start** *URL*
5. **template http welcome** *URL*
6. **template http introduction** *URL*
7. **template http admin-introduction** *URL*
8. **template http completion** *URL*
9. **template http error** *URL*
10. **end**

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： `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 provisioning registrar** 例： `Router(config)# crypto provisioning registrar`	SDP レジストラになるようデバイスを設定し、tti-registrar 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	**template http start** *URL* 例： `Router(tti-registrar)# template http start tftp://registrar.company.com/start.html`	カスタム開始ページ テンプレートを使用するよう TTI レジストラに指示します。 （注） このコマンドは、開始ページ機能を使用する場合に必要です。このコマンドが発行されていない場合、ようこそページがイントロデューサとペティショナの最初の通信になります。
ステップ 5	**template http welcome** *URL* 例： `Router(tti-registrar)# template http welcome tftp://registrar.company.com/welcome.html`	（任意）デフォルト テンプレートではなく、カスタムようこそテンプレートを使用します。
ステップ 6	**template http introduction** *URL* 例： `Router(tti-registrar)#`	（任意）デフォルト テンプレートではなく、カスタム紹介テンプレートを使用します。

	コマンドまたはアクション	目的
	`template http introduction tftp://registrar.company.com/intro.html`	
ステップ 7	**template http admin-introduction** *URL* 例： `Router(tti-registrar)# template http admin-introduction tftp://registrar.company.com/admin-intro.html`	（任意）デフォルト テンプレートではなく、カスタム管理紹介テンプレートを使用します。
ステップ 8	**template http completion** *URL* 例： `Router(tti-registrar)# template http completion tftp://registrar.company.com/completion.html`	（任意）デフォルト テンプレートではなく、カスタム完了テンプレートを使用します。
ステップ 9	**template http error** *URL* 例： `Router(tti-registrar)# template http error tftp://registrar.company.com/error.html`	（任意）デフォルト テンプレートではなく、カスタム エラー テンプレートを使用します。
ステップ 10	**end** 例： `Router(tti-registrar)# end`	（任意）tti-registrar コンフィギュレーション モードを終了します。

例

次に、開始、紹介、および完了の各テンプレートを使用した例を示します。

template http start tftp://registrar.company.com/start.html

template http introduction tftp://registrar.company.com/intro.html

template http completion tftp://registrar.company.com/completion.html

PKI への登録のための Secure Device Provisioning（SDP）の設定例

SDP レジストラの確認の例

show running-config コマンドの次のサンプル出力では、証明書サーバー「cs1」が設定され、レジストラとペティショナ間の SDP 交換と関連付けられていることが確認できます。

```
Router# show running-config
Building configuration...
Current configuration : 5902 bytes
!
! Last configuration change at 09:34:44 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36a
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$b3jz$CKquLGjFIE3AdXA2/R19./
enable password lab
!
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name company.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.company.com 10.23.2.131
!
!
crypto pki server cs1
 issuer-name CN=company,L=city,C=US
 hash sha1
 lifetime crl 336
 lifetime certificate 730
!
crypto pki trustpoint pki-36a
 enrollment url http://pki-36a:80
 ip-address FastEthernet0/0
 revocation-check none
!
crypto pki trustpoint cs1
 revocation-check crl
 rsakeypair cs1 2048
```

```
!
!
crypto pki certificate chain pki-36a
 certificate 03
  308201D0 30820139 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333334 345A170D 30363031 33303039 33333434 5A303A31 38301606 092A8648
  86F70D01 09081309 31302E32 332E322E 32301E06 092A8648 86F70D01 09021611
  706B692D 3336612E 63697363 6F2E636F 6D305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100AFFA 8F429618 112FAB9D 01F3352E 59DD3D2D AE67E31D
  370AC4DA 619735DF 9CF4EA13 64E4B563 C239C5F0 1578B773 07BED641 A18CA629
  191884B5 61B66ECF 4D110203 010001A3 30302E30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 141DA8B1 71652961 3F7D69F0 02903AC3 2BADB137
  C6300D06 092A8648 86F70D01 01040500 03818100 67BAE186 327CED31 D642CB39
  AD585731 95868683 B950DF14 3BCB155A 2B63CFAD B34B579C 79128AD9 296922E9
  4DEDFCAF A7B5A412 AB1FC081 09951CE3 08BFFDD9 9FB1B9DA E9AA42C8 D1049268
  C524E58F 11C6BA7F C750320C 03DFB6D4 CBB3E739 C8C76359 CE939A97 B51B3F7F
  3FF;A9D82 9CFDB6CF E2503A14 36D0A236 A1CCFEAE
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
crypto pki certificate chain cs1
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A02;
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
```

```
crypto provisioning registrar
 pki-server cs1
!
!
!
crypto isakmp policy 1
 hash sha
!
!
crypto ipsec transform-set test_transformset esp-aes
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.1.10
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Loopback0
 ip address 10.23.2.131 255.255.255.255
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet0/0
 ip address 10.23.2.2 255.255.255.192
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map test_cryptomap
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 10.23.2.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.2.62
!
!
access-list 170 permit ip host 10.23.2.2 host 10.23.1.10
dialer-list 1 protocol ip permit
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

SDP ペティショナの確認の例

SDP 交換が完了したら、ペティショナは自動的にレジストラを登録し、証明書を取得します。**show running-config** コマンドによる次のサンプル出力では、トラストポイントが実際に存在することを確認する設定が自動的に生成されているところを示しています。

```
Router# show running-config
Building configuration...
Current configuration : 4650 bytes
!
! Last configuration change at 09:34:53 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36b
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$JYgw$060JKXgl6dERLZpU9J3gb.
enable password lab
!
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name company.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.company.com 10.23.2.131
!
!
crypto pki trustpoint tti
 enrollment url http://pki-36a.company.com:80
 revocation-check crl
 rsakeypair tti 1024
 auto-enroll 70
!
!
crypto pki certificate chain tti
 certificate 02
  308201FC 30820165 A00302012;02020102 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333333 385A170D 30363031 33303039 33333338 5A302231 20301E06 092A8648
  86F70D01 09021611 706B692D 3336622E 63697363 6F2E636F 6D30819F 300D0609
  2A864886 F70D0101 01050003 818D0030 81890281 8100E383 35584B6C 24751E2C
  F4088F06 C00BFECE 84CFF8EB 50D52044 03D14A2B 91E5A260 7D07ED24 DB599D27
  432065D9 0E459248 D7CDC15D 654E2AF6 BA27D79C 23850306 3E96C508 F311D333
  76FDDC9C A810F75C FCD10F1B 9A142F0C 338B6DB3 346D3F24 97A4B15D 0A9504E7
  1F6CB769 85E9F52B FE907AAF 63D54D66 1A715A20 D7DB0203 010001A3 30302E30
  0B060355 1D0F0404 030205A0 301F0603 551D2304 18301680 141DA8B1 71652961
  3F7D69F0 02903AC3 2BADB137 C6300D06 092A8648 86F70D01 01040500 03818100
```

```
  C5E2DA0E 4312BCF8 0396014F E18B3EE9 6C970BB7 B8FAFC61 EF849568 D546F73F
  67D2A73C 156202DC 7404A394 D6124DAF 6BACB8CF 96C3141D 109C5B0E 46F4F827
  022474ED 8B59D654 F04E31A2 C9AA1152 75A0C455 FD7EEEF5 A505A648 863EE9E6
  C361D9BD E12BBB36 16B729DF 823AD5CC 404CCE48 A4379CDC 67FF6362 0601B950
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
no crypto engine accelerator
!
!
crypto isakmp policy 1
 hash sha
!
!
crypto ipsec transform-set test_transformset esp-aes
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.2.2
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Ethernet0/0
 ip address 10.23.1.10 255.255.255.192
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 half-duplex
 crypto map test_cryptomap
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
```

```
 half-duplex
!
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip default-gateway 10.23.1.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.1.62
!
!
access-list 170 permit ip host 10.23.1.10 host 10.23.2.2
dialer-list 1 protocol ip permit
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

AAA リストの RADIUS または TACACS+ サーバーへの追加の例

ここでは、次の設定例を示します。

TACACS+ AAA サーバーデータベースの例

次に、ユーザ情報が TACACS+ AAA データベースに追加されている例を示します。ユーザー名は「user1」です。パスワードは「cisco」です。「user1」には、iosconfig1 と iosconfig2 の 2 つの Cisco IOS 設定テンプレート変数が設定されています。変数は設定テンプレート ファイルで $1 および $2 を置き換えます。題名「CN=user1, O=company, C=US」も設定されます。この題名は、ペティショナ デバイスから受信される以降の登録要求（PKCS10）で題名フィールドを置き換えます。

```
user = user1
    password = clear "pswd"
    service=tti
        ! The certificate server inserts the following subject name to the certificate.
        set subjectname="CN=user1, O=company, C=US"
        ! Up to nine template variables may be added.
        set iosconfig1="ntp server 10.3.0.1"
        set iosconfig2="hostname user1-vpn"
```

RADIUS AAA サーバーデータベースの例

次に、ユーザ情報が RADIUS AAA サーバ データベースに追加された例を示します。ユーザー名は「user1」です。パスワードは「cisco」です。「user1」には、iosconfig1 と iosconfig2 の 2 つの Cisco IOS 設定テンプレート変数が設定されています。変数は設定テンプレート ファイルで $1 および $2 を置き換えます。題名「CN=user1, O=company, C=US」も設定されます。この題名は、ペティショナ デバイスから受信される以降の登録要求（PKCS10）で題名フィールドを置き換えます。

```
user = user1
    password = clear "pswd"
    radius=company
    reply_attributes=9,1="tti:subjectname=CN=user1, O=company, C=US"
    ! Up to nine template variables may be added.
    9,1="tti:iosconfig1=ntp server 10.3.0.5"
    9,1="tti:iosconfig2=hostname user1-vpn"
```

TACACS+ および RADIUS AAA サーバー上の AAA リストの例

次の設定例は、TACACS+ サーバで AAA 認証が、RADIUS サーバで AAA 認可が設定されていることを示しています。

（注） 通常、認証と認可は同じサーバをポイントします。

```
Router(config)# tacacs-server host 10.0.0.48 key cisco
Router(config)# aaa authentication login authen-tac group tacacs+
Router(config)# radius-server host 10.0.1.49 key cisco
Router(config)# aaa authorization network author-rad group radius
```

UsingaConfigurationTemplateFile の例

イントロデューサ名に基づいて、異なる設定テンプレート ファイルを使用できます。たとえば、異なるユーザに対する複数のテンプレートがある場合、各ファイルのファイル名にユーザ名を含め、レジストラで次のように設定します。

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/config-$n.txt
```

この例では、設定ファイルのデフォルト テンプレート （24 ページ）に示されているデフォルト設定ファイルが使用されます。**template config** コマンドは CGI スクリプトを参照しないためです。

CGI スクリプトの例

次に、「mysdpcgi」という CGI スクリプトが実行される例を示します。

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/cgi-bin/mysdpcgi post
```

次に、「mysdpcgi」という CGI スクリプトが上記の例の **template config** コマンドで実行される例を示します。

```
#!/usr/bin/perl -w
# for debugging use the -debug form
# use CGI (-debug);
use CGI;
# base64 decoding is being used.
use MIME::Base64;
# The following has been commented out, but left for your information.
#
# Reading everything that has been received from stdin and writing it to the debug log
to #see what has been sent from the registrar.
#
# Remember to reset the STDIN pointer so that the normal CGI processing can get the
input.
#
# print STDERR "mysdpcgi.cgi dump of stdin:\n";
# if($ENV{'REQUEST_METHOD'} eq "GET"){
#    $input_data = $ENV{'QUERY_STRING'};
# }
# else {
#    $data_length = $ENV{'CONTENT_LENGTH'};
#    $bytes_read = read(STDIN, $input_data, $data_length);
# }
# print STDERR $input_data,"\n";
# exit;

$query = new CGI;
my %av_table;
# A basic configuration file is being sent back, therefore it is being indicated as plain
 # text in the command below.
print $query->header ("text/plain");
print "\n";
# For testing, parameters can be passed in so that the test applications can
# see what has been received.
#
# print STDERR "The following are the raw AV pairs mysdpcgi.cgi received:\n";
# for each $key ($query->param) {
#    print STDERR "! $key is: \n";
#    $value = $query->param($key);
#    print STDERR "! ",$value;
#    print STDERR "! \n";
#}
# The post process AV pairs are identical to those in Cisco IOS and may be used to produce
 # AV pair specific configurations as needed.
%av_table = &postprocessavpairs($query->param);
```

```
# Decoded values may be written out.
# WARNING: Some error_logs cannot handle the amount of data and will freeze.
# print STDERR "The following are the decoded AV pairs mysdpcgi.cgi received:\n";
# now write the values out
# while ( ($a, $v) = each(%av_table) ) {
#    print STDERR "$a = $v\n";
# }
# Identifying the AV pairs and specifying them in the config.
while ( ($a, $v) = each(%av_table) ) {
    if ($a eq "TTIIosRunningConfig") {
        $search = "hostname ";
        $begin = index($v, $search) + length($search);
        $end = index($v, "\n", $begin);
        $hostname = substr($v, $begin, $end - $begin);
    }
    if ($a eq "TTIIosVersion") {
        $search = "Version ";
        $begin = index($v, $search) + length($search);
        $end = index($v, "(", $begin);
        $version = substr($v, $begin, $end - $begin);
    }
}
print <<END_CONFIG;
!
! Config auto-generated by sdp.cgi
! This is for SDP testing only and is not a real config
!
!
\$t
!
\$c
!
cry pki trust Version-$version-$hostname
! NOTE: The last line of the config must be 'end' with a blank line after the end
# statement.
END_CONFIG
;
# Emulate IOS tti_postprocessavpairs functionality
sub postprocessavpairs {
    @attributes = @_;
    # Combine any AV pairs that were split apart
    $n = 0; #element index counter
    while ($attributes[$n]) {
 # see if we are at the start of a set
 if ($attributes[$n] =~ m/_0/) {
     # determine base attribute name
     $a = (split /_0/, $attributes[$n])[0];
     # set initial (partial) value
     $v = $query->param($attributes[$n]);

     # loop and pull the rest of the matching
     # attributes's values into v (would be
     # faster if we stop at first non-match)
     $c = $n+1;
     while ($attributes[$c]) {
  if ($attributes[$c] =~ m/$a/) {
      $v = $v.$query->param($attributes[$c]);
  }
  $c++;
     }

     # store in the av hash table
     $av_table{$a} = $v;
 } else {
```

```
    # store in hash table if not part of a set
    if ($attributes[$n] !~ m/_\d/) {
 $av_table{$attributes[$n]} = $query->param($attributes[$n]);
    }
}
$n++;
   }
   # de-base64 decode all AV pairs except userdevicename
   while ( ($a, $v) = each(%av_table) ) {
       if ($a ne "userdevicename") {
           $av_table{$a} = decode_base64($av_table{$a});
       }
   }
   return %av_table;
}
```

（注） CGI スクリプトは、Cisco IOS Release 12.4(6)T 以降では、**template config** コマンド に **post** キーワードを指定せずに実行することはできません。

証明書を使用した認証のペティショナとレジストラの設定の例

次に、mytrust というトラストポイントで発行された証明書を使用する場合のペティショナの設定方法の例を示します。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning petitioner

Router(tti-petitioner)# trustpoint signing mytrust

Router(tti-petitioner)# end
```

次に、ペティショナ署名証明書を確認し、認可検索を行う場合のレジストラの設定方法の例を示します。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning registrar

Router(tti-registrar)# authentication trustpoint mytrust

Router(tti-registrar)# authorization login certificate

Router(tti-registrar)# authorization username subjectname all

Router(tti-registrar)# end
```

認証リストおよび認可リストを使用した管理イントロデューサの設定例

次に、認証リスト「authen-tac」および認可リスト「author-tac」を使用した管理イントロデューサの設定方法の例を示します。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning registrar
Router(tti-registrar)# administrator
 authentication list authen-tac
Router(tti-registrar)# administrator
 authorization list author-tac
Router(tti-registrar)# end
```

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
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
証明書登録	「 PKI の証明書登録の設定 」 モジュール
証明書サーバ設定	「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
PKI AAA 統合の概念と設定作業	「PKI での証明書の失効および許可の設定」モジュール
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*』
USB トークンの設定	『*Cisco IOS Security Configuration Guide: Secure Connectivity*』の「Storing PKI Credentials」の章 SDP および USB トークンを使用した PKI クレデンシャルの導入に関するその他の 12.4T 機能については、機能情報表を参照してください。
iPhone、iPod touch、および iPad と会社のシステムとの統合	『*Apple iPhone Enterprise Deployment Guide*』
推奨される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 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

PKI への登録のための Secure Device Provisioning（SDP）の設定に関する機能情報

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* での *SDP* の機能情報

機能名	リリース	機能情報
Secure Device Provisioning（SDP）接続テンプレート	12.4(20)T	この機能により、サービス プロバイダーを通してインターネット接続が行われるようにデバイスを設定できます。

機能名	リリース	機能情報
USB トークンと Secure Device Provisioning（SDP）の連携機能	12.4(15)T	この機能により、SDP を介して特定のネットワーク デバイスからリモート デバイスにクレデンシャルを転送するメカニズムとして USB トークンを使用することで、リモート デバイスをプロビジョニングできるようになります。 次のコマンドが導入されました。**binary file**、**crypto key move rsa**、**template file**。
SDP 拡張テンプレートの CGI サポート	12.4(6)T	この機能によりユーザーは、デバイス名だけでなく、その Cisco IOS の現行バージョンおよび現行の設定に基づいてブートストラップ設定を SDP ペティショナに送信するよう SDP レジストラを設定できます。 次のコマンドが、この機能によって変更されました。**template config.**
Secure Device Provisioning（SDP）開始ページ	12.4(4)T	この機能によりユーザーは、開始ページからレジストラの紹介 URL に連絡することで TTI トランザクションを開始するよう、ブラウザを設定できます。したがって、ユーザはペティショナのようこそページから TTI トランザクションを開始する必要はなくなります。 この機能により、次のコマンドが導入されました。**template http admin-introduction**、**template http completion**、**template http error**、**template http introduction**、**template http start**、**template http welcome.**
Administrative Secure Device Provisioning Introducer	12.3(14)T	この機能により、デバイスを PKI ネットワークに紹介し、AAA データベースのレコード ロケータのデバイス名としてユーザ名を提供する場合に、管理イントロデューサの役割を果たすことができます。 この機能により、次のコマンドが導入されました。**administrator authentication list**、**administrator authorization list.**
Easy Secure Device Deployment	12.3(8)T	この機能は、SDP をサポートできるようにします。SDP は、ネットワーク管理者が大規模ネットワークで新しいデバイスを展開できるようにする Web ベースの登録インターフェイスを実現します。 次のコマンドが導入または変更されました。**crypto wui tti petitioner**、**crypto wui tti registrar**、**pki-server**、**template config**、 **template username**、**trustpoint (tti-petitioner).**

機能名	リリース	機能情報
Easy Secure Device Deployment AAA Integration	12.3(8)T	この機能により外部 AAA データベースが統合され、ローカルなシスコ証明書サーバのイネーブル パスワードを使用しなくても、SDP イントロデューサが AAA データベースに対して認証できるようにします。 次のコマンドが導入または変更されました。**authentication list (tti-registrar)**、**authorization list (tti-registrar)**、**debug crypto wui template config**、 **template username**。
Secure Device Provisioning（SDP）証明書を使用した認可	12.3(14)T	この機能により、その他の認証局（CA）サーバで発行された証明書が SDP 導入に使用できるようになります。 この機能により、次のコマンドが導入されました。**administrator authentication list**、**administrator authorization list**
iPhone の SDP	15.1(2)T	Cisco IOS 15.1(2)T および Apple iPhone OS 3.0 リリースが導入されたため、Cisco IOS ネットワーク デバイスで Apple iPhone がサポートされるようになりました。Cisco IOS ルータは SDP レジストラを使用して iPhone を導入し、IPSec VPN、SCEP サーバ、および PKI 証明書の導入テクノロジーを使用してネットワーク アプリケーションに安全にアクセスできるようにします。 この機能により、次のコマンドが導入されました。**match authentication trustpoint, match certificate** 、 **match url, mime-type**、**template location, template variable p, url-profile.**

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。