PKI クレデンシャルの保存

Rivest、Shamir、Adelman（RSA）キーと証明書などの公開キー インフラストラクチャ（PKI）は、NVRAM やフラッシュ メモリなどのルータまたは USB eTtoken 64 KB スマート カード上の特定の場所に保存できます。USB トークンを使用すると、セキュアな設定配布や、トークン上のキー生成、署名、認証などの RSA 処理、配置のためのバーチャル プライベート ネットワーク（VPN）クレデンシャルを USB トークンのストレージが提供されます。

PKI クレデンシャルを保存するための前提条件

ローカル証明書の保管場所を指定するための前提条件

ローカル証明書の保管場所を指定するためには、ご使用のシステムが次の要件を満たしている必要があります。

- Cisco IOS Release 12.4(2)T PKI 対応イメージまたはそれ以降のイメージ
- PKI クレデンシャルを個別のファイルとして保存できるプラットフォームであること。
- 設定内に証明書が少なくとも 1 つ含まれていること。
- アクセス可能なローカル ファイル システムがあること。

PKI クレデンシャルの保管場所として USB トークンを指定するための前提条件

USB トークンを使用するためには、ご使用のシステムが次の要件を満たしている必要があります。

- Cisco 871 ルータ、Cisco 1800 シリーズ ルータ、Cisco 2800 シリーズ ルータ、Cisco 3800 シリーズ ルータ、または Cisco 7200VXR NPE-G2 プラットフォームを使用していること。
- サポートされているいずれかのプラットフォーム上で、少なくとも Cisco IOS Release 12.3(14)T イメージが稼働していること。
- シスコのサポート対象 USB トークン（Safenet/Aladdin eToken PRO 32 KB または 64 KB）
- k9 イメージを使用していること。

PKI クレデンシャルの保存に関する制約事項

ローカル証明書の保管場所を指定する場合の制約事項

証明書をローカルな保管場所に保存する場合には、次のような制約事項があります。

- 使用できるのはローカル ファイル システムだけです。リモート ファイル システムを選択すると、エラー メッセージが表示され、コマンドは無効になります。
- ローカルファイルシステムでサポートされていれば、サブディレクトリを指定できます。NVRAM では、サブディレクトリはサポートされていません。

保管場所として USB トークンを指定する場合の制約事項

USB トークンを使用して PKI データを保存する場合には、次のような制約事項があります。

- USB トークンがサポートされるためには、ファイルをセキュアに保存できる 3DES（k9）Cisco IOS ソフトウェア イメージが必要です。
- イメージは USB トークンからは起動できません（ただし、設定は USB トークンからでも起動できます）。
- USBハブは現在、サポートされていません。そのため、サポートされるデバイスの数は、多くても使用できる USB ポートの数までです。

PKI クレデンシャルの保存について

ローカルな保管場所への証明書の保存

デフォルトでは、証明書はNVRAMに格納されます。ただし、ルータによっては、証明書を正常に保存するために必要なサイズの NVRAM が搭載されていないことがあります。

シスコのプラットフォームはすべて、NVRAMおよびフラッシュ ローカル ストレージをサポートしています。ご使用のプラットフォームによっては、ブートフラッシュ、スロット、ディス

ク、USB フラッシュ、USB トークンなど、サポートされているその他のローカル ストレージを使用できます。

実行時には、証明書を保存するアクティブなローカル ストレージ デバイスを指定できます。

PKI クレデンシャルと USB トークン

ご使用のルータ上でセキュアな USB トークンを使用するためには、次に説明する事柄について十分な知識が必要です。

USB トークンの動作のしくみ

スマートカードはプラスティック製の小型カードで、データの保存や処理を行うためのマイクロプロセッサやメモリが搭載されています。USB インターフェイスを備えたスマート カードが USB トークンです。USB トークンでは、記憶域の容量（32KB）内であれば、どのようなタイプのファイルでもセキュアに保存できます。USB トークンに保存されたコンフィギュレーション ファイルに対する暗号化およびアクセスは、ユーザ PIN を介してだけ行えます。デバイスにコンフィギュレーションファイルをロードするには、デバイスのコンフィギュレーション ファイルをセキュアに配布できるよう適切な PIN が設定されている必要があります。

USB トークンをデバイスに装着したら、その USB トークンにログインする必要があります。ログイン後は、ユーザ PIN（デフォルトは 1234567890）や、ログインが拒否されるようになるまで許容されるログイン試行の失敗回数（デフォルトは 15 回）など、さまざまなデフォルト設定を変更できます。USB トークンのアクセス方法および設定方法については、「USB トークンへのログインと USB トークンの設定」を参照してください。

USB トークンへ正常にログインした場合は、**copy** コマンドを使用して、デバイスから USB トークンへファイルをコピーできます。USB トークンの RSA キーおよび関連する IPsec トンネルは、デバイスがリロードされるまで使用できます。キーが削除され IPsec トンネルが切断されるまでの時間を指定する場合は、**crypto pki token removal timeout** コマンドを発行します。デフォルト タイムアウトはゼロのため、eToken がデバイスから削除されると RSA キーが自動的に削除されるようになります。デフォルト値は、実行中のコンフィギュレーションで次のように表示されます。

```
crypto pki token default removal timeout 0
```

次の表に、USB トークンの機能を示します。

表 *1 : USB* トークンの主な機能性

機能	USB トークン
アクセシビリティ	デジタル証明書、事前共有キー、およびデバイス設定を USB トークンからデバイスへセキュアに保存したり転送したりするためのものです。
ストレージのサイズ	32 KB または 64 KB

機能	USB トークン
ファイル タイプ	• 通常、IPsec VPN 用のデジタル証明書、事前共有キー、およびデバイス設定を保存する場合には、ファイル タイプを指定します。 • USB トークンには、Cisco IOS イメージは保存できません。
セキュリティ	• ファイルに対する暗号化およびアクセスは、ユーザ PIN を介してだけ行えます。 • ファイルは、ノンセキュアなフォーマットでも保存できます。
ブート設定	• デバイスではブート時に、USB トークンに保存されている設定を使用できます。 • デバイスではブート時に、USB トークンに保存されているセカンダリ設定を使用できます（セカンダリ設定を使用すると、ユーザは各自の IPsec 設定をロードできます）。

USB トークンの応用上の利点

Cisco ルータ上で USB トークンがサポートされていることにより、応用上次のような利点が生まれます。

移動可能な証明書：配置する VPN クレデンシャルを外部デバイスに保存できます。

USB トークンでは、スマート カード テクノロジーにより、IPsec VPN の導入に必要なデジタル証明書や設定を保存できます。これにより、ルータにおいて RSA 公開キーを生成し、少なくとも 1 つの IPsec トンネルを認証できるようになりました（ルータでは複数の IPsec トンネルを開始できるため、USB トークンには、必要に応じて複数の証明書を保存できるようになっています）。

VPN クレデンシャルを外部デバイスに保存すると、機密データが漏洩する危険性は低くなります。

ファイルをセキュアに配置するための PIN 設定

USB トークンには、ユーザが設定した PIN を介してルータにおける暗号化をイネーブルにする際に使用できるコンフィギュレーション ファイルを保存できます（つまり、デジタル証明書、事前共有キー、および VPN は使用されません）。

軽減されるまたは不要になる手動での設定作業

USB トークンを使用すると、リモート ソフトウェアの設定やプロビジョニングの際、手動で行う作業がほとんど（あるいは完全に）必要なくなります。設定は自動プロセスとして構成されます。具体的には、ルータに装着した USB トークンにブートストラップ設定を保存しておくと、そのブートストラップ設定によりルータが起動します。さらにこのルータは、ブートス

トラップ設定によって TFTP サーバへ接続され、その TFTP サーバに保存されている設定に基づいて、すべてのルータ設定が行われます。

RSA 処理

USB トークンは、ストレージ デバイス以外に、暗号化装置として使用できます。USB トークンを暗号化装置として使用すると、トークンでキー生成、署名、認証などの RSA 操作を実行できます。

ご使用のトークン ストレージ デバイス上に配置されているクレデンシャルからは、モジュラスが 2048 ビット以下の汎用 RSA キー ペア、特殊 RSA キー ペア、暗号化 RSA キー ペア、またはシグニチャ RSA キーペアを生成できます。秘密キーは、デフォルトでは配布されず、トークン上に保存されたままです。ただし、秘密キーの保管場所を設定することは可能です。

USB トークン上に常駐するキーは、生成された段階でトークンの永続的な保管場所に保存されます。キーの削除操作を行うと、トークンに保存されているキーは、永続的な保管場所からただちに削除されます（トークン上に常駐していないキーは、**write memory** またはそれに類するコマンドが発行されると、トークン以外の保管場所で保存や削除が行われます）。

セキュア デバイス プロビジョニング（SDP）環境におけるリモート デバイスの設定およびプロビジョニング

SDP は USB トークンの設定に使用される場合があります。設定された USB トークンを送付すれば、リモート ロケーションにあるデバイスをプロビジョニングできます。つまり、あるネットワーク デバイスから別のリモート ネットワーク デバイスへ暗号化された情報を送る際に USB トークンを使用することで、USB トークンを段階的に配置できます。

SDP で USB トークンを使用する方法については、「その他の関連資料」に記載されている参照先を参照してください。

PKI データの保管場所の設定方法

証明書のローカル ストレージ場所の指定

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki certificate storage** *location-name*
4. **exit**
5. **copy** *source-url destination-url*
6. **show crypto pki certificates storage**

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 certificate storage** *location-name* 例： `Device(config)# crypto pki certificate storage flash:/certs`	証明書のローカルな保管場所を指定します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	**copy** *source-url destination-url* 例： `Device# copy system:running-config nvram:startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションに保存します。 （注） 設定は、実行コンフィギュレーションがスタートアップ コンフィギュレーションに保存された場合にだけ有効になります。
ステップ 6	**show crypto pki certificates storage** 例： `Device# show crypto pki certificates storage`	（任意）PKI 証明書の保管場所に関する現在の設定を表示します。

例

次に、**show crypto pki certificates storage** コマンドの出力例を示します。ここでは、証明書が disk0 の certs サブディレクトリに保存されています。

```
Device# show crypto pki certificates storage
Certificates will be stored in disk0:/certs/
```

Cisco デバイスにおける USB トークンの設定と使用

USB トークンによる設定の保存

手順の概要

1. **enable**
2. **configure terminal**
3. **boot config** *usbtoken[0-9]:filename*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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	**boot config** *usbtoken[0-9]:filename* 例： `Device(config)# boot config usbtoken0:file`	スタートアップ コンフィギュレーション ファイルがセキュアなUSBトークンに保存されるよう指定します。

USB トークンへのログインと USB トークンの設定

RSA キーと USB トークンの併用方法

- RSA キーは、USB トークンがルータへ正常にログインした後にロードされます。
- デフォルトの場合、新規に生成された RSA キーは、最後に装着された USB トークンに保存されます。再生成されたキーは、元の RSA キーが生成されたのと同じ場所に保存する必要があります。

手動ログイン用のデバイスの設定

自動ログインとは異なり、手動ログインを使用する場合は、ユーザが実際のUSBトークンPINを把握している必要があります。

（注） 手動ログインまたは自動ログインのいずれかを使用する必要があります。

手動ログインは、PIN をデバイス上に保存するのが適していない場合に使用できます。また、初期導入時やハードウェア交換時に、デバイスを現地の業者から調達したり、リモートサイトへ直送したりする場合にも、手動ログインが適しています。手動ログインは、権限の有無にかかわらず実行できます。また、手動ログインを実行すると、USB トークン上のファイルおよび RSA キーが、Cisco IOS ソフトウェアで使用可能になります。セカンダリ コンフィギュレーション ファイルを設定する場合は、ログインを実行するユーザの権限がある場合にだけ手動ログインを実行できます。そのため、何らかの目的で、手動ログインを実行し、USB トークン上にセカンダリ コンフィギュレーション ファイルを設定する場合は、権限をイネーブルにする必要があります。

手動ログインは、失われたデバイス設定のリカバリを行う場合にも使用できます。通常 VPN を使用してコア ネットワークへ接続しているリモート サイトが存在する状況では、設定および RSA キーが失われた場合、USB トークンが備えているアウトオブバンド サービスが必要となります。USB トークンには、ブート設定、セカンダリ設定、および接続を認証するための RSA キーを保存できます。

手順の概要

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]
3. **show usbtoken** *0-9*:*filename*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 token** *token-name* [**admin**] **login** [*pin*] 例： `Device# crypto pki token usbtoken0 admin login 5678`	USB トークンに手動でログインします。 **admin** キーワードを最初に指定していない場合は、このキーワードオプションで **crypto pki token** コマンドを再び入力できます。
ステップ 3	**show usbtoken** *0-9*:*filename* 例： `Device# show usbtoken0:usbfile`	（任意）USB トークンがデバイスにログインしているかどうかを確認します。

次の作業

USB トークンへのログインが完了すると、次のような作業が行えます。

• USB トークンを詳細に設定する。「USB トークンの設定」の項を参照してください。

- ユーザ PIN の変更、ルータから USB トークンに設定されたキーの保管場所へのファイルのコピー、USB トークンの変更など、USB トークンの管理作業を行う。「USB トークンにおける管理機能の設定」の項を参照してください。

USB トークンの設定

USB トークンに対しては、自動ログインの設定後、さらに次のような設定を行えます。

PIN およびパスフレーズ

自動ログインにおける PIN のセキュリティをさらに強化するため、NVRAM に保存されている PIN を暗号化し、USB トークンにパスフレーズを設定できます。パスフレーズを設定すると、他のユーザには PIN そのものではなく、そのパスフレーズを周知すればよいため、PIN の安全性を維持できます。

このパスフレーズは、USB トークンをデバイスに装着した後、PIN を復号化する際に必要となります。PIN が復号化されると、デバイスはその PIN を使用して USB トークンにログインします。

（注） ユーザがログインするには特権レベル 1 が必要です。

USB トークンのロック/ロック解除

USB トークン自体をロック（暗号化）またはロック解除（復号化）できます。

USB トークンは、ロック解除すると使用できるようになります。ロック解除した場合、Cisco IOS ソフトウェアでは、その USB トークンは自動ログインされたものと見なされ、その USB トークン上にあるいずれかのキーがロードされます。また、セカンダリ コンフィギュレーション ファイルがトークン上に存在する場合は、ログインしたユーザの権限レベルとは独立したフル ユーザ権限（権限レベル 15）を使用して、そのセカンダリ コンフィギュレーション ファイルが実行されます。

トークンをロックした場合は、トークンからログアウトする場合とは異なり、トークンからロードされた RSA キーがすべて削除され、セカンダリ アンコンフィギュレーション ファイルが（もし設定されていれば）実行されます。

セカンダリ コンフィギュレーション ファイルとセカンダリ アンコンフィギュレーション ファイル

USB トークン上に存在するコンフィギュレーション ファイルは、セカンダリ コンフィギュレーション ファイルと呼ばれます。セカンダリ コンフィギュレーション ファイルを作成および設定する場合、セカンダリ コンフィギュレーション ファイルの有無は、NVRAM に保存された Cisco IOS 設定内のセカンダリ コンフィギュレーション ファイル オプションの存在によって決定されます。ユーザがトークンを取り外した後またはトークンからログアウトした後に、無効タイマーで設定された期間が経過すると、別途用意されているセカンダリ アンコンフィギュレーション ファイルが実行され、セカンダリ コンフィギュレーションのすべての要素が、実行コンフィギュレーションから削除されます。セカンダリ コンフィギュレーション ファイル

およびセカンダリ アンコンフィギュレーション ファイルは、ログインしたユーザの権限レベルとは関係なく、権限レベル 15 で実行されます。

手順の概要

1. **enable**
2. **crypto pki token** *token-name* **unlock** [*pin*]
3. **configure terminal**
4. **crypto pki token** *token-name* **encrypted-user-pin** [**write**]
5. **crypto pki token** *token-name* **secondary unconfig** *file*
6. **exit**
7. **crypto pki token** *token-name* **lock** [*pin*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 token** *token-name* **unlock** [*pin*] 例： `Device# crypto pki token mytoken unlock mypin`	（任意）ロックされている USB トークンを使用できるようにします。 ロック解除した場合、Cisco IOS ソフトウェアでは、その USB トークンは自動ログインされたものと見なされ、その USB トークン上にあるいずれかのキーがロードされます。また、セカンダリ コンフィギュレーション ファイルが存在する場合、このファイルは実行されます。
ステップ 3	**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 4	**crypto pki token** *token-name* **encrypted-user-pin** [**write**] 例： `Device(config)# crypto pki token mytoken encrypted-user-pin write`	（任意）NVRAM に保存されている PIN を暗号化します。
ステップ 5	**crypto pki token** *token-name* **secondary unconfig** *file* 例：	（任意）セカンダリ コンフィギュレーション ファイルとその保管場所を指定します。

	コマンドまたはアクション	目的
	`Device(config)# crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg`	
ステップ 6	**exit** 例： `Device(config)# exit`	特権 EXEC モードを開始します。
ステップ 7	**crypto pki token** *token-name* **lock** [*pin*] 例： `Device# crypto pki token mytoken lock mypin`	（任意）トークンからロードされたRSAキーをすべて削除し、セカンダリ アンコンフィギュレーション ファイルが存在する場合は、それを実行します。

例

次の例は、ユーザ PIN の設定、ユーザ PIN の暗号化、デバイスのリロード、およびユーザ PIN のロック解除の各プロセスを順に示したものです。

```
! Configuring the user PIN

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

Device(config)# crypto pki token usbtoken0: userpin

Enter password: mypassword

! Encrypt the user PIN

Device(config)# crypto pki token usbtoken0: encrypted-user-pin

Enter passphrase: mypassphrase

Device(config)# exit

Device#

Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console

Device# show running config

crypto pki token usbtoken0 user-pin *encrypted*

! Reloading the router.

Device> enable

Password:

! Decrypting the user pin.

Device# crypto pki token usbtoken0: unlock

Token eToken is usbtoken0

Enter passphrase: mypassphrase
```

```
Token login to usbtoken0(eToken) successful

Device#

Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken

Login Successful
```

次に示すのは、実行コンフィギュレーションからセカンダリ コンフィギュレーションの要素を削除する際に使用されるセカンダリ アンコンフィギュレーション ファイルの設定例です。セカンダリ コンフィギュレーション ファイルを使用して PKI トラストポイントが設定されている場合を例にとると、それに対応するアンコンフィギュレーション ファイル mysecondaryunconfigfile.cfg には、次のようなコマンド ラインが設定されます。

```
no crypto pki trustpoint token-tp
```

トークンが取り外された後で、次のコマンドが実行されると、デバイスの実行コンフィギュレーションから、トラストポイントおよびそれに関連付けられた証明書が削除されます。

```
 Device# configure terminal
Device(config)# no crypto pki token mytoken secondary unconfig mysecondaryunconfigfile.cfg
```

次の作業

USB トークンへのログインおよび USB トークンの設定が完了すると、次のような作業が行えます。ユーザ PIN の変更、ルータから USB トークンに設定されたキーの保管場所へのファイルのコピー、USB トークンの変更など、USB トークンの管理作業を行う。「USB トークンにおける管理機能の設定」の項を参照してください。

USB トークンにおける管理機能の設定

ここでは、ユーザPIN、USBトークンに対するログイン試行の失敗回数の上限、クレデンシャルの保管場所など、さまざまなデフォルト設定を変更する手順について説明します。

手順の概要

1. **enable**
2. **crypto pki token** *token-name* **admin**] **change-pin** [*pin*]
3. **crypto pki token** *token-name device-name***: label** *token-label*
4. **configure terminal**
5. **crypto key storage** *device-name***:**
6. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *device-name***:**] [**redundancy**] [**on** *device-name*]**:**
7. **crypto key move rsa** *keylabel* [**non-exportable** | [**on** | **storage**]] *location*
8. **crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]
9. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
10. **exit**
11. **copy usbflash**[*0-9*]**:***filename destination-url*
12. **show usbtoken**[*0-9*]**:***filename*

13. **crypto pki token** *token-name* **logout**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • 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 token** *token-name* **admin**] **change-pin** [*pin*] 例： `Device# crypto pki token usbtoken0 admin change-pin`	（任意）USB トークン上のユーザ PIN 番号を変更します。 • PIN が変更されない場合は、デフォルトの PIN（1234567890）が使用されます。 （注） PIN の変更後は、ログインの失敗回数を 0 にリセットする必要があります（**crypto pki token max-retries** コマンドを使用）。許容されるログインの失敗回数の上限は、15（デフォルト）に設定されています。
ステップ 3	**crypto pki token** *token-name device-name*: **label** *token-label* 例： `Device# crypto pki token mytoken usb0: label newlabel`	（任意）USB トークンの名前を設定または変更します。 • *token-label* 引数には、英数字（ダッシュおよびアンダースコアを含む）からなる 31 文字以下の文字列を指定できます。 ヒント このコマンドは、自動ログインやセカンダリ コンフィギュレーション ファイルなどのトークン固有の設定用として複数の USB トークンを設定する場合に有用です。
ステップ 4	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 5	**crypto key storage** *device-name*: 例： `Device(config)# crypto key storage usbtoken0:`	（任意）新規作成した RSA キーに対するデフォルトの保管場所を設定します。 （注） 設定の内容にかかわらず、既存のキーは、ロード元のデバイスに保存されます。

	コマンドまたはアクション	目的
ステップ 6	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *device-name***:**] [**redundancy**] [**on** *device-name*]**:** 例： `Device(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:`	（任意）証明書サーバの RSA キー ペアを生成します。 • **storage** キーワードを使用すると、キーの保管場所を指定できます。 • *key-label* 引数を指定することによってラベル名を指定する場合、**crypto pki server** *cs-label* コマンドによって証明書サーバ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、デバイスの完全修飾ドメイン名（FQDN）であるデフォルト値が使用されます。 **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 キーのモジュラス サイズは 1024 ビットです。推奨される CA キーのモジュラスは 2048 ビットです。CA キーのモジュラス サイズの範囲は 350 ～ 4096 ビットです。 • **on** キーワードは、指定したデバイス上で 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 7	**crypto key move rsa** *keylabel* [**non-exportable** | [**on** | **storage**]] *location* 例： `Device(config)# crypto key move rsa keypairname non-exportable on token`	（任意）既存の Cisco IOS クレデンシャルを、現在の保管場所から指定した保管場所へ移動します。 デフォルトの場合、RSA キー ペアは現在のデバイス上に保存されたままになります。 デバイス上でキーを生成しそれをトークンに移動するまでの所要時間は 1 分未満です。トークン上でキーを生成する際に **on** キーワードを使用すると、

	コマンドまたはアクション	目的
		USB トークン上で使用可能なハードウェアキー生成ルーチンに応じて、5 ～ 10 分程度の時間がかかります。 Cisco IOS で生成された既存の RSA キー ペアが USB トークンに保存され、登録に使用される場合は、それら既存の RSA キー ペアを代替場所に移動して永続的に保存する必要があります。 このコマンドは、USB トークンと SDP を使用してクレデンシャルを配置する場合に有用です。
ステップ 8	**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*] 例： `Device(config)# crypto pki token usbtoken0 removal timeout 60`	（任意）USB トークンがデバイスから取り外されてから、USB トークンに保存されている RSA キーが削除されるまで、デバイスが待機する時間を秒単位で設定します。 （注） このコマンドが発行されない場合は、USB トークンがデバイスから取り外された直後に、すべての RSA キーが削除されるほか、USB トークンに関連付けられている IPsec トンネルもすべて切断されます。
ステップ 9	**crypto pki token** {*token-name* | **default**} **max-retries** [*number*] 例： `Device(config)# crypto pki token usbtoken0 max-retries 20`	（任意）USB トークンへのアクセスが拒否されるまでに許容されるログイン試行の連続失敗回数の上限を設定します。 • デフォルト値は 15 です。
ステップ 10	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 11	**copy usbflash**[*0-9*]**:***filename destination-url* 例： `Device# copy usbflash0:file1 nvram:`	USB トークンからデバイスへファイルをコピーします。 • *destination-url*：サポートされているオプションのリストについては、**copy** コマンドに関するセクションを参照してください。
ステップ 12	**show usbtoken**[*0-9*]**:***filename* 例： `Device# show usbtoken:usbfile`	（任意）USB トークンに関する情報を表示します。このコマンドを使用すると、USB トークンがデバイスにログインしているかどうかを確認できます。

	コマンドまたはアクション	目的
ステップ 13	**crypto pki token** *token-name* **logout** 例： `Device# crypto pki token usbtoken0 logout`	USB トークンからデバイスをログアウトします。 （注） USD トークンに何らかのデータを保存する場合は、再度トークンにログインする必要があります。

USB トークンに関するトラブルシューティング

ここでは、次の各 Cisco IOS コマンドについて説明します。これらのコマンドは、USB トークンの使用中に発生しうる問題についてのトラブルシューティングに使用できます。

USB ポート接続のトラブルシューティング

show file systems コマンドを使用すると、USB モジュールが USB ポートに差し込まれていることをルータが認識しているかどうかを判定できます。差し込まれている USB モジュールは、ファイルシステムのリスト上に表示されます。これらのモジュールがリスト上に表示されない場合は、次のいずれかの問題が発生している可能性があります。

- USB モジュールとの接続に問題がある。
- ルータ上で稼働している Cisco IOS イメージによりサポートされていない USB モジュールがある。
- USB モジュールそのものにハードウェア上の問題がある。

次に示すのは、 **show file systems** コマンドによる出力例です。この中には USB トークンも表示されています。USB モジュールが現れるのはリストの最下行です。

```
Device# show file systems
File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
      491512      486395     nvram     rw   nvram:
           -           -    opaque     wo   syslog:
           -           -    opaque     rw   xmodem:
           -           -    opaque     rw   ymodem:
           -           -   network     rw   rcp:
           -           -   network     rw   pram:
           -           -   network     rw   ftp:
           -           -   network     rw   http:
           -           -   network     rw   scp:
           -           -   network     rw   https:
           -           -    opaque     ro   cns:
    63158272    33037312  usbflash     rw   usbflash0:
       32768         858  usbtoken     rw   usbtoken1:
```

シスコによりサポートされている USB トークンの特定

show usb device コマンドを使用すると、USB トークンがシスコによりサポートされているかどうかを判定できます。このコマンドの次の出力例では、太字で記されているのが、モジュールがサポートされているかどうかを示す箇所です。

```
Router# show usb device
Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0
Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA
    Interface:
        Number:0
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

USB トークンのデバイス問題の特定

show usb controllers コマンドを使用すると、USB フラッシュモジュールにハードウェア上の問題があるかどうかを判別できます。**show usb controllers** コマンドの出力結果にエラーが表示された場合は、USB モジュールにハードウェア上の問題があると考えられます。

USB フラッシュモジュールに対するコピー操作が正常に行われていることを確認する場合にも、この **show usb controllers** コマンドを使用できます。ファイルのコピーを実行した後で、**show usb controllers** コマンドを発行すると、データ転送が正常に行われたことを示す内容が表示されます。

次に示すのは、使用中の USB フラッシュモジュールの **show usb controllers** コマンドによる出力例です。

```
Router# show usb controllers
Name:1362HCD
```

```
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF
Int Level:1
Transfer Completion Codes:
         Success              :920            CRC             :0
         Bit Stuff            :0              Stall           :0
         No Response          :0              Overrun         :0
         Underrun             :0              Other           :0
         Buffer Overrun       :0              Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2              Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0              Bulk Transfer   :0
         Isochronous Transfer :0              Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0              Bulk Transfer   :26
         Isochronous Transfer :0              Control Transfer:894
USBD Failures:
         Enumeration Failures :0              No Class Driver Found:0
         Power Budget Exceeded:0
USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3              Command Fail    :0
         Good Status Timed out:0              Device not Found:0
         Device Never Opened  :0              Drive Init Fail :0
         Illegal App Handle   :0              Bad API Command :0
         Invalid Unit Number  :0              Invalid Argument:0
         Application Overflow :0              Device in use   :0
         Control Pipe Stall   :0              Malloc Error    :0
         Device Stalled       :0              Bad Command Code:0
         Device Detached      :0              Unknown Error   :0
         Invalid Logic Unit Num:0
USB Aladdin Token Driver Counters:
         Token Inserted       :1              Token Removed   :0
```

```
        Send Insert Msg Fail :0                 Response Txns   :434
        Dev Entry Add Fail   :0                 Request Txns    :434
        Dev Entry Remove Fail:0                 Request Txn Fail:0
        Response Txn Fail    :0                 Command Txn Fail:0
        Txn Invalid Dev Handle:0
USB Flash File System Counters:
        Flash Disconnected   :0                 Flash Connected :1
        Flash Device Fail    :0                 Flash Ok        :1
        Flash startstop Fail :0                 Flash FS Fail   :0
USB Secure Token File System Counters:
        Token Inserted       :1                 Token Detached  :0
        Token FS success     :1                 Token FS Fail   :0
        Token Max Inserted   :0                 Create Talker Failures:0
        Token Event          :0                 Destroy Talker Failures:0
        Watched Boolean Create Failures:0
```

USB トークン情報の表示

dir コマンドと **filesystem** キーワードオプション **usbtoken***0-9***:** を使用すると、USB トークン上にあるすべてのファイル、ディレクトリ、およびそれらの権限文字列を表示できます。

次の出力例は、USB トークンに関する情報を表示したものです。

```
Device# dir usbtoken1:
Directory of usbtoken1:/
    2  d---           64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---         4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---            0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---          512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---            0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---            0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---            0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----          940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----         1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

次の出力例では、デバイスが認識しているすべてのデバイスのディレクトリ情報を表示します。

```
Device# dir all-filesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles
No space information available
Directory of flash:/
    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T
129880064 bytes total (99753984 bytes free)
Directory of nvram:/
  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data
491512 bytes total (486395 bytes free)
```

```
Directory of usbflash0:/
    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/
    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config
32768 bytes total (858 bytes free)
```

PKI データの保存に関する設定例

例：特定のローカルな保管場所への証明書の保存

次に示すのは、certsサブディレクトリに証明書を保存する場合の設定例です。ここでは、certsサブディレクトリは存在しないため、自動的に作成されています。

```
Router# dir nvram:
 114  -rw-        4687                    <no date>  startup-config
 115  ----        5545                    <no date>  private-config
 116  -rw-        4687                    <no date>  underlying-config
   1  ----          34                    <no date>  persistent-data
   3  -rw-         707                    <no date>  ioscaroot#7401CA.cer
   9  -rw-         863                    <no date>  msca-root#826E.cer
  10  -rw-         759                    <no date>  msca-root#1BA8CA.cer
  11  -rw-         863                    <no date>  msca-root#75B8.cer
  24  -rw-        1149                    <no date>  storagename#6500CA.cer
  26  -rw-         863                    <no date>  msca-root#83EE.cer
129016 bytes total (92108 bytes free)
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
  14  -rw-         707  May 27 2005 02:09:02 +00:00  ioscaroot#7401CA.cer
  15  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#826E.cer
  16  -rw-         759  May 27 2005 02:09:02 +00:00  msca-root#1BA8CA.cer
  17  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#75B8.cer
  18  -rw-        1149  May 27 2005 02:09:02 +00:00  storagename#6500CA.cer
  19  -rw-         863  May 27 2005 02:09:02 +00:00  msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)
! The certificate files are now on disk0/certs:
```

例：USB トークンへのログインと USB トークンへの RSA キーの保存

次に示すのは、USB トークンにログインして RSA キーを生成し、その RSA キーを USB トークンに保存する場合の設定例です。

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the
eToken ! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

次に示すのは、USB トークンから正常にロードされた保存済みログイン情報の **show crypto key mypubkey rsa** コマンドによる出力例です。USB トークン上に保存されているクレデンシャルは、保護領域内に存在します。USB トークン上にクレデンシャルを保存する場合、それらのファイルは /keystore というディレクトリに保存されます。ただし、キー ファイルは、コマンドライン インターフェイス（CLI）では表示されません。

```
Router#
show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
```

```
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
ルータへの USB モジュールの接続	『*Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide*』
eToken および USB フラッシュのデータ シート	『*USB eToken and USB Flash Features Support*』
RSA キー	PKI 内での RSA キーの展開
ファイル管理（ファイルのロード、コピー、および再起動）	『*Cisco Configuration Fundamentals Configuration Guide*』（Cisco.com）
USB トークンによる RSA 処理：証明書サーバの設定	「PKI 展開での Cisco IOS 証明書サーバの設定および管理」の機能に関する資料。 「Generating a Certificate Server RSA Key Pair」項、「Configuring a Certificate Server Trustpoint」項、および関連する例を参照してください。
USB トークンの RSA 処理：初期自動登録時における USB トークンを使用した RSA 処理	『Configuring Certificate Enrollment for a PKI』の「Configuring Certificate Enrollment or Autoenrollment」項を参照してください。
SDP のセットアップ、設定、および USB トークンとの使用	PKI クレデンシャルの展開での SDP と USB トークンの使用方法については、「Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI」にある機能名の機能情報の項を参照してください。

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

PKI クレデンシャルの保存に関する機能情報

表 *2 : PKI* クレデンシャルの保存に関する機能情報

機能名	リリース	機能情報
証明書：保管場所の指定		この機能を使用すると、証明書を個別のファイルとして保存する機能をサポートしているプラットフォームにおいて、ローカル証明書の保管場所を指定できます。シスコのプラットフォームはすべて、デフォルトの保管場所として使用する NVRAM、およびフラッシュ ローカル ストレージをサポートしています。ご使用のプラットフォームによっては、ブートフラッシュ、スロット、ディスク、USB フラッシュ、USB トークンなど、サポートされているその他のローカル ストレージを使用できます。 この機能により、次のコマンドが導入されました。**crypto pki certificate storage**、**show crypto pki certificates storage**。
ソフトウェア暗号エンジンサポートでの RSA 4096 ビット キー生成	15.1(1)T	**crypto key generate rsa** コマンドの **modulus** キーワードの値の範囲は、360 ～ 2048 ビットから 360 ～ 4096 ビットに拡張されました。

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。