CHAPTER 2

証明書およびキーの管理

（注） この章の情報は、特に記載のない限り、ACE モジュールと ACE アプライアンスの両方に適用されます。この章で説明する機能は、特に記載のない限り、IPv4 と IPv6 に適用されます。

この章では、インポートとエクスポートの機能を使用して、Cisco ACE アプリケーション コントロール エンジン でさまざまな証明書および RSA キー ペア ファイルを管理する方法を説明します。また、認証局（CA）から証明書を取得するために証明書署名要求（CSR）を作成して送信するプロセスについても説明します。

この章の内容は、次のとおりです。

SSL デジタル証明書とキー ペア

デジタル証明書とキー ペアは、ユーザを認証するためのデジタル識別情報の一種です。VeriSign や Thawte などの CA は、公開キーの有効性を証明する証明書を発行します。クライアント証明書またはサーバ証明書には、次の識別情報属性があります。

- CA（証明書発行者）の名前と CA のデジタル署名
- シリアル番号
- 証明書で認証されるクライアントまたはサーバの名前（証明書サブジェクト）
- サブジェクトの公開キー
- 証明書の有効期限を示すタイム スタンプ

CA には、SSL 証明書と証明書失効リスト（CRL）を作成するために使用する 1 つ以上の署名証明書があります。各署名証明書には、CA 署名の作成に使用される照合秘密キーがあります。CA は（公開キーが組み込まれている）署名証明書を公開するため、SSL 証明書または CRL が実際に特定の CA により署名されたものであることを確認する場合には、この署名証明書にアクセスし使用することができます。

ACE では、次のアプリケーション用に証明書および対応するキー ペアが必要です。

- SSL 終了：ACE が SSL プロキシ サーバとして動作し、クライアントとの間の SSL セッションを終端します。SSL 終了の場合、サーバ証明書および対応するキー ペアを取得する必要があります。
- SSL 開始：ACE がクライアントとして動作し、SSL サーバとの間の SSL セッションを開始します。SSL 開始ではクライアント証明書および対応するキー ペアを使用できますが、SSL サーバでクライアント認証が有効でない限り、必要ではありません。

（注） ACE は、サーバ証明書とクライアント証明書としてではなく、認証局（CA）としてのみ、4096 の証明書と 4096 のキー ペアをサポートします。また、ワイルドカードの証明書もサポートします。

2 台のデバイスが SSL セッションを確立するために、RSA キー ペアは、SSL ハンドシェイク時に ACE とそのピアで必要です。キー ペアとは、公開キーおよびそれに対応する秘密キーを意味します。ハンドシェイク時に RSA キー ペアを使用してセッション キーが暗号化され、そのセッション キーは、ハンドシェイクのあと両方のデバイスがデータを暗号化するために使用します。

SSL ハンドシェイク プロセスの詳細については、第 1 章「概要」の「SSL ハンドシェイク」の項を参照してください。

SSL 終了と SSL 開始用に ACE を設定する前に、デジタル証明書および対応する公開キーと秘密キー ペアを、必要な ACE コンテキストにインポートします。

冗長構成では、ACE で、アクティブ コンテキスト内に存在する SSL 証明書およびキー ペアとフォールト トレラント（FT）グループのスタンバイ コンテキストの同期は行われません。ACE で設定の同期が実行され、スタンバイで必要な証明書とキーが見つからなかった場合は、設定同期が失敗して、スタンバイ コンテキストが STANDBY_COLD ステートに移行します。

スタンバイ コンテキストに証明書とキーをコピーするには、**crypto export** コマンドを使用して、アクティブ コンテキストから FTP または TFTP サーバに証明書とキーをエクスポートしてから、**crypto import** コマンドを使用して、スタンバイ コンテキストに証明書とキーをインポートします。また、アクティブ コンテキストに証明書をインポートするときに使用したのと同じ方法で、スタンバイ コンテキストに証明書とキーを直接インポートすることもできます。この 2 番目の方法は、証明書とキーがエクスポート不可能としてアクティブ コンテキストにインポートされた場合に必要です。証明書とキーのインポートおよびエクスポートに関する詳細については、「証明書とキー ペア ファイルのインポートまたはエクスポート」を参照してください。

この場合、スタンバイ コンテキストを STANDBY_HOT ステートに戻すには、必要な SSL 証明書とキーをスタンバイ コンテキストにインポートしてから、FT グループのアクティブ コンテキスト内のコンフィギュレーション モードで、次のコマンドを入力して、アクティブ コンテキスト設定のバルク同期を実行します。

1. **no ft auto-sync running-config**
2. **ft auto-sync running-config**

冗長性の詳細については、『*Administration Guide, Cisco ACE Application Control Engine*』を参照してください。

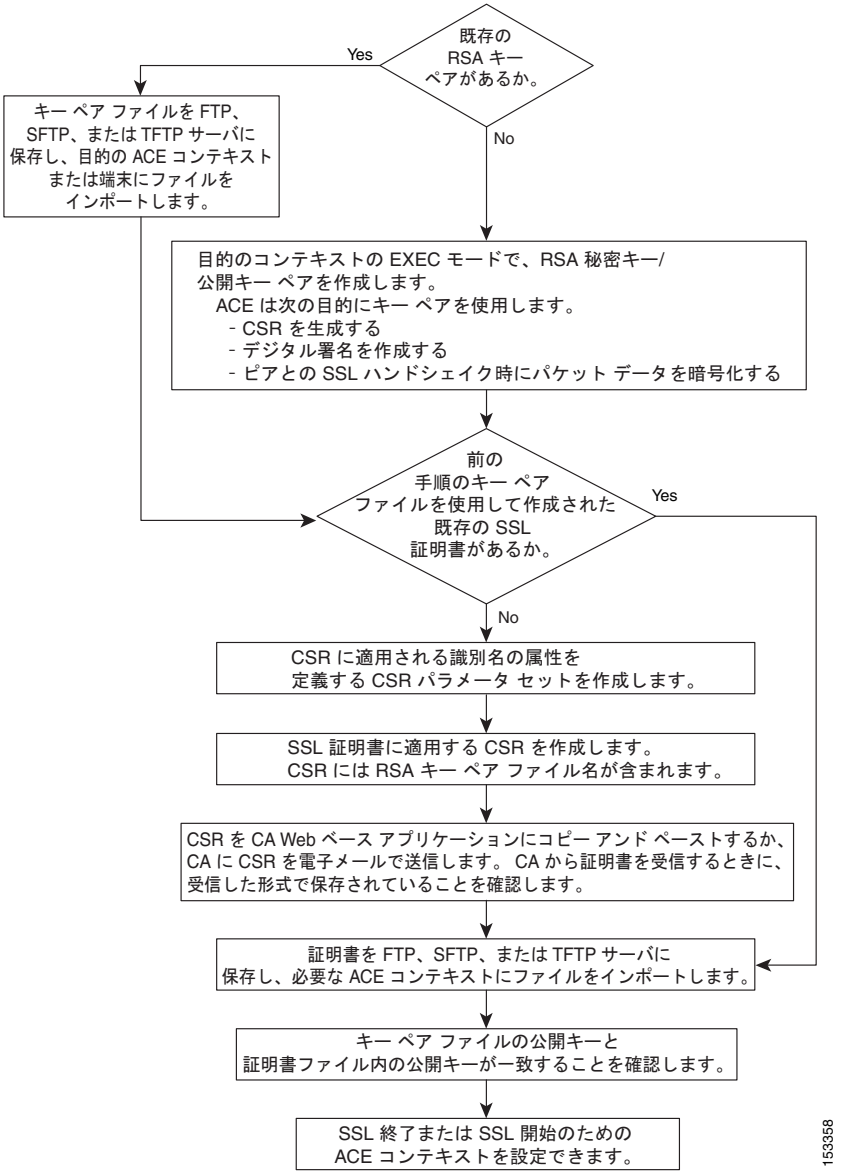
証明書および対応するキー ペアがない場合は、ACE を使用して、RSA キー ペア（2048 ビットまで）と証明書署名要求（CSR）を生成できます。CA に証明書を申請する必要がある場合は、CSR を作成できます。CA は CSR に署名し、認証したデジタル証明書を返します。

ACE は、サンプル証明書およびキー ペア ファイルをデモ目的でも提供します。これらのファイルの詳細については、「ACE のサンプル証明書とキー ペアの使用」の項を参照してください。

（注） キー ペアの生成時や証明書のインポート時に強力なセキュリティ ポリシーを実施するには、ACE のユーザ ロールを理解する必要があります。ユーザ ロールの詳細については、『*Virtualization Guide, Cisco ACE Application Control Engine*』を参照してください。

図 2-1 は、ACE 用に RSA キー ペアと SSL 証明書を設定する方法の概要を示します。

図 2-1　SSL キーおよび証明書の設定の概要



ACE のサンプル証明書とキー ペアの使用

ACE には、プレインストールされた証明書とキー ペアが含まれています。この証明書は、デモ目的でのみ使用され、有効なドメインはありません。この証明書は、cisco-sample-cert という名前の基本的な拡張子を使用して自己署名された証明書です。このキー ペアは、cisco-sample-key という名前の RSA 1024 ビット キー ペアです。これらのファイルは、**show crypto files**、**show crypto key**、および **show crypto certs** コマンドを使用して表示できます。

これらのファイルを SSL プロキシ サービスに追加するには、**cert** コマンドと **key** コマンドを使用します。詳細については、「証明書の指定」の項と「キー ペアの指定」の項を参照してください。これらのファイルは、同じファイル名で、任意のコンテキストで使用できます。**crypto verify** コマンドを使用して、キー ペアを確認できます。

ACE ではこれらのファイルをエクスポートできますが、これらの名前でファイルをインポートすることはできません。**crypto delete** コマンドを使用してこれらのファイルを削除することはできません。ただし、ACE をアップグレードする場合、これらのファイルはアップグレード イメージで提供されるファイルで上書きされます。

キー ペアおよび証明書署名要求の生成

既存の証明書および一致するキー ペアがない場合、ACE には、キー ペアまたは CSR を生成するための、一連の証明書およびキー管理ユーティリティが含まれています。CA が CSR に署名すると、ACE で使用できる証明書になります。

既存の証明書および対応するキー ペアがある場合は、ACE の目的のコンテキストにインポートできます。証明書と秘密キーのインポートの詳細については、「証明書とキー ペア ファイルのインポートまたはエクスポート」の項を参照してください。

ここでは、次の内容について説明します。

- RSA キー ペアの生成
- CSR パラメータ セットの作成および定義
- 証明書署名要求の生成

RSA キー ペアの生成

ACE は、最大 2048 ビットの RSA キー ペアの生成をサポートします。RSA キー ペアを生成するには、EXEC モードで **crypto generate key** コマンドを使用します。

このコマンドの構文は次のとおりです。

crypto generate key [**non-exportable**] *bitsize filename*

引数およびキーワードは次のとおりです。

- **non-exportable**：（任意）ACE がキー ペアをエクスポート不可能としてマークすることを指定します。この場合、キー ペア ファイルを ACE からエクスポートできません。
- *bitsize*：キー ペアのセキュリティ強度です。Web トランザクションの安全を確保するために使用される RSA キー ペアのサイズは、キー ペア ファイルのビット数で決まります。長いキーは、RSA セキュリティ ポリシーの強度を高めることで、より安全な実装になります。使用可能なエントリ（ビット単位）は次のとおりです。
  - 512（最低限のセキュリティ）
  - 768（通常のセキュリティ）
  - 1024（高セキュリティ、レベル 1）
  - 1536（高セキュリティ、レベル 2）
  - 2048（高セキュリティ、レベル 3）
- *filename*：生成された RSA キー ペア ファイルに割り当てる名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。キー ペアのファイル名は、ACE が識別目的でのみ使用します。

たとえば、RSA キー ペア ファイル MYRSAKEY.PEM を生成するには、次のように入力します。

```
host1/Admin# crypto generate key non-exportable 2048 MYRSAKEY.PEM
 Generating 2048 bit RSA key pair
host1/Admin#
```

RSA キー ペアを生成したら、次のタスクを実行できます。

- CSR パラメータ セットの作成：CSR パラメータ セットは、ACE が CSR 生成プロセス中に使用する distinguished name 属性を定義します。CSR コンフィギュレーション ファイル作成の詳細については、「CSR パラメータ

セットの作成および定義」の項を参照してください。

- RSA キー ペア ファイル用の CSR の生成と、署名のための CA への CSR 要求の転送：この操作では、RSA 秘密キーは ACE 内で直接作成され、外部に転送される必要がないため、セキュリティが強化されます。生成した各キー ペアには、対応する証明書が付属する必要があります。CSR の生成の詳細については、「証明書署名要求の生成」の項を参照してください。

CSR パラメータ セットの作成および定義

CSR パラメータ セットでは、ACE が CSR 生成プロセス中に CSR に適用する *distinguished name* 属性が定義されます。distinguished name 属性は、サイトの認証に必要な情報を CA に提供します。CSR パラメータ セットを定義すると、同じ distinguished name 属性を持つ複数の CSR を生成できます。

ACE の各コンテキストには、最大 8 つの CSR パラメータ セットを格納できます。

ここでは、次の内容について説明します。

- CSR パラメータ セットの作成
- 通常名の指定
- 国の指定
- 州または地域の指定
- シリアル番号の指定
- 地域の指定
- 組織名の指定
- 組織単位の指定
- 電子メール アドレスの指定

CSR パラメータ セットの作成

コンフィギュレーション モードで **crypto csr-params** コマンドを使用して、CSR パラメータ セットを作成できます。コンテキストごとに最大 8 個の CSR パラメータ セットを作成できます。

このコマンドの構文は次のとおりです。

crypto csr-params *csr_param_name*

csr_param_name 引数は CSR パラメータ セットの名前です。最大 64 文字の英数字からなる、引用符で囲まれていない、スペースを含まないテキスト文字列を入力します。

たとえば、CSR パラメータ セット CSR_PARAMS_1 を作成するには、次のように入力します。

```
host1/Admin(config)# crypto csr-params CSR_PARAMS_1
```

CSR パラメータ セットを作成したあと、CLI が CSR パラメータ コンフィギュレーション モードになり、識別名パラメータを定義できます。

```
host1/Admin(config-csr-params)#
```

識別名は、いくつかの必須パラメータとオプション パラメータで構成されます。ACE では、次の CSR パラメータ セット属性を定義する必要があります。

- Country name
- State or province
- Common name

（注） 必須の CSR パラメータ セット属性を設定しない場合、その CSR パラメータ セットを使用して CSR を生成しようとすると、ACE でエラー メッセージが表示されます。

既存の CSR パラメータ セットを削除するには、次のように入力します。

```
host1/Admin(config)# no csr-params CSR_PARAMS_1
```

既存の CSR パラメータ セットに関連する情報を表示するには、**show crypto csr-params** コマンドを使用します（第 6 章「SSL 情報および統計情報の表示」を参照）。

通常名の指定

common-name コマンドを CSR パラメータ コンフィギュレーション モードで使用して、CSR パラメータ セットの必須の通常名パラメータを定義できます。

このコマンドの構文は次のとおりです。

common-name *name*

name 引数は、SSL サイトのドメイン名または個々のホスト名である必要があります。スペースを含まず引用符で囲まれていないテキスト文字列、またはスペースを含み引用符で囲まれたテキスト文字列を、最大 64 の英数字で入力します。

たとえば、通常名 WWW.ABC123.COM を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# common-name WWW.ABC123.COM
```

CSR パラメータ セットから既存の通常名を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no common-name
```

国の指定

CSR パラメータ コンフィギュレーション モードで **country** コマンドを使用すると、CSR パラメータ セットの必須の国名パラメータを定義できます。

このコマンドの構文は次のとおりです。

country *name*

name 引数は、SSL サイトが存在する国を表す 2 文字のコード（国コードの ISO 3166 のリストを参照）です。スペースを含まず引用符で囲まれていないテキスト文字列を最大 2 文字で入力します。

たとえば、国 US （United States）を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# country US
```

CSR パラメータ セットから既存の国を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no country
```

州または地域の指定

CSR パラメータ コンフィギュレーション モードで **state** コマンドを使用すると、CSR パラメータ セットの必須の州名パラメータを定義できます。

このコマンドの構文は次のとおりです。

state *name*

name 引数は、SSL サイトが存在する州の名前です。アンパサンド（&）文字およびスペースを含め、最大 40 文字の英数字で、引用符で囲まれていないテキスト文字列を入力します。

たとえば、GA（Georgia）州を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# state GA
```

CSR パラメータ セットから既存の州を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no state
```

シリアル番号の指定

CSR パラメータ コンフィギュレーション モードで **serial-number** コマンドを使用すると、CSR パラメータ セットの必須のシリアル番号パラメータを定義できます。

（注） 入力したシリアル番号が、CA 独自のシリアル番号で上書きされることがあります。

このコマンドの構文は次のとおりです。

serial-number *number*

number 引数は、証明書に割り当てるシリアル番号です。アンパサンド（&）文字を含め最大 16 文字の英数字で、スペースが含まれず、引用符で囲まれていないテキスト文字列を入力します。

たとえば、シリアル番号 1001 を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# serial-number 1001
```

CSR パラメータ セットから既存のシリアル番号を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no serial-number
```

地域の指定

CSR パラメータ コンフィギュレーション モードで **locality** コマンドを使用すると、CSR パラメータ セットの地域パラメータ（オプション パラメータ）を定義できます。

このコマンドの構文は次のとおりです。

locality *name*

name 引数は、証明書に含める地域の名前です。スペースとアンパサンド（&）文字を含め最大 40 文字の英数字からなる文字列を、引用符で囲まずに入力します。

たとえば、地域 ATHENS を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# locality ATHENS
```

CSR パラメータ セットから既存の地域を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no locality ATHENS
```

組織名の指定

CSR パラメータ コンフィギュレーション モードで **organization-name** コマンドを使用すると、CSR パラメータ セットの組織名パラメータ（オプション パラメータ）を定義できます。

このコマンドの構文は次のとおりです。

organization-name *name*

name 引数は、証明書に含める組織名です。スペースを含め最大 64 文字の英数字からなる文字列を、引用符で囲まずに入力します。ACE はアンパサンド（&）文字もサポートします。

たとえば、組織 ABC123 SYSTEMS INC. を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# organization-name ABC123 SYSTEMS INC
```

CSR パラメータ セットから既存の組織名を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no organization-name ABC123 SYSTEMS
INC
```

組織単位の指定

CSR パラメータ コンフィギュレーション モードで **organization-unit** コマンドを使用すると、CSR パラメータ セットの組織単位パラメータ（オプション パラメータ）を定義できます。

このコマンドの構文は次のとおりです。

organization-unit *unit*

unit 引数は、組織単位の名前です。スペースを含め最大 64 文字の英数字からなる文字列を、引用符で囲まずに入力します。ACE はアンパサンド（&）文字もサポートします。

たとえば、組織単位 SSL ACCELERATOR を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# organization-unit SSL ACCELERATOR
```

CSR パラメータ セットから既存の組織単位を削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no organization-unit SSL ACCELERATOR
```

電子メール アドレスの指定

CSR パラメータ コンフィギュレーション モードで **email** コマンドを使用すると、CSR パラメータ セットの電子メール アドレス パラメータ（オプション パラメータ）を定義できます。

このコマンドの構文は次のとおりです。

email *address*

address 引数は、サイトの電子メール アドレスです。スペースを含まない最大 40 文字で、引用符で囲まれていない英数字の文字列を入力します。

たとえば、電子メール アドレス WEBADMIN @ABC123.COM を指定するには、次のように入力します。

```
host1/Admin(config-csr-params)# email WEBADMIN@ABC123.COM
```

CSR パラメータ セットから既存の電子メール アドレスを削除するには、次のように入力します。

```
host1/Admin(config-csr-params)# no email
```

証明書署名要求の生成

新しい証明書を要求する場合や、証明書を更新する場合には、証明書署名要求（CSR）ファイルを生成する必要があります。生成した CSR を CA に送信すると、CA は RSA 秘密キーを使用して CSR に署名し、CSR が証明書になります。

RSA キー ペア ファイル用の CSR ファイルを生成し、証明書要求を CA に転送するには、RSA キー ペア ファイルを含むコンテキストの EXEC コマンド モードで *crypto generate csr* コマンドを使用します。このコマンドは、PEM 形式でエンコードされた PKCS10 の CSR を生成します。

このコマンドの構文は次のとおりです。

crypto generate csr *csr_params key_filename*

引数は次のとおりです。

- *csr_params*：識別名属性が含まれている CSR パラメータ セット（「CSR パラメータ セットの作成および定義」の項を参照）。スペースを含まない最大 64 文字で、引用符で囲まれていない英数字の文字列を入力します。ACE は、CSR パラメータ セットに含まれている識別名属性を CSR に適用します。
- *key_filename*：CSR の基になるキーが含まれている RSA キー ペア ファイル名。（このキーは、ACE が CSR に埋め込む公開キーです）。スペースを含まない最大 40 文字で、引用符に囲まれていない英数字の文字列を入力します。RSA キー ペア ファイルが現在のコンテキストの ACE にロードされていることを確認します。適切なキー ペアがない場合、ACE はエラー メッセージをログします。

たとえば、CSR パラメータ セット CSR_PARAMS_1 および MYRSAKEY_1.PEM ファイル内の RSA キー ペアに基づいて CSR を生成するには、次のように入力します。

```
host1/Admin# crypto generate csr CSR_PARAMS_1 MYRSAKEY_1.PEM
-----BEGIN CERTIFICATE REQUEST-----
MIIBcDCCARoCAQAwgbQxCzAJBgNVBAYTAlVTMRIwEAYDVQQIEwlTb21lU3RhdGUx
ETAPBgNVBAcTCFNvbWVDaXR5MRcwFQYDVQQKEw5BIENvbXBhbnkgTmFtZTEbMBkG
A1UECxMSV2ViIEFkbWluaXN0cmF0aW9uMR0wGwYDVQQDExR3d3cuYWNvbXBhbnlu
YW1lLmNvbTEpMCcGCSqGSIb3DQEJARYad2ViYWRtaW5AYWNvbXBhbnluYW1lLmNv
bSAwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAtBNcNXMBqh5cJHbWFsqe9LMUO90T
pYG7gF5ODvtFGREMkHh7s6S1GF131IBWCSelG4Q/qEztjCO7y3pyjruVNQIDAQAB
oAAwDQYJKoZIhvcNAQEEBQADQQCMmXRdNPBDtMQPFvylpED5UMbeaMRm2iaC+1uZ
IaHmdoX4h5eckauu9pPgSxczau8w68PF+PDS9DAAMeRDxisL
-----END CERTIFICATE REQUEST-----
host1/Admin#
```

crypto generate csr コマンドは、PKCS10 CSR を PEM 形式で生成し、CSR を画面に出力します。主要な CA の大部分には、画面に証明書要求をカット アンド ペーストする必要がある、Web ベースのアプリケーションがあります。必要に応じて、ファイルに CSR をカット アンド ペーストできます。ACE では、CSR のコピーをローカルに保存しません。ただし、同じ CSR パラメータ セットとキー ペア ファイルを使用して、同じ要求をいつでも再生成できます。

（注） エクスポートが制限されるブラウザに対して 128 ビット暗号化を可能にするグローバル サイト証明書が必要な場合は、CA に StepUp/GSC またはチェーン証明書を申請します。証明書を受け取ったら、ACE で使用するように準備してください。詳細については、「グローバル サイト証明書の準備」のセクションを参照してください。

CA に CSR を送信すると、1 ～ 7 営業日以内に署名付き証明書を受信します。証明書を受け取ったら、目的の ACE コンテキストに証明書をインポートします（「証明書とキー ペア ファイルのインポート」の項を参照）。

グローバル サイト証明書の準備

エクスポート ブラウザは、40 ビット暗号化を使用して、SSL サーバへの接続を開始することがあります。従来のサーバ証明書では、ブラウザとサーバが SSL ハンドシェイクを完了し、40 ビット キーを使用してアプリケーション データを暗号化します。

グローバル サイト証明書は、エクスポートが制限されるブラウザに対して 128 ビットの暗号化を可能にする、拡張サーバ証明書です。サーバがグローバル証明書でブラウザに応答すると、クライアントは自動的に接続を再ネゴシエートして、128 ビットの暗号化を使用します。

CA にグローバル サイト証明書を申請した場合は、グローバル証明書と中間 CA 証明書の両方を取得します。中間 CA 証明書がグローバル証明書を検証します。次の URL から、VeriSign の中間証明書を取得できます。

http://www.verisign.com/support/install/intermediate.html

グローバル サイト証明書と中間 CA 証明書を受信したら、目的の ACE コンテキストにインポートします（「証明書とキー ペア ファイルのインポート」の項を参照）。その後、両方の証明書を含む証明書チェーン グループを作成します（「チェーン グループの作成」の項を参照）。ACE は、最初の SSL ハンドシェイク時に、クライアントにこのチェーン グループを送信します。

証明書とキー ペア ファイルのインポートまたはエクスポート

リモートのセキュア サーバから ACE に、PEM エンコードされた証明書とキー ペア ファイルをインポートできます。これらのファイルを転送するには、ACE とリモート サーバとの間のセキュアな暗号化されたトランスポート メカニズムを使用することを推奨します。

ACE はセキュア シェル（SSHv2）プロトコルをサポートします。このプロトコルは、セキュアでないネットワーク上で 2 台のホスト間にセキュアな暗号化通信を提供します。ネットワーク デバイス間でのファイル転送用に、ACE は、セキュア ファイル転送プロトコル（SFTP）、ファイル転送プロトコル（FTP）、および簡易ファイル転送プロトコル（TFTP）をサポートします。これらの 3 つのプロトコルのうち、SFTP がセキュアで暗号化された接続を提供する唯一のプロトコルであるため、SFTP を使用することを推奨します。

ACE に証明書またはキー ペア ファイルをインポートする前に、次のタスクを実行する必要があります。

- ACE で、ACE への SSH アクセスが有効になっていて、SSH クライアントからの接続を受け入れることを確認します。デフォルトでは、SSH アクセスが有効になっています。SSH アクセスを制限する場合、ACE は SSH クライアントからの接続を受け入れず、インポート コマンドが失敗します（エラー メッセージが作成されます）。

（注）　Catalyst 6500 シリーズ スイッチまたは Cisco 7600 シリーズ ルータ上でセキュア シェル デーモンを設定する方法の詳細については、『*Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide*』または『*Cisco 7600 Series Router Cisco IOS Software Configuration Guide*』を参照してください。

- SFTP サーバで、サーバが正しく設定されていることを確認します。ユーザ ディレクトリは、証明書とキー ペアが存在するディレクトリをポイントする必要があります。このパスは、ACE が、証明書とキーが SFTP サーバから、または SFTP サーバに、正常にコピーされていることを確認するために必要です。

ここでは、次の内容について説明します。

- 証明書とキー ペア ファイルのインポート
- 証明書とキー ペア ファイルのエクスポート

証明書とキー ペア ファイルのインポート

ACE は、キーで署名された PEM エンコード キー ペアおよび証明書のインポートをサポートします（ワイルドカード証明書を含む）。ACE は、公開キーのサイズとして最大で 4096 ビットを許可します。秘密キーの最大サイズは 2048 ビットです。

リモート サーバから ACE に証明書またはキー ペア ファイルをインポートするには、EXEC モードで　**crypto import** コマンドを使用します。個々の証明書およびキー、または複数の証明書とキーをインポートできます。ネットワーク デバイスは SSL ハンドシェイク時に身元を証明するため証明書と対応する公開キーをあわせて使用するため、必ず証明書ファイルと対応するキー ペア ファイルの両方をインポートしてください。

（注）　既存のローカル ファイルと同じファイル名を持つファイルをインポートしようとすると、ACE は既存のファイルを上書きしません。更新されたファイルをインポートする前に、ローカル ファイルを削除するか、またはインポートされたファイルの名前を変更します。詳細については、「証明書とキー ペア ファイルの削除」または「SSL 証明書のアップグレード」の項を参照してください。

このコマンドの構文は次のとおりです。

crypto import [**non-exportable**] {**bulk sftp** [**passphrase** *passphrase*] *ip_addr username remote_url*} | {{**ftp** | **sftp**} [**passphrase** *passphrase*] *ip_addr username remote_filename local_filename*} | {**tftp** [**passphrase** *passphrase*] *ip_addr remote_filename local_filename*} | **terminal** *local_filename* [**passphrase** *passphrase*]

キーワード、引数、およびオプションは次のとおりです。

- **non-exportable**：（任意）インポートしたファイルをエクスポート不可能としてマークします。この場合、ファイルを ACE からエクスポートできません。
- **bulk**：複数の証明書またはキー ペア ファイルを同時にインポートすることを指定します。
- **ftp**：ファイル転送プロセスとしてファイル転送プロトコル（FTP）を指定します。
- **sftp**：ファイル転送プロセスとしてセキュア ファイル転送プロトコル（SFTP）を指定します。
- **tftp**：ファイル転送プロセスとして簡易ファイル転送プロトコル（TFTP）を指定します。
- **passphrase** *passphrase*：（任意）ファイルがパス フレーズを使用して作成されたことを示します。このファイルを使用するには、ファイル転送要求とともにパス フレーズを送信する必要があります。パス フレーズには最大 40 文字を含めることができます。また、暗号化された PEM ファイルと PKCS ファイルのみが対象となります。
- *ip_addr*：リモート サーバの IP アドレスです。ドット付き 10 進表記で IP アドレスを入力します（たとえば、192.168.12.15）。
- *username*：リモート サーバへのアクセスに必要なユーザ名です。コマンドを実行すると、ACE は、リモート サーバのユーザ名のパスワードを要求します。最大 64 文字の名前を入力します。スペースや以下の特殊文字は使用しないでください。

  ;<>\|‘@$&()

- *remote_url*：**bulk** キーワードを使用してインポートするリモート サーバにある証明書またはキー ペア ファイルへのパスです。ACE は、パスで指定されたファイルのみフェッチします。再帰的にリモート ディレクトリをフェッチすることはありません。ワイルドカードを含むファイル名のパスを入力します（たとえば、/remote/path/*.pem）。ACE は、IEEE Std

1003.1-2004 の「シェルおよびユーティリティ」ボリュームのセクション 2.13 の指定に従って、POSIX パターン マッチング表記をサポートします。この表記には、「*」、「?」、および「[」のメタ文字が含まれます。

リモート ディレクトリからすべてのファイルをフェッチするには、ワイルドカード文字で終わるリモート パス（/remote/path/* など）を指定します。スペースや以下の特殊文字は使用しないでください。

;<>\|‘@$&()

ACE は、ワイルドカードの基準に一致するリモート サーバのファイルをすべてフェッチします。ただし、名前が最大で 40 文字のファイルだけをインポートします。ファイルの名前が 40 文字を超えていると、ACE はそのファイルをインポートせず、廃棄します。

- *remote_filename*：インポートするリモート サーバに存在する証明書またはキー ペア ファイルの名前です。
- *local_filename*：ファイルを ACE にインポートしたときに、そのファイルを保存するために付ける名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- **terminal**：端末画面に証明書とキー ペア情報をペーストすることによって、カット アンド ペーストを使用してファイルをインポートできます。ASCII 形式の PEM ファイルを表示するには、terminal を使用します。

ACE は、terminal を使用した最大行幅が 130 文字の PEM エンコード SSL 証明書とキーのインポートをサポートします。SSL 証明書またはキーがラップされない場合、または行ごとに 130 文字を超える場合は、ビジュアル（vi）エディタやメモ帳などのテキスト エディタを使用して、証明書またはキーを行ごとに 130 文字未満に手動でラップします。FTP、SFTP、または TFTP を使用し、行幅を無視して証明書またはキーをインポートすることもできます。これらのうちでは、安全な SFTP を使用することを推奨します。

たとえば、SFTP サーバからすべての RSA キー ファイルをバルク インポートするには、次のようにします。

```
host1/Admin# crypto import bulk sftp 1.1.1.1 JOESMITH /USR/KEYS/*.PEM
Initiating bulk import. Please wait, it might take a while...
Connecting to 1.1.1.1...
Password: password
...
Bulk import complete. Summary:
    Network errors:                          0
    Bad file URL: 0
    Specified local files already exists:    0
    Invalid file names:                      1
```

```
    Failed reading remote files:            5
    Failed reading local files:             0
    Failed writing local files:             0
    Other errors:                           0
    Successfully imported:                  10
host1/Admin#
```

（注）　**crypto import bulk** コマンドの実行中に Ctrl+C を使用してこのコマンドをキャンセルすることはできません。

バルク インポートの後、ACE は、発生したエラーおよび成功したインポートの数を示すステータス カウンタを表示します。Other errors カウンタは、他のカウンタに含まれていないエラーのバケットです。このカウンタは、次のエラーがあると増分します。

- 暗号ファイルの最大数に到達
- 認識されない形式（PEM、DER、または PKCS12）、または許容できないキー サイズ（2,048 ビット キー以下）のファイル
- その他のカウンタの 1 つで示されない、インポート時に発生した SFTP エラー

また、暗号ファイルのストレージ容量に到達した場合、ACE に次のメッセージが表示されます。

```
Warning: Crypto files' storage limit hit, further file import stopped.
```

SFTP サーバから RSA キー ファイル *MYRSAKEY.PEM* をインポートするには、次のように入力します。

```
host1/Admin# crypto import non-exportable sftp 1.1.1.1 JOESMITH
/USR/KEYS/MYRSAKEY.PEM MYKEY.PEM
Password: ********
Passive mode on.
Hash mark printing on (1024 bytes/hash mark).
#
Successfully imported file from remote server.
host1/Admin#
```

次の例は、**terminal** キーワードを使用して証明書情報を MYCERT.PEM ファイルにペーストできるようにする方法を示しています。

```
host1/Admin# crypto import terminal MYCERT.PEM
Enter PEM formatted data ending with a blank line or "quit" on a line
by itself
```

```
--------BEGIN CERTIFICATE-----------------------
MIIC1DCCAj2gAwIBAgIDCCQAMA0GCSqGSIb3DQEBAgUAMIHEMQswCQYDVQQGEwJa
QTEVMBMGA1UECBMMV2VzdGVybiBDYXBlMRIwEAYDVQQHEwlDYXBlIFRvd24xHTAb
BgNVBAoTFFRoYXd0ZSBDb25zdWx0aW5nIGNjMSgwJgYDVQQLEx9DZXJ0aWZpY2F0
aW9uIFNlcnZpY2VzIERpdmlzaW9uMRkwFwYDVQQDExBUaGF3dGUgU2VydmVyIENB
MSYwJAYJKoZIhvcNAQkBFhdzZXJ2ZXItY2VydHNAdGhhd3RlLmNvbTAeFw0wMTA3
-----------END CERTIFICATE-----------------------
quit
```

証明書とキー ペア ファイルのエクスポート

証明書またはキー ペア ファイルを ACE からリモート サーバまたは端末画面にエクスポートするには、EXEC コマンド モードで **crypto export** コマンドを使用します。

（注） 証明書またはキー ペア ファイルを ACE にインポートしたときにエクスポート不可能とマークした場合、そのファイルはエクスポートできません。

このコマンドの構文は次のとおりです。

crypto export *local_filename* {**ftp** | **sftp** | **tftp** | **terminal**} {*ip_addr*} {*username*} {*remote_filename*}

キーワード、引数、およびオプションは次のとおりです。

- *local_filename*：エクスポートする ACE にあるファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- **ftp**：ファイル転送プロセスとしてファイル転送プロトコル（FTP）を指定します。
- **sftp**：ファイル転送プロセスとしてセキュア ファイル転送プロトコル（SFTP）を指定します。FTP や TFTP より安全である SFTP を使用することを推奨します。
- **tftp**：ファイル転送プロセスとして簡易ファイル転送プロトコル（TFTP）を指定します。
- **terminal**：コピー アンド ペーストのために、端末上にファイルの内容を表示します。コンソールから証明書または秘密キーの情報をカット アンド ペーストする必要がある場合は、**terminal** キーワードを使用します。ASCII 形式の PEM ファイルを表示するには、terminal を使用します。

- *ip_addr*：リモート サーバの IP アドレスまたは名前です。ドット付き 10 進表記で IP アドレスを入力します（たとえば、192.168.12.15）。
- *username*：リモート サーバへのアクセスに必要なユーザ名です。ACE は、コマンドを実行するときにパスワードの入力を求めます。
- *remote_filename*：リモート サーバ上にファイルを保存するときに付ける名前です。

転送タイプ **ftp**、**sftp**、または **tftp** を選択した場合、**terminal** キーワードの後にリストされたリモート サーバ変数が、ACE で使用されます（これらの変数は **terminal** には使用されません）。これらの転送タイプのいずれかを選択し、リモート サーバ変数を定義しない場合、ACE は変数情報の提供を求めます。

たとえば、SFTP を使用してキーファイル MYKEY.PEM を ACE からリモート SFTP サーバにエクスポートするには、次のように入力します。

```
host1/Admin# crypto export MYKEY.PEM sftp 192.168.1.2 JOESMITH
/USR/KEYS/MYKEY.PEM
User password: ****
Writing remote file /usr/keys/mykey.pem
host1/Admin#
```

SSL 証明書のアップグレード

アクティブな SSL セッションまたは保留中の SSL セッションを中断せずに SSL 証明書をアップグレードするには、次の手順を実行します。

ステップ 1　EXEC モードで **crypto import** コマンドを使用して新しい SSL 証明書をインポートし、新しい名前で保存します。「証明書とキー ペア ファイルのインポート」の項を参照してください。たとえば、SFTP サーバから証明書をインポートするには、次のコマンドを入力します。

```
host1/Admin# crypto import non-exportable sftp 1.1.1.1 JOESMITH
/USR/CERTS/MY_CERT.PEM MY_NEW_CERT.PEM
Password: ********
Passive mode on.
Hash mark printing on (1024 bytes/hash mark).
#
Successfully imported file from remote server.
host1/Admin#
```

ステップ 2　SSL プロキシ サービスがアクティブにフローを処理している間に、SSL プロキシ コンフィギュレーション モードで **cert** コマンドを使用して、SSL プロキシ サービス内の証明書ファイルの関連付けを新しい証明書に変更します。第 3 章「SSL 終了の設定」の「証明書の指定」の項を参照してください。

たとえば、MY_NEW_CERT.PEM 証明書ファイル内の証明書を PSERVICE_SERVER SSL プロキシ サービスと関連付けるには、次のコマンドを入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# cert MY_NEW_CERT.PEM
```

キー ペアと比較した証明書の確認

デジタル証明書は、キー ペアの公開キーに基づいて構築され、1 個のキー ペアでしか使用できません。証明書ファイル内の公開キーをキー ペア ファイル内の公開キーと比較し、それらが同一であることを確認するには、EXEC コマンド モードで **verify** コマンドを使用します。

（注）　証明書内の公開キーがキー ペア ファイル内の公開キーと一致しない場合、ACE はエラー メッセージをログします。

このコマンドの構文は次のとおりです。

crypto verify *key_filename cert_filename*

引数は次のとおりです。

- *key_filename*：指定された証明書と比較して確認するのに ACE が使用する、コンテキスト キー ペア ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- *cert_filename*：指定されたキー ペアと比較して確認するのに ACE が使用する、コンテキスト証明書ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。

たとえば、MYRSAKEY.PEM ファイルと MYCERT.PEM ファイルの公開キーが一致することを確認するには、次のように入力します。

```
host1/Admin# crypto verify myrsakey.pem mycert.pem
```

```
keypair in myrsakey.pem matches certificate in mycert.pem
```

次の例は、公開キーが一致しない場合に ACE で表示される内容を示しています。

```
host1/Admin# crypto verify myrsakey_2.pem mycert.pem
Keypair in myrsakey_2.pem does not match certificate in mycert.pem
host1/Admin#
```

証明書とキー ペア ファイルの削除

EXEC コマンド モードで **crypto delete** コマンドを使用すると、有効でなくなった証明書とキー ペア ファイルを削除できます。ACE は既存の証明書またはキー ペア ファイルを上書きしないため、ファイルを削除すると、更新されたファイルをインポートできます。

このコマンドの構文は次のとおりです。

crypto delete {*filename* | **all**}

次のキーワードと引数があります。

- *filename*：削除する証明書またはキー ペア ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- **all**：すべての証明書およびキー ペア ファイルをコンテキストから削除します。プレインストールされたサンプルの証明書およびキー ファイルは削除されません。**all** キーワードを使用すると、ACE は以下のメッセージを表示して、削除確認を求めます。

  ```
  This operation will delete all crypto files for this context from
  the disk, but will not interrupt existing SSL services.  If new
  SSL files are not applied SSL services will be disabled upon next
  vip inservice or device reload.
  Do you wish to proceed? (y/n)  [n]
  ```

ACE にロードされた、使用可能な証明書およびキー ペア ファイルのリストを表示するには、**show crypto files** コマンドを使用します。

（注）

crypto delete コマンドは、指定されたコンテキスト暗号ファイルをフラッシュ メモリから削除します。ただし、既存の SSL サービスは中断されません。削除された SSL ファイルを交換しない場合、次に **vip inservice** コマンドを入力するとき、またはデバイスのリロードが発生したときに、SSL サービスは無効になります。

たとえば、キー ペア ファイル MYRSAKEY.PEM を削除するには、次のように入力します。

```
host1/Admin# crypto delete MYRASKEY.PEM
```

チェーン グループの作成

チェーン グループは、ACE がハンドシェイク時にピアに送信する*証明書チェーン*を指定します。証明書チェーンは証明書の階層型リストで、サブジェクトの証明書、ルート CA 証明書、および中間 CA 証明書を含みます。証明書の検証では、証明書チェーンで提供される情報を使用して、証明書階層リストをルート CA まで遡って信頼できる CA を検索できます。ルート CA 証明書に達する前に信頼できる CA が見つかることがあります。この場合は、そこで検索を停止します。

SSL プロキシ サービスを定義するときに、チェーン グループを使用してサービスを設定できます（第 3 章「SSL 終了の設定」の「SSL プロキシ サービスの作成および定義」の項を参照）。

ACE は、次の証明書チェーン グループ機能をサポートします。

- チェーン グループには最大で 8 個の証明書チェーンを含めることができます。
- ACE の各コンテキストには、最大で 8 個のチェーン グループを含めることができます。
- チェーン グループの最大サイズは 11 KB です。

チェーン グループを作成するには、コンフィギュレーション モードで **crypto chaingroup** コマンドを使用します。

このコマンドの構文は次のとおりです。

crypto chaingroup *group_name*

group_name 引数は、チェーン グループの名前です。スペースを含まない最大 64 文字で、引用符で囲まれていない英数字の文字列を入力します。

たとえば、チェーン グループ MYCHAINGROUP を作成するには、次のように入力します。

```
host1/Admin(config)# crypto chaingroup MYCHAINGROUP
```

チェーン グループを作成したあと、CLI は、グループに必要な証明書ファイルを追加するチェーン グループ コンフィギュレーション モードになります。

```
host1/Admin(config-chaingroup)#
```

既存のチェーン グループを削除するには、次のように入力します。

```
host1/Admin(config)# no crypto chaingroup MYCHAINGROUP
```

証明書ファイルをチェーン グループに追加するには、チェーン グループ コンフィギュレーション モードで **cert** コマンドを使用します。最大で 9 個の証明書を使用して、チェーン グループを設定できます。

このコマンドの構文は次のとおりです。

cert *cert_filename*

cert_filename 引数は、ACE に格納されている既存の証明書ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。

（注） チェーン グループ証明書を変更した場合、その変更は、**chaingroup** コマンドを使用して SSL プロキシ サービスに関連付けられたチェーン グループを再指定した後に有効になります。第 3 章「SSL 終了の設定」の「SSL プロキシ サービスの作成および定義」の項を参照してください。

通常は、証明書を確認するデバイスが正しい順序を決定するため、証明書を階層順でチェーン グループに追加する必要はありません。ただし、モバイル デバイスによっては、証明書を正しく順序付けできず、エラー メッセージを表示することがあります。その場合は、証明書を正しい順序でチェーン グループに追加する必要があります。

既存の証明書ファイルのリストを表示するには、**show crypto files** コマンドを使用します（第 6 章「SSL 情報および統計情報の表示」の「証明書情報の表示」の項を参照）。

たとえば、チェーン グループに証明書ファイル MYCERTS.PEM および MYCERTS_2.PEM を追加するには、次のように入力します。

```
host1/Admin(config-chaingroup)# cert MYCERTS.PEM
host1/Admin(config-chaingroup)# cert MYCERTS_2.PEM
```

チェーン グループから証明書ファイルを削除するには、次のように入力します。

```
host1/Admin(config-chaingroup)# no cert MYCERTS_2.PEM
```

認証のための証明書グループの設定

ACE では、認証グループを作成することで、証明書の署名者として信頼できる 10 個の SSL 証明書のグループを実装できます。認証グループを作成し証明書を割り当てたら、SSL 終了設定内のサービスに認証グループを割り当てて、クライアント認証を有効にできます。クライアント認証については、第 3 章「SSL 終了の設定」の「クライアント認証のイネーブル化」の項を参照してください。

また、SSL 開始設定のサービスにグループを割り当てて、ACE がグループ証明書でサーバ証明書を認証できるようにすることも可能です。サーバ認証については、第 4 章「SSL 開始の設定」の「サーバ認証の認証グループの設定」の項を参照してください。

認証グループを作成し、認証グループ コンフィギュレーション モードにアクセスするには、コンフィギュレーション モードで **crypto authgroup** コマンドを使用します。このコマンドの構文は次のとおりです。

crypto authgroup *group_name*

group_name 引数は、証明書認証グループの名前です。スペースを含まない最大 64 文字で、引用符で囲まれていない英数字の文字列を入力します。

たとえば、認証グループ AUTH-CERT1 を作成するには、次のように入力します。

```
host1/Admin(config)# crypto authgroup AUTH-CERT1
```

認証グループを作成したら、グループに必要な証明書ファイルを追加する認証グループ コンフィギュレーション モードにアクセスします。

```
host1/Admin(config-authgroup)#
```

既存の認証グループを削除するには、次のように入力します。

```
host1/Admin(config)# no crypto authgroup AUTH-CERT1
```

認証グループに証明書ファイルを追加するには、認証グループ コンフィギュレーション モードで **cert** コマンドを使用します。最大 10 個の証明書で認証グループを設定できます。

このコマンドの構文は次のとおりです。

cert *cert_filename*

cert_filename 引数は、ACE に格納されている既存の証明書ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。

（注） 認証グループを変更すると、その変更はただちに有効になります。

既存の証明書ファイルのリストを表示するには、**show crypto files** コマンドを使用します（第 6 章「SSL 情報および統計情報の表示」の「証明書情報の表示」の項を参照）。証明書を確認するデバイスが正しい順序を決定するため、証明書を階層順で追加する必要はありません。

たとえば、認証グループに証明書ファイル MYCERTS.PEM および MYCERTS_2.PEM を追加するには、次のように入力します。

```
host1/Admin(config-authgroup)# cert MYCERTS.PEM
host1/Admin(config-authgroup)# cert MYCERTS_2.PEM
```

認証グループから証明書ファイルを削除するには、次のように入力します。

```
host1/Admin(config-authgroup)# no cert MYCERTS_2.PEM
```