SSL 証明書とキーの設定

この章では、SSL 証明書とキーの生成とインポートの方法、および証明書とキーをファイルに関連付けて CSS で使用する方法について説明します。この章の主な内容は次のとおりです。

SSL 証明書とキーの概要

デジタル証明書とキー ペアは、ユーザ認証用のデジタル ID の一種です。証明書は、VeriSign や Thawte などの Certificate Authority（CA; 認証局）が発行します。クライアントまたはサーバの証明書には、発行元認証局の名前、デジタル署名、シリアル番号、証明書の発行対象であるクライアントまたはサーバの名前、公開キー、証明書の失効日を示すタイムスタンプが記述してあります。

CA は、自身がクライアントまたはサーバの証明書の発行元であることを証明する、信頼できる CA 証明書も提供します。この証明書によって、同じ CA が発行した Certificate Revocation List（CRL; 証明書失効リスト）の発行元も、同様に証明されます。信頼できる CA 証明書には、CA の識別名、公開キー、デジタル署名が記載されています。

CSS で証明書とキーが必要になるのは次の場合です。

- SSL 終了：サーバの証明書とキーの取得が必要。
- SSL 開始：クライアントの証明書とキーの取得が必要。
- クライアント認証：クライアント証明書と証明書失効リスト（CRL）の発行元 CA が正しいことを確認するために、その CA から信頼できる CA 証明書を取得しなければならない。

SSL 終了、クライアント認証、または SSL 開始を設定するには、その前にデジタル証明書を CSS ディスク（フラッシュ ディスクまたはハードディスク）にダウンロードする必要があります。SSL 終了または SSL 開始の場合は、公開キー / 秘密キーのペアも CSS 上に保存しなければなりません。CSS では、デジタル証明とキー ペアは、安全な領域内にある暗号化ファイルに保存されます。

サーバおよびクライアントの証明書の場合は、CA から受領したファイルを使用する方法、既存のセキュア サーバから証明書とキーをインポートする方法、CSS で独自の証明書とキーを生成する方法が使用できます。CSS では、テスト用の証明書とキーを CSS 内で作成できます。環境によっては、信頼できる認証局が発行した証明書とキーの代わりに、自分で生成した証明書を使用できる場合があります。たとえば、企業内の Web サイトで CSS と SSL を使用する環境では、信頼できる CA から受領した証明書は必ずしも不可欠ではありません。CSS 内で生成された証明書とキー ペアで、イントラネット SSL の要件を満たせることがあります。

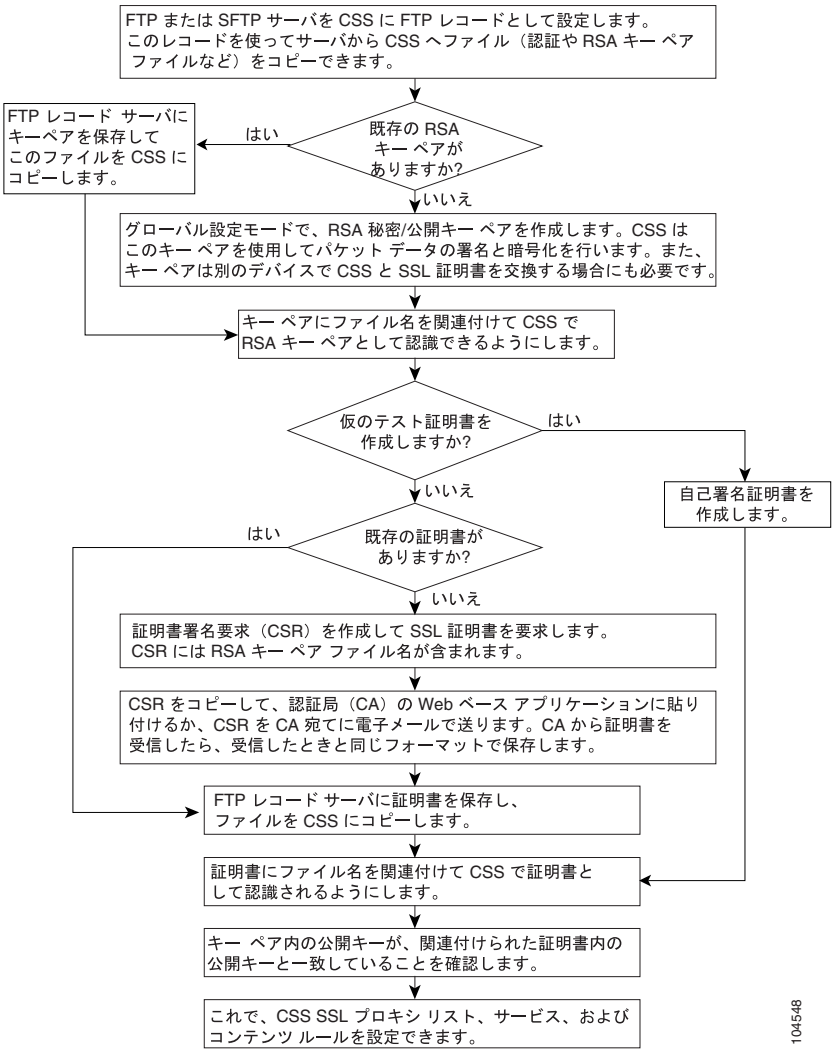
証明書またはキー ペアを CSS にインポートした後、ファイル名を付けなければなりません。このファイル名は、SSL 終了、クライアント認証、SSL 開始を設定するときに使用します。

注意　CSS で証明書とキーをインポートまたはエクスポートする場合、CSS がネットワークマウントしたリモート システム上のファイル システムからブートする（ディスクレス環境で動作する）ように設定されていないことを確認してください。CSS のネットワークマウント方式のブートは、SSL 終了ではサポートされません。つまり、証明書とキーは、CSS と SSL モジュールにローカルである必要があります。

（注）　SSL 証明書とキー ペアをインポートまたは生成する際に十分なセキュリティを確保するためにも、管理者は CSS の各種ユーザ モードを理解し、それらのユーザ モードを厳格なパスワード ポリシーで保護する必要があります。詳細については、『*Cisco Content Services Switch Command Reference*』の第 2 章「CLI Commands」の「(config) username-technician」を参照してください。

図 3-1 に、CSS での RSA キー ペアと SSL サーバ 証明書の設定方法の概要を示します。

図 3-1　SSL キーとサーバ証明書の設定の概要



CSS での証明書と秘密キーの生成

既存の証明書と秘密キーがある場合は、それらを CSS ディスクにインポートできます。既存の証明書と秘密キーのインポートの詳細については、「証明書と秘密キーのインポートまたはエクスポート」を参照してください。

CSS の既存のキー、Diffie-Hellman パラメータ、および証明書がない場合は、CSS に組み込まれている一連の証明書および秘密キー管理ユーティリティでそれらを生成できます。これらのユーティリティを使用すると、RSA 秘密キー、DSA 秘密キー、Diffie-Hellman パラメータ ファイル、Certificate Signing Request（CSR; 証明書署名要求）、および自己署名の仮証明書を簡単に生成できます。

（注）　**ssl genrsa**、**gencsr**、**gendsa**、および **gencert** のすべてのコマンドで、有効な証明書またはキー ペアを作成できます。ただし、この証明書はほとんどの Web ブラウザで、認識できない CA によって署名されたものとみなされることに留意してください。生成された証明書は仮証明書であり、有効期限は 1 年間です。**ssl gencsr** コマンドを実行すると、Privacy Enhanced Mail（PEM）形式で符号化された PKCS10 で証明書要求が生成されます。

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
- DSA キー ペアの生成
- Diffie-Hellman キー パラメータの生成
- 証明書署名要求の生成のための RSA キーの使用
- 自己署名証明書の生成

RSA キー ペアの生成

RSA キー ペアは、パケット データの署名と暗号化に使用されます。このキー ペアは、他の装置（クライアントまたはサーバ）と CSS 間で SSL 証明書を交換する前に用意する必要があります。キー ペアとは、公開キーとそれに対応する秘密（シークレット）キーのことです。生成された RSA キー ペアは CSS にファイルとして保存されます。

非対称暗号化のための RSA 秘密 / 公開キー ペアを生成するには、**ssl genrsa** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

ssl genrsa *filename numbits* **"***password***"**

変数の内容は次のとおりです。

- *filename*：生成される RSA キー ペア ファイルの名前。31 文字以内のテキスト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、CSS で識別を行うためだけに使用します。
- *numbits*：キー ペアの強度。キー ペア ファイルのビット数によって、Web トランザクションの保護に使用される RSA キー ペアのサイズが決まります。キーが長ければ RSA セキュリティ ポリシーの強度が高まり、セキュリティが強化されます。有効なエントリ（ビット数）は、512（最小強度のセキュリティ）、768（標準強度のセキュリティ）、1024（高い強度のセキュリティ）、および 2048（最大強度のセキュリティ）です。
- **"***password***"**：RSA 秘密キーをファイルとして CSS に保存する前に Data Encryption Standard（DES; データ暗号規格）を使用して RSA 秘密キーを符号化するときに使用するパスワード。ファイルを符号化すると、CSS にインポートした証明書や秘密キーへの不正なアクセスを防止できます。35 文字以内のテキスト文字列のパスワードを引用符で囲んで入力します。入力したパスワードは、CSS 実行設定に DES で符号化された文字列として示されます。

たとえば、RSA キー ペア *myrsakeyfile1* を生成するには、次のコマンドを実行します。

```
(config) # ssl genrsa myrsakeyfile1 1024 "passwd123"
 Please be patient this could take a few minutes
```

RSA キー ペアを生成したら、RSA キー ペア ファイルの証明書署名要求（CSR）ファイルを生成し、証明書要求を認証局（CA）に転送できます。これによって、CSS 上の RSA 秘密キーが直接使用され、外部から転送する必要がなくなるため、セキュリティ層が強化されます。次に、CA が証明要求に応答し、認証された証明書を返送してくるまで、内部でテスト用に使用する仮の証明書を作成できます。生成した各キー ペアを使用するには、対応する証明書が必要です。

さらに、この章の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の説明に従って、生成された RSA キー ペアに RSA キー ペア名を関連付ける必要があります。

DSA キー ペアの生成

DSA は、米国の National Institutes of Standards and Technology（NIST; 国立標準技術研究所）で開発された公開キー交換暗号化システムです。DSA は、デジタル署名だけに使用でき、秘密 / 公開キーの交換には使用できません。生成された DSA キー ペアは CSS にファイルとして保存されます。

非対称暗号化のための DSA 秘密 / 公開キー ペアを生成するには、**ssl gendsa** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

ssl gendsa *filename numbits* **“***password***”**

変数の内容は次のとおりです。

- *filename*：生成する DSA キー ペア ファイルの名前。31 文字以内のテキスト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、CSS で識別を行うためだけに使用します。
- *numbits*：キー ペアの強度。キー ペア ファイルのビット数によって、Web トランザクションの保護に使用される DSA キー ペアのサイズが決まります。キーが長ければ DSA セキュリティ ポリシーの強度が高まり、セキュリティが強化されます。有効なエントリ（ビット数）は、512（最小強度のセキュリティ）、768（標準強度のセキュリティ）、および 1024（高い強度のセキュリティ）です。
- **“***password***”**：DSA 秘密キーをファイルとして CSS に保存する前に DES を使用して DSA 秘密キーを符号化するときに使用するパスワード。ファイルを符号化すると、CSS にインポートした証明書や秘密キーへの不正なアクセスを防止できます。35 文字以内のテキスト文字列のパスワードを引用符で囲んで入力します。入力したパスワードは、CSS 実行設定に DES 符号化文字列として示されます。

たとえば、DSA キー ペア *mydsakeyfile2* を生成するには、次のコマンドを実行します。

```
(config) # ssl gendsa mydsakeyfile2 512 "passwd123"
Please be patient this could take a few minutes
```

さらに、この章の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の説明に従って、生成された DSA キー ペアに DSA キー ペア名を関連付ける必要があります。

Diffie-Hellman キー パラメータの生成

Diffie-Hellman は、共有キー合意アルゴリズムの一種です。Diffie-Hellman キー交換では、複雑なアルゴリズムと公開 / 秘密キーを使用してパケット データを暗号化し、復号化します。生成された Diffie-Hellman キー パラメータ ファイルが CSS に保存されます。**ssl gendh** コマンドを使用して、Diffie-Hellman キー合意パラメータ ファイルを生成します。

（注）　Diffie-Hellman キー合意パラメータ ファイルの生成には 20 分ほどかかる場合があり、CPU にも大きな負荷を与えます。**ssl gendh** ユーティリティの実行中は、パフォーマンスへの悪影響を防ぐためにも、CSS のトラフィック負荷が高くならないように注意してください。

このコマンドのシンタックスは次のとおりです。

ssl gendh *filename numbits* **"***password***"**

変数の内容は次のとおりです。

- *filename*：Diffie-Hellman キー パラメータを保存するファイルの名前。31 文字以内のテキスト文字列を引用符で囲まずに入力します。ファイル名は、CSS で識別を行うためだけに使用します。
- *numbits*：キーの強度。このファイルのビット数によって、Web トランザクションの保護に使用される Diffie-Hellman キーのサイズが決まります。キーが長ければ Diffie-Hellman セキュリティ ポリシーの強度が高まり、セキュリ

ティが強化されます。有効なエントリ（ビット数）は、512（最小強度のセキュリティ）、768（標準強度のセキュリティ）、1024（高い強度のセキュリティ）、および 2048（最大強度のセキュリティ）です。

- *“password”*：Diffie-Hellman キーをファイルとして CSS に保存する前に DES を使用して Diffie-Hellman キーを符号化するときに使用するパスワード。ファイルを符号化すると、CSS にインポートした証明書や秘密キーへの不正なアクセスを防止できます。35 文字以内のテキスト文字列のパスワードを引用符で囲んで入力します。入力したパスワードは、CSS 実行設定に DES 符号化文字列として示されます。

たとえば、Diffie-Hellman キー パラメータ リスト *dhparamfile2* を生成するには、次のコマンドを実行します。

```
(config) # ssl gendh dhparamfile2 512 “passwd123”
Please be patient this could take a few minutes
```

さらに、この章の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の説明に従って、生成された Diffie-Hellman パラメータ ファイルに Diffie-Hellman パラメータ ファイル名を関連付ける必要があります。

証明書署名要求の生成のための RSA キーの使用

RSA キー ペア ファイルの証明書署名要求（CSR）ファイルを生成し、その証明要求を CA に転送するには、**ssl gencsr** *rsakey* コマンドを使用します。このコマンドでは、PEM 形式で符号化された PKCS10 で CSR が生成されます。

CSR ファイルは、新しい証明書の署名を要求する場合や、証明書を更新する場合に生成する必要があります。CA が RSA 秘密キーを使用して署名すると、CSR は証明書になります。

rsakey 変数には、RSA 証明書の作成に使用されるキーを指定します。このキーは、証明書に埋め込まれている公開キーです。

CSR 生成用に RSA キーペアを使用する際は、RSA キーペア ファイルが CSS にロードされていることを確認します。生成された RSA キーペアと RSA キーペア名を関連付けます（「証明書ファイルおよび秘密キー ファイルと名前との関連付け」参照）。適切なキー ペアが存在しない場合、CSS のログにエラー メッセージが出力されます。

たとえば、RSA キー ペア *myrsakey1* に基づいて CSR を生成するには、次のコマンドを実行します。

```
CSS11503(config)# ssl gencsr myrsakey1
You are about to be asked to enter information
that will be incorporated into your certificate
request. What you are about to enter is what is
called a Distinguished Name or a DN.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [US]US
State or Province (full name) [SomeState]MA
Locality Name (city) [SomeCity]Boxborough
Organization Name (company name) [Acme Inc]Cisco Systems, Inc.
Organizational Unit Name (section) [Web Administration]Web Admin
Common Name (your domain name) [www.acme.com]www.cisco.com
Email address [webadmin@acme.com]webadmin@cisco.com

-----BEGIN CERTIFICATE REQUEST-----
MIIBWDCCAQICAQAwgZwxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJNQTETMBEGA1UE
BxMKQm94Ym9yb3VnaDEcMBoGA1UEChMTQ2lzY28gU3lzdGVtcywgSW5jLjESMBAG
A1UECxMJV2ViIEFkbWluMRYwFAYDVQQDEw13d3cuY2lzY28uY29tMSEwHwYJKoZI
hvcNAQkBFhJra3JvZWJlckBjaXNjby5jb20wXDANBgkqhkiG9w0BAQEFAANLADBI
AkEAqHXjtQUVXvmo6tAWPiMpe6oYhZbJUDgTxbW4VMCygzGZn2wUJTgLrifDB6N3
v+1tKFndE686BhKqfyOidml3wQIDAQABoAAwDQYJKoZIhvcNAQEEBQADQQA94yC3
4SUJJ4UQEnO2OqRGLOZpAElc4+IV9aTWK6NmiZsM9Gt0vPhIkLx5jjhVRLlb27Ak
H6D5omXa0SPJan5x
-----END CERTIFICATE REQUEST-----

CSS11503(config)#
```

ssl gencsr コマンドを実行すると、PEM 形式で符号化された PKCS10 で CSR が生成され、それが画面に出力されます。主な認証局の多くが、画面に表示された証明書要求をそのままカット アンド ペーストして送付できる Web アプリケーションを提供しています。画面に表示された証明書は、必要に応じてファイルにもカット アンド ペーストできます。CSR は CSS に保存されません。

（注）　輸出規制されたブラウザでの 128 ビットの暗号化を許可するグローバル サイト証明書が必要な場合は、CA に SETUP/SGC 証明書またはチェーン証明書を要求してください。証明書を受け取ったら、その証明書を CSS で使用できるように準備する必要があります。詳細については、「グローバル サイト証明書の準備」を参照してください。

CSR を認証局（CA）に送ると、署名付きの証明書が 7 日以内に届きます。CSR を受け取ったら、その CSR を CSS にインポートして関連付ける必要があります。CSR のインポート方法については、「証明書と秘密キーのインポートまたはエクスポート」を参照してください。証明書の関連付けの方法については、「証明書ファイルおよび秘密キー ファイルと名前との関連付け」を参照してください。

署名付きの証明書が届くまでの間に、CSR を作成し独自の秘密キーでその CSR に署名して仮証明書を作成し、CSR ファイルをテストすることができます。これによって有効な証明書が生成されますが、この証明書はほとんどの Web ブラウザで、認識できない CA によって署名されたものとみなされます。仮証明書の生成方法については、「自己署名証明書の生成」を参照してください。

自己署名証明書の生成

SSL テストを行う場合は、CSR を生成し、独自の秘密キーでその CSR に署名して、仮の証明書を作成できます。生成された仮証明書の有効期限は 30 日間です。仮証明書を作成し、CSS のディスクにあるファイルに保存するには、**ssl gencert** コマンドを使用します。

（注）　**ssl gencert** コマンドを実行すると、有効な証明書が作成されます。ただし、この証明書はほとんどの Web ブラウザで、認識できない CA によって署名されたものとみなされます。

証明書を作成する際は、次の点を考慮してください。

- 証明書の基になるキー ペア（RSA または DSA）
- 証明書に署名するために使用されるキー

ssl gencert コマンドを実行すると、RSA キー ペアまたは DSA キー ペアを使用して RSA 証明書や DSA 証明書に署名できます。

（注）　CSS では DSA キーによる RSA 証明書への署名（および RSA キーによる DSA 証明への署名）が可能ですが、RSA キーによる RSA 証明書への署名（および DSA キーによる DSA への署名）の方が一般的です。

このコマンドのシンタックスは次のとおりです。

ssl gencert certkey *certkey* **signkey** *signkey certfile* **"***password***"**

変数の内容は次のとおりです。

- **certkey** *certkey*：証明書の基になる RSA キー ペアまたは DSA キー ペアの名前。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- **signkey** *signkey*：証明書の署名に使用される RSA または DSA キー ペア。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- *certfile*：証明書をファイルとして CSS に保存するために使用するファイル名。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- ***"password"***：証明書をファイルとして CSS に保存する前に DES を使用して証明書ファイルを符号化するために使用するパスワード。ファイルを符号化すると、CSS にインポートした証明書や秘密キーへの不正なアクセスを防止できます。35 文字以内のテキスト文字列のパスワードを引用符で囲んで入力します。入力したパスワードは、CSS 実行設定に DES 符号化文字列として示されます。

たとえば、証明書 *mycertfile2* を対話操作で生成するには、次のコマンドを実行します。

```
CSS11503(config)# ssl gencert certkey myrsakey signkey myrsasignkey
myrsacertfile "passwd123"
You are about to be asked to enter information
that will be incorporated into your certificate
request. What you are about to enter is what is
called a Distinguished Name or a DN.
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [US]US
State or Province (full name) [SomeState]MA
Locality Name (city) [SomeCity]Boxborough
Organization Name (company name) [Acme Inc]Cisco Systems, Inc.
Organizational Unit Name (section) [Web Administration]Web Admin
Common Name (your domain name) [www.acme.com]www.cisco.com
Email address [webadmin@acme.com]webadm@cisco.com

CSS11503(config)#
```

さらに、この章の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の説明に従って、この仮証明書の内容をファイル名に関連付ける必要があります。

グローバル サイト証明書の準備

輸出が制限されていないブラウザでは、40 ビットの暗号化を使用して SSL サービスへの接続を開始できます。通常のサーバ証明書を使用すると、ブラウザとサーバは SSL ハンドシェイクを実行し、40 ビットの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、輸出規制されたブラウザでの 128 ビットの暗号化を許可する拡張されたサーバ証明書です。サーバがグローバル証明書を使用してブラウザに応答すると、クライアントでは自動的に接続を再取り決めして 128 ビットの暗号化を使用します。

CA にグローバル サイト証明書を要求した場合は、グローバル証明書とその中間 CA 証明書の両方を取得する必要があります。中間 CA 証明書で、グローバル証明書を検証します。中間証明書は、http://www.verisign.com/support/install/intermediate.html のリンクから取得できます。

中間証明書を取得したら、両方の証明書を 1 つのファイルにコピーして、チェーン証明書を作成します。CSS は、証明書チェーン全体を 1 つのファイルとして最初の SSL ハンドシェイク時にクライアントに返します。

サーバのグローバル証明書と中間証明書を FTP サーバにコピーします。CSS 用のチェーン証明書を作成する場合は、グローバル証明書と中間証明書を正しい順序でコピーする必要があります。ファイルには、最初にサーバ グローバル サイト証明書を貼り付けてから、中間証明書を貼り付けます。2 つの証明書の間には改行を 1 行挿入する必要があります。

「証明書と秘密キーのインポートまたはエクスポート」の説明に従って、ファイルを保存して CSS にインポートします。

証明書と秘密キーのインポートまたはエクスポート

既存または新しい証明書と秘密キーは、リモートのセキュア サーバに保存されている 1 つまたは複数のファイルから CSS ディスクにインポートできます。証明書の作成方法については、「CSS での証明書と秘密キーの生成」を参照してください。

これらのファイルを転送する場合は、CSS とリモート サーバ間で安全な暗号化転送メカニズを使用することをお勧めします。CSS は Secure Shell プロトコル（SSHv2）をサポートしています。SSH プロトコルを使用すれば、保護されていないネットワークを介して 2 台のホスト間で安全な暗号化通信を行うことができます。CSS は、Secure File Transfer Protocol（SFTP; セキュア ファイル転送プロトコル）およびファイル転送プロトコル（FTP）を使用してネットワーク デバイス間でのファイルの転送をサポートします。2 つのファイル転送プロトコルのうち、転送メカニズムとしては SFTP を選択することをお勧めします。SFTP は FTP と似ていますが、安全で暗号化された接続を使用します。

証明書やキーを CSS にインポートする前に、次の点に注意してください。

- CSS で、CSS への SSH アクセスを有効にして SSH クライアントからの接続を受け入れることができるようにしてください。また、証明書やキーを転送する前にセキュア管理ライセンス キーをインストールしてください。デフォルトでは、SSH アクセスは、**no restrict ssh** コマンドによってグローバルに有効化されています。SSH アクセスが制限されているか、ライセンス キーがインストールされていないと、CSS はクライアントからの SSH 接続を受け入れることができず、**copy ssl sftp** コマンドが失敗し、エラー メッセージが生成されます。

（注）　CSS での Secure Shell Daemon（SSHD）の設定の詳細については、『*Cisco Content Services Switch Security Configuration Guide*』を参照してください。

- SFTP サーバでは、ユーザ ディレクトリが証明書とキーの保存されているディレクトリを示すように、サーバが適切に設定されているか確認してください。SFTP サーバとの間で証明書とキーを適切にコピーするためにも、このパスは正しく設定する必要があります。

注意　SSH を使用する場合は、ネットワーク マウントしたリモート システムのファイル システムから CSS をブートする（ディスクレス環境）ようになっていないことを確認してください。SSH が有効で、かつ、ネットワークマウントされたファイル システムから CSS がネットワーク ブートされた場合は、SSH プロトコルの初期化が失敗して、エラー メッセージが CSS のログに出力されます。この動作は、証明書とキーのインポートやエクスポートに影響します。

証明書と秘密キーのインポート用デフォルト SFTP または FTP サーバの設定

開始する前に、**ftp-record** コマンドを使用して、インポートした証明書や秘密キーを CSS のディスクにダウンロードするために使用する SFTP サーバまたは FTP サーバを定義します。CSS からサーバにアクセスするときに使用する SFTP または FTP レコード ファイルの作成用の **ftp-record** コマンドの使用方法については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

（注）　**copy ssl** コマンドで使用する FTP レコードを定義する際に、ベース ディレクトリを使用するようであれば、そのディレクトリを SSH サーバが格納されているディレクトリからの相対ディレクトリとして指定してください。たとえば、ユーザ名が *sshlogin* で、SSH サーバが d:¥Program Files¥Network にインストールされている場合、ファイルのデフォルト ディレクトリは d:¥Program Files¥Network¥ssh になります。SFTP サーバとの間で証明書とキーを適切にコピーするためには、このパスを正しく設定する必要があります。

たとえば、*ssl_record* を定義するには、次のコマンドを入力します。

```
# ftp-record ssl_record 192.168.19.21 johndoe "abc123" /home/johndoe
```

証明書と秘密キーの CSS への転送

copy ssl コマンドを使用すると、CSS との間で証明書と秘密キーのインポートまたはエクスポートを簡単に実行できます。インポートされたファイルはすべて、CSS の安全な場所に保存されます。このコマンドは、スーパーユーザ モードでだけ実行できます。

このコマンドのシンタックスは次のとおりです。

copy ssl [*protocol*] *ftp_record* [**import** *filename* [*format*] **"***password***"** {**"***passphrase***"**}|**export** *filename2* **"***password***"**]

変数の内容は次のとおりです。

- *protocol* ：証明書や秘密キー ファイルを転送するときに使用するプロトコルのタイプ。有効なエントリは、**sftp** または **ftp** です。転送メカニズムとして、最も安全な SFTP プロトコルの使用をお勧めします。
- *ftp_record*：リモート ホスト情報を格納している作成済み FTP レコードの名前
- **import** ：ファイルをリモート サーバからインポートする。
- *filename*：サーバからインポートするファイルの名前。ファイルへの完全パスを使用します。最大で 128 文字を使用できます。
- *format* ：インポートする証明書のファイル形式。PEM 形式に変換され、DES 符号化が行われた証明書ファイルは、CSS SCM 上の特殊（かつ安全）なディレクトリに保存されます。インポート ファイルの有効な形式は次のとおりです。
  - **DER**：Distinguished Encoding Rules（DER）を使用した ASN.1 のバイナリ符号化（DER 符号化形式の X509 証明書）。たとえば、Microsoft Windows NT IIS 4.0 サーバからインポートされた証明書がこれに該当します。
  - **PEM**：Privacy Enhanced Mail（PEM）による Base64 符号化（PEM 符号化形式の X509 証明書）。たとえば、Apache/SSL UNIX サーバからインポートされた証明書がこれに該当します。
  - **PKCS12**：RSA Data Security, Inc. が策定した証明書と秘密キーの保存形式の規格。たとえば、Microsoft Windows 2000 IIS 5.0 サーバからインポートされた証明書がこれに該当します。

- *“password”*:インポートした証明書または秘密キーを DES 符号化する場合に使用するパスワード。インポートしたファイルを符号化すると、CSS 上の証明書や秘密キーへの不正なアクセスを防ぐことができます。35 文字以内のテキスト文字列のパスワードを引用符で囲んで入力します。入力したパスワードは、CSS 実行設定に DES 符号化文字列として示されます。
- *“passphrase”*：（PEM ファイルのオプション）CSS へインポートする時に証明書やキーを暗号化するためのパスフレーズ。パスフレーズ テキストを、引用符で囲んで入力します。

（注）　PKCS12 ファイル（.pfx）のパスフレーズを入力する必要があります。CSS が、パスフレーズを使用してファイルを復号化します。

- **export** ：リモート サーバにファイルをエクスポートする。
- *filename2*：サーバのファイルに付ける名前。ファイルへの完全パスを使用します。スペースを含まない 32 文字以内のテキスト文字列を引用符で囲まずに入力します。

（注）　インポートしたファイルには、証明書、RSA/DSA キー ペア、または Diffie-Hellman パラメータが含まれます。特定のコンテンツをファイル名に関連付けて、ファイルに証明書、秘密キー、Diffie-Hellman パラメータのいずれが含まれるかを示す必要があります。「証明書ファイルおよび秘密キー ファイルと名前との関連付け」を参照してください。

たとえば、証明書 *rsacert.pem* をリモート サーバから CSS にインポートするには、次のように入力します。

```
# copy ssl sftp ssl_record import rsacert.pem PEM “passwd123”
Connecting
Completed successfully
```

証明書 *rsakey.pem* をリモート サーバから CSS にインポートするには、次のように入力します。

```
# copy ssl sftp ssl_record import rsakey.pem PEM "passwd123"
Connecting
Completed successfully
```

証明書 *rsacert.pem* を CSS からリモート サーバにエクスポートするには、次のように入力します。

```
# copy ssl sftp ssl_record export rsacert.pem "passwd123"
```

copy ssl コマンドで証明書またはキーのインポートに失敗した場合は、次の点を確認してください。

- ftp レコードのユーザ アカウントとパスワードが正しいかどうか
- ベース ディレクトリが ssh または ssh/path になっているかどうか
- SSH サーバに到達可能かどうか
- ftp レコードに設定されている SSH サーバの IP アドレスが正しいかどうか

証明書ファイルおよび秘密キー ファイルと名前との関連付け

証明書とキー ペア ファイルのインポートや生成が完了すると、これらのファイルに証明書、秘密キー、Diffie-Hellman パラメータのいずれが含まれているかを CSS に示す必要があります。これは、証明書の名前、公開 / 秘密キー ペアの名前、または Diffie-Hellman パラメータの名前を特定のインポート済みファイルに関連付けることで行います。

各種の証明書と秘密キーのコマンドで指定したエントリをファイルに関連付けると、そのバインディングが CSS の実行設定に保存されます。設定変更を保存して次回のリブート時に CSS でその設定を使用できるようにするには、CSS からログアウトしたり CSS を再度ブートしたりする前に、実行設定ファイルの内容を起動設定ファイルにコピーする必要があります。CSS を再度ブートすると、証明書アソシエーションとキー アソシエーションが自動的にロードされます。

ここでは、次の内容について説明します。

証明書とファイルとの関連付け

インポートまたは生成した証明書に証明書の名前を関連付けるには、**ssl associate cert** コマンドを使用します。先頭に **no** を付けてこのコマンドを実行すると、ファイルとの関連付けが解除されます。

このコマンドのシンタックスは次のとおりです。

ssl associate cert *certname filename*

変数の内容は次のとおりです。

- *certname*：関連付ける証明書名。31 文字以内のテキスト文字列を引用符で囲まずに入力します。

- *filename*：証明書を含むファイルの名前。128 文字以内の名前を入力します。インポートまたは生成された証明書のリストを表示するには、**ssl associate cert** *certname* **?** コマンドを使用します。

たとえば、インポートされた証明書ファイル *rsacert.pem* に証明書名 *myrsacert*1 を関連付けるには、次のコマンドを入力します。

```
(config) # ssl associate cert myrsacert1 rsacert.pem
```

ファイルとの関連付けを解除するには、次のコマンドを入力します。

```
(config) # no ssl associate ssl cert myrsacert1
```

（注）　この **no** を指定したコマンドは、関連付けられた証明書がアクティブな SSL プロキシ リストで使用中になっている場合は機能しません。

RSA キー ペアとファイルとの関連付け

インポートまたは生成した RSA キー ペアに RSA キー ペア名を関連付けるには、**ssl associate rsakey** コマンドを使用します。先頭に **no** を付けてこのコマンドを実行すると、ファイルとの関連付けが解除されます。

このコマンドのシンタックスは次のとおりです。

ssl associate rsakey *keyname filename*

変数の内容は次のとおりです。

- *keyname*：関連付ける RSA キー ペア名。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- *filename*：RSA キー ペアを含むファイルの名前。128 文字以内の名前を入力します。インポートまたは生成された RSA キーのリストを表示するには、**ssl associate rsakey** *keyname* **?** コマンドを使用します。

たとえば、RSA キー名 *myrsakey1* を、インポートされた *rsakey.pem* に関連付けるには、次のコマンドを入力します。

```
(config) # ssl associate rsakey myrsakey1 rsakey.pem
```

ファイルとの関連付けを解除するには、次のコマンドを入力します。

```
(config) # no ssl associate rsakey myrsakey1
```

（注）　この **no** を指定したコマンドは、関連付けられた RSA キー ペアがアクティブな SSL プロキシ リストで使用中になっている場合は機能しません。

DSA キー ペアとファイルとの関連付け

インポートまたは生成した DSA キー ペアに DSA キー ペア名を関連付けるには、**ssl associate dsakey** コマンドを使用します。先頭に **no** を付けてこのコマンドを実行すると、ファイルとの関連付けが解除されます。

このコマンドのシンタックスは次のとおりです。

ssl associate dsakey *keyname filename*

変数の内容は次のとおりです。

- *keyname*：関連付ける DSA キー ペア名。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- *filename*：DSA キー ペアを含むファイルの名前。128 文字以内の名前を入力します。インポートまたは生成された DSA キーのリストを表示するには、**ssl associate dsakey** *keyname* **?** コマンドを使用します。

たとえば、DSA キー名 *mydsakey1* を、インポートされた *dsakey.pem* に関連付けるには、次のコマンドを入力します。

```
(config) # ssl associate dsakey mydsakey1 dsakey.pem
```

ファイルとの関連付けを解除するには、次のコマンドを入力します。

```
(config) # no ssl associate dsakey mydsakey1
```

（注）　この **no** を指定したコマンドは、関連付けられた DSA キー ペアがアクティブな SSL プロキシ リストで使用中になっている場合は機能しません。

Diffie-Hellman パラメータ名とファイルとの関連付け

インポートまたは生成した Diffie-Hellman パラメータ ファイルに Diffie-Hellman パラメータ名を関連付けるには、**ssl associate dhparam** コマンドを使用します。先頭に **no** を付けてこのコマンドを実行すると、ファイルとの関連付けが解除されます。

このコマンドのシンタックスは次のとおりです。

ssl associate dhparam *paramname filename*

変数の内容は次のとおりです。

- *paramname*：関連付ける Diffie-Hellman パラメータ名。31 文字以内のテキスト文字列を引用符で囲まずに入力します。
- *filename*：Diffie-Hellman パラメータを含むファイルの名前。128 文字以内の名前を入力します。インポートまたは生成された Diffie-Hellman ファイルのリストを表示するには、**ssl associate dhparam** *filename* **?** コマンドを使用します。

たとえば、インポートされた *dhparams.pem* に Diffie-Hellman パラメータ名 *mydhparam1* を関連付けるには、次のコマンドを入力します。

```
(config) # ssl associate dhparam mydhparam1 dhparams.pem
```

ファイルとの関連付けを解除するには、次のコマンドを入力します。

```
(config) # no ssl associate dhparam mydhparam1
```

（注）　この **no** を指定したコマンドは、関連付けられた Diffie-Hellman パラメータがアクティブな SSL プロキシ リストで使用中になっている場合は機能しません。

証明書とキー ペアの確認

デジタル証明書は、公開キーに基づいて作成され、1 つのキー ペアとだけ使用できます。証明書内の公開キーを、関連付けられた秘密キーと共に保存されている公開キーと比較し、同一であることを確認するには、**ssl verify** コマンドを使用します。関連付けられた証明書とキー ペアのリストを表示するには、**ssl verify ?** コマンドを使用します。

（注）　証明書が公開 / 秘密キー ペアと一致しない場合、エラー メッセージがログに記録されます。

このコマンドのシンタックスは次のとおりです。

ssl verify *certname keyname*

変数の内容は次のとおりです。

- *certname* : 指定したキー ペアの確認対象となる証明書に関連付けられている名前
- *keyname* : 指定した証明書の確認対象となるキー ペアに関連付けられている名前

たとえば、デジタル証明書 *myrsacert1* とキー ペア *myrsakey1* を比較して検証するには、次のコマンドを実行します。

```
(config)# ssl verify myrsacert1 myrsakey1
Certificate and key match
```

証明書と秘密キーの CSS からの削除

無効になった証明書と秘密キーを CSS から削除するには、**clear ssl file** コマンドを使用します。**clear ssl file** コマンドは、証明書や秘密キーに関連付けられているファイルには無効です。この場合は、最初に **no ssl associate** コマンドを指定して関連付けを解除します（「証明書ファイルおよび秘密キー ファイルと名前との関連付け」参照）。

このグローバル設定モードのコマンドのシンタックスは次のとおりです。

clear ssl file *filename password*

変数の内容は次のとおりです。

- *filename* : CSS から削除する証明書、キー ペア、または Diffie-Hellman パラメータ ファイルの名前
- *password* : ファイルを CSS に最初にインポート（または CSS で最初に生成）したときに、DES で符号化するために使用したパスワード。このパスワードが正確に一致しないと、ファイルは削除できません。

たとえば、*dsacert.pem* を CSS から削除するには、次のコマンドを入力します。

```
# clear ssl file dsacert.pem "passwd123"
```