Cisco Meeting Server

Cisco Meeting Server リリース 2.1

スケーラブルで復元力のあるサーバ導入向けの証明書のガイドライン

2016 年 12 月 22 日

目次

1 はじめに

Cisco Meeting Server は以前は Acano サーバと呼ばれていました。Cisco Meeting Server は、現在 Cisco UCS サーバの事前設定された新しいバージョン（Cisco Meeting Server 1000）でホストされます。Acano X-Series ハードウェアまたは仕様に基づいた VM サーバでもホスト可能です。

Cisco Meeting Server は非常に安全性が高く、サーバ上で実行されるほとんどのサービスおよびアプリケーションは、通信に TLS 暗号化プロトコルを使用します。TLS を使用することで、通信する両者は X.509 証明書と公開キーを交換して互いを認証し、暗号化アルゴリズムを交換して 2 者間で送受信するデータを暗号化することができます。

この証明書のガイドラインでは、スケーラブルで復元力のある導入向けに証明書を作成およびインストールする方法を説明します。

注：以降、本ガイドでは Cisco Meeting Server ソフトウェアを Meeting Server と表記します。

1.1 本ガイドの使用方法

この章の後半部分では、Meeting Server 導入環境に証明書を展開する際に理解しておく必要がある概念について説明します。PKI、証明書、および信頼ストアをすでに十分に理解している場合はスキップしてください。

第 2 章では、スケーラブルで復元力のあるサーバ モデル内で証明書が必要な場所、および必要な証明書のタイプについて説明します。

第 3 章では、証明書の作成方法について説明します。

第 4 章では、Meeting Server での証明書のインストールについて説明します。

第 5 章では、証明書関連の一般的な問題のトラブルシューティングに関する情報を紹介します。

第 6 章では、自己署名証明書を簡単に作成する方法について説明します。

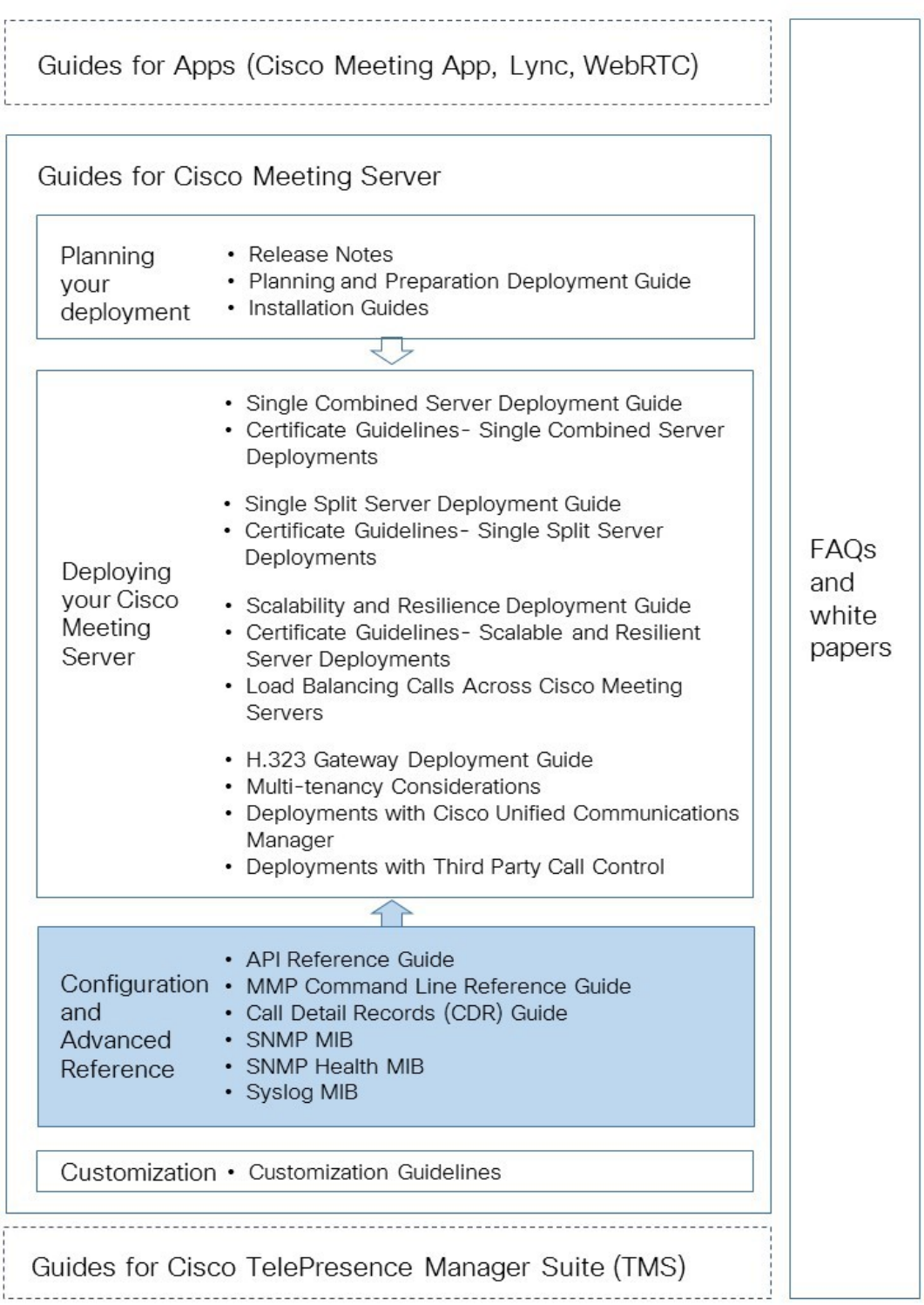
付録 A では、Meeting Server の pki コマンドではなく OpenSSL を使用する場合について説明します。

付録 B では、証明書ファイルおよび秘密キーに使用できるファイル名拡張子の概要を示します。

付録 C では、MMP pki コマンドの一覧を示します。

本ガイドは、Meeting Server のドキュメント設定（図 1）に含まれます。

図 1：Cisco Meeting Server を扱うガイドの概要



これらのマニュアルは、cisco.com から入手できます。

1.2 PKI の概要

Public Key Infrastructure（PKI）は、通信の安全を確保し、通信する両者の ID を確認するためのメカニズムを提供します。暗号化によって通信が保護され、公開/秘密キー ペアとデジタル アイデンティティ証明書を使用して ID が検証されます。

1.2.1 公開/秘密キー ペア

公開キーと秘密キーのペアは、数学的に相関付けられた、一意に関連する 2 つの暗号キーで構成されます。公開キーで暗号化されたデータは、対応する秘密キー（一般に公開しないキー）でのみ復号化できます（逆も同様です）。

1.2.2 証明書

証明書は公開キーのラッパーで、公開キーの所有者に関する情報を提供します。通常は、証明書の発行先エンティティの名前、所有者の連絡先情報、有効期間（証明書が有効な期間）、発行者（証明書を発行した機関）が含まれます。証明書は、所有者の同一性を確認できる信頼される機関によって署名される必要があります。認証局（CA）とは、個人、組織、およびネットワーク上のコンピュータの ID を証明する信頼できる機関です。

エンティティから証明書を要求されると、認証局は最初に公開/秘密キー ペアを生成します。次に、エンティティの公開キーおよび ID 情報（表 1 を参照）を含む証明書署名要求（.csr）ファイルを作成します。エンティティは自身の秘密キーを使用して .csr ファイルに署名し、処理のために CA に .csr ファイルを送信します。エンティティが Verisign などのパブリック CA に .csr ファイルを送信できるか、または Active Directory Certificate Services Role がインストールされた Active Directory サーバなどの内部 CA を使用できるかは、必要な検証のレベルによって異なります。

CA は .csr ファイルと公開キーを使用してエンティティの ID を確認します。確認に成功すると、CA は証明書に記載されたエンティティが公開キーと秘密キーのセットの所有者であることを証明するデジタル アイデンティティ証明書をエンティティに発行します。エンティティはデジタル アイデンティティ証明書を使用して、ネットワーク上の他のエンティティに対して、公開キーが秘密キーの所有者に確実に関連付けられていることを高レベルで保証します。

表 1：.csr ファイルに含まれる情報

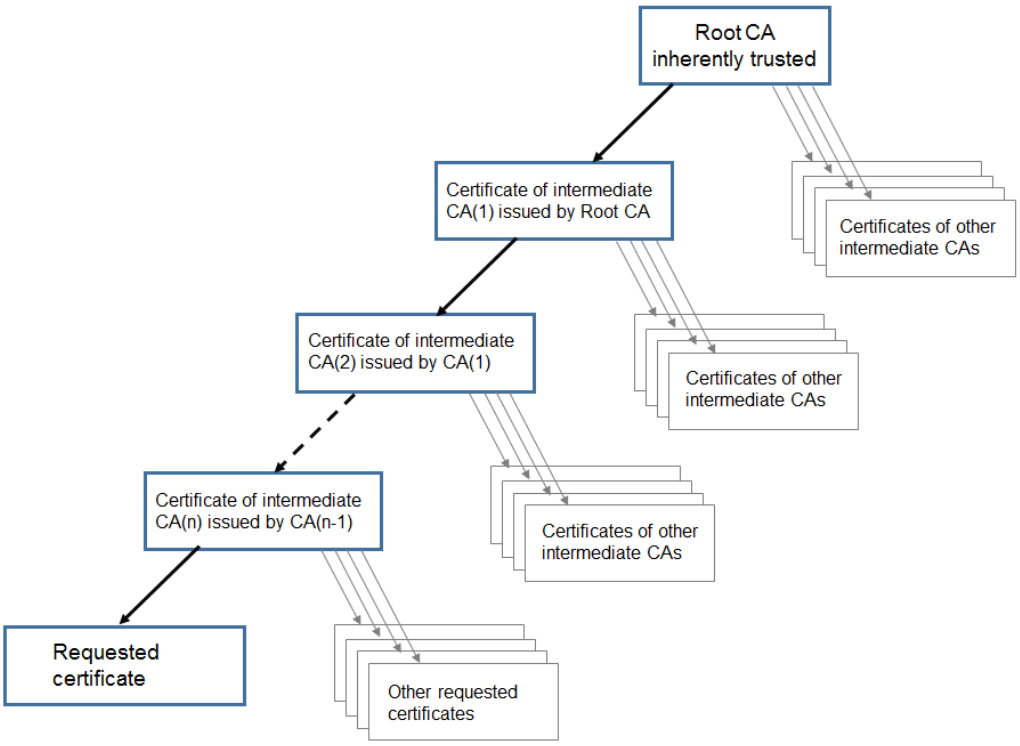
情報	説明
共通名（CN）	保護の対象となる完全修飾ドメイン名です（例：「www.example.com」）。
組織名またはビジネス名（O）	通常は会社の正式な法人名です。Ltd.、Inc.、Corp. などのサフィックスも含める必要があります。
組織部門または部署名（OU）	サポート、IT、エンジニアリング、財務など。

情報	説明
所在地（L）	市区町村。ロンドン、ボストン、ミラノ、ベルリンなど。
都道府県（ST）	バッキンガムシャー、ニュージャージーなど。短縮形を使用しないでください。
国（C）	組織の所在地の国を表す 2 文字の ISO コードです。US、GB、FR など。
電子メールアドレス	組織に連絡するための電子メールアドレスです。通常は、証明書の管理者または IT 部門の電子メールアドレスを使用します。
サブジェクト代替名（subjectAltName）	X509 バージョン 3（RFC 2459）以降、SSL 証明書では証明書に一致する名前を複数指定できるようになりました。たとえば、subjectAltName（SAN）に電子メールアドレス、IP アドレス、通常の DNS ホスト名を含めることができます。

1.2.3 信頼チェーン

別のエンティティからチャレンジを実行されたエンティティが認証用に証明書を提供する場合、自身の証明書と共に、チャレンジを実行した相手が信頼できる認証局（一般にルート認証局と呼ばれます）までの連鎖を確立する一連の他の証明書も提示する必要があります。エンティティの証明書からルート CA まで連鎖する、このような証明書の階層を「信頼チェーン」と呼びます。多くは、エンティティの証明書に順番に署名した別の認証局（中間 CA）の証明書にルート CA が署名している場合に該当します。この場合、エンティティは自身の証明書と、ルート CA が発行した中間 CA の証明書の両方を提示する必要があります。エンティティが自身の証明書のみを提示して、信頼できるルート CA への連鎖が確立されなかった場合、チャレンジを実行した相手は提示された証明書を信頼しません。エンティティの証明書からルート CA まで連鎖する一連の証明書は、中間 CA に対して発行されているため、「中間証明書」と呼ばれます。

図 2：信頼証明書チェーン



接続デバイスが信頼チェーンを確認できるようにするには、すべての証明書に「発行先（Issued To）」と「発行者（Issued By）」の 2 つのフィールドが必要です。中間 CA は信頼を確立するために、信頼チェーンの確認を続けている接続デバイスに対して、必要に応じてこれら 2 つのフィールドにさまざまな情報を示します。ルート CA 証明書は「発行先（Issued To）」と「発行者（Issued By）」そのものであるため、それ以上確認することはできません。

たとえば、エンティティ B（Web クライアント）がエンティティ A（Web サーバ：www.example.com）に認証チャレンジを実行した場合、エンティティ A はエンティティ B に証明書と証明書チェーンを提示する必要があります。

図 3：エンティティ A の証明書チェーン

証明書 1 - 発行先（Issued To）：example.com、発行者（Issued By）：中間 CA 1
証明書 2 - 発行先（Issued To）：中間 CA 1、発行者（Issued By）：中間 CA 2
証明書 3 - 発行先（Issued To）：中間 CA 2、発行者（Issued By）：ルート CA

エンティティ B が自身の信頼ストアにルート CA の証明書を保存すると、エンティティ A とエンティティ B の間にセキュアな接続が確立されます。エンティティ B はエンティティ A の公開キーを使用してメッセージを暗号化し、エンティティ A に送信できます。秘密キーにはエンティティ A のみがアクセスできるため、エンティティ A のみがこのメッセージを復号化できます。

注：このプロセスは「証明書チェーン」と呼ばれ、中間 CA 証明書は「チェーン証明書」と呼ばれる場合があります。

1.2.4 証明書バンドル

証明書バンドルとは、ルート CA の証明書およびチェーン内のすべての中間証明書のコピーを保持する単一のファイル（.pem、.cer、または.crt 拡張子）です。証明書は、証明書バンドル内の最後がルート CA の証明書になる順番で並べる必要があります。外部クライアント（たとえば Web ブラウザと XMPP クライアント）でセキュアな接続を設定するには、Web Bridge と XMPP サーバがそれぞれ証明書および証明書バンドルを提示する必要があります。Call Bridge が SIP ピアへの TLS トランクを確立する場合、Call Bridge は証明書と証明書バンドルを SIP エンドポイントに提示する必要があります。

証明書バンドルは、メモ帳などのプレーン テキスト エディタを使用して作成できます。-----BEGIN CERTIFICATE----- タグと -----END CERTIFICATE----- タグの間の文字をすべてドキュメントに挿入する必要があります。証明書と証明書の間にスペースは挿入しません。たとえば、証明書 1 の -----END CERTIFICATE----- と証明書 2 の -----BEGIN CERTIFICATE----- の間にスペースまたは余分な行がないようにします。ファイルの最後に 1 行追加する必要があります。.pem、.cer、または .crt 拡張子でファイルを保存します。

1.2.5 信頼ストア

Web ブラウザなどのクライアントは、信頼できる署名機関、つまり「信頼チェーン」によって信頼できるサーバのリストを保持しています。これらの信頼できる CA はクライアントの「信頼ストア」内に保持されます。信頼できる CA が失効リストを発行すると、クライアントは信頼ストアを更新して、失効リストに含まれるエンティティをストアから削除します。

接続クライアント（またはデバイス）は証明書を信頼するために、その証明書の CA が自身の信頼ストアに保持されているかどうかを確認します。信頼できる CA が発行した証明書ではない場合、接続クライアントは発行元の CA の証明書が信頼できる CA によって発行されたものであるかどうかを確認します。信頼できる CA が確認されるか、信頼できる CA を検出できなくなるまでチェーンに対してこの処理が繰り返されます。信頼できる CA が確認されると、クライアントとサーバの間にセキュアな接続が確立されます。信頼できる CA を検出できなかった場合、通常は接続クライアントにエラー メッセージが表示されます。

2 導入に必要な証明書

この章では、スケーラブルで復元力のあるサーバ導入環境でセキュアな接続を確立するために証明書が必要な場所、および必要な証明書のタイプについて説明します。

注：ローカル ファイアウォールの SIP および Lync コール トラバーサルは、Meeting Server 2.0 の時点でもベータ機能となります。実稼働環境では使用しないでください。この機能を評価する場合、Call Bridge と SIP Edge の間に信頼を設定する必要があることに注意してください。詳細については、『Scalable and Resilient Server Deployment Guide』を参照してください。

2.1 スケーラブルで復元力のあるサーバ導入とは

スケーラブルで復元力のあるサーバ導入環境では、次のアプリケーションが単一のサーバ上に存在するか、Core サーバと Edge サーバに分かれて配置されます。

- Call Bridge
- TURN サーバ
- XMPP サーバ
- Web Bridge
- データベース
- ロード バランサ（分割導入のみ）

配置状況は導入の要件によって異なります。

Web 管理インターフェイスは、Call Bridge に対して Web インターフェイスとして機能します。

複数の Meeting Server を使った大規模な導入環境では、すべてのサーバですべてのアプリケーションを有効にする必要はありません。有効にするアプリケーションを決定する際の推奨事項については、『Scalable and Resilient Server Deployment Guide』を参照してください。

スケーラブルで復元力のある導入を単一の複合導入または単一の分割導入と比較した場合、証明書要件の主な違いは、クラスタ化されたデータベースの証明書が必要な点です。詳細は、3.1.4 項に記載されています。

注：TURN サーバに TLS を設定する場合、TURN サーバには WebRTC クライアントによって信頼される証明書/キー ペアが必要になります。

2.1.1 その他のコンポーネント

上記に記載されているコンポーネントに加えて、Meeting Server で次のコンポーネントも有効にすることができます。

- H.323 ゲートウェイ
- レコーダ
- Streamer

H.323 ゲートウェイは、Call Bridge をホストしている同じ Meeting Server に配置できます。ただし、レコーダーと Streamer は Call Bridge のホスト サーバとは別の Meeting Server インスタンスでホストする必要があります。レコーダーまたは Streamer が Call Bridge（ローカル）と同じサーバにホストされている場合は、テスト目的、またはごく小規模な導入環境でのみ使用してください。

注：本ガイドでは、H.323 ゲートウェイ コンポーネントの証明書情報は扱いません。これらの詳細については、『Cisco Meeting Server H.323 Gateway Deployment Guide』を参照してください。

注：Meeting Server でレコーダーまたは Streamer を有効にする場合は、ローカル CA の署名付き証明書ファイル、キー ファイル、および証明書バンドルをサーバにアップロードし、Call Bridge 証明書をレコーダーまたは Streamer の信頼ストアに追加する必要があります。詳細は、導入ガイドの「Recording Meetings」または「Streaming Meetings」の章にそれぞれ記載されています。

2.2 パブリック CA または内部 CA の署名付き証明書

Meeting Server 上のアプリケーションが外部デバイスに接続する場合、外部デバイスに信頼される必要があるため、パブリック CA によって署名された証明書が必要です。Meeting Server の内部で相互作用するアプリケーションは、内部 CA によって署名された証明書のみを必要とします。内部 CA 署名付き証明書は、Active Directory Certificate Services Role がインストールされている Active Directory サーバなどの組織の認証局（3.3 項を参照）、またはローカル認証局で生成できます。

パブリック CA 署名付き証明書が必要なアプリケーションを表 2 に示します。内部 CA 署名付き証明書のみを必要とするアプリケーションを表 3 に示します。

Meeting Server でのワイルドカード証明書の使用、および他の証明書関連の FAQ については、こちらのリンクをご覧ください。

表 2：パブリック CA 署名付き証明書（スケーラブルで復元力のあるサーバ モデル）

パブリック CA 署名付き証明書を必要とするアプリケーション	理由
Web Bridge（WebRTC クライアントを使用する場合のみ）	WebRTC クライアントが接続を信頼するには、Web Bridge のパブリック CA 署名付き証明書が必要です。
XMPP サーバ（ネイティブ シスコ ミーティング アプリケーションを使用する場合のみ）	ネイティブ シスコ ミーティング アプリケーションが接続を信頼するには、XMPP サーバのパブリック CA 署名付き証明書が必要です。
Call Bridge（Lync とのダイレクト フェデレーション用に Meeting Server がパブリック ネットワークに接続されている場合のみ）	Lync Edge サーバでダイレクト フェデレーションを実行するには、Call Bridge のパブリック CA 署名付き証明書が必要です。
TURN サーバ	TURN サーバに TLS を設定する場合、WebRTC クライアントが接続を信頼できるように、TURN サーバには Web Bridge 用に作成されるような証明書/キー ペアが必要です。証明書は、Web Bridge 証明書に使用した同じ認証局によって署名される必要があります。

表 3：内部 CA 署名付き証明書（スケーラブルで復元力のあるサーバ モデル）

内部 CA 署名付き証明書を使用できるアプリケーション	理由
Web 管理	Meeting Server では HTTPS 接続のみが可能なため、Web 管理の証明書が必要です。 注：Meeting Server API は Web 管理インターフェイス経由でルーティングされます。したがって、Web 管理ではなく API を使用して Call Bridge を設定した場合も証明書が必要です。
Call Bridge	Web Bridge は Call Bridge の証明書を要求し、これを信頼する必要があります。 Active Directory サーバも Call Bridge の証明書を信頼する必要があります。 また、導入環境に TLS を使用する SIP トランクが存在する場合は、Call Bridge が SIP コール制御デバイスと相互認証する際に証明書が必要になります。
ロード バランサ（分割サーバ導入でネイティブ アプリを使用する場合のみ）	ネイティブ シスコ ミーティング アプリケーションを導入している場合、Core サーバと Edge サーバ間のトランクが、ロード バランサによって提示される証明書を認証（信頼）する必要があります。 注：外部のネイティブ シスコ ミーティング アプリケーションはロード バランサ証明書を確認しません。
トランク（分割サーバ導入でネイティブ アプリを使用する場合のみ）	ネイティブ シスコ ミーティング アプリケーションを導入している場合、Edge サーバのロード バランサが、トランクによって提示される証明書を認証（信頼）する必要があります。

内部 CA 署名付き証明書を使用できるアプリケーション	理由
クラスタリング データベース	データベース クラスタリングでは、機密性と認証の両方の目的で公開/秘密キー暗号化が使用されます。データベースをホストする各サーバには、同じ CA によって署名された一連の証明書が必要です。3.1.4 項を参照してください。
レコーダ	Meeting Server でレコーダーを有効にした場合、Call Bridge はレコーダーの署名付き証明書を要求します。さらにレコーダーは Call Bridge の証明書を要求し、これを信頼する必要があります。
Streamer	Meeting Server で Streamer を有効にした場合、Call Bridge は Streamer の署名付き証明書を要求します。さらに Streamer は Call Bridge の証明書を要求し、これを信頼する必要があります。

3 証明書の取得

第 2 章では、導入環境でセキュアな接続を確立するために証明書が必要な場所、および必要な証明書のタイプ（パブリック CA または内部 CA の署名付き）について説明します。この章では、さまざまなタイプの証明書を取得する方法を中心に説明し、インストール先については第 4 章で取り上げます。

注：Lync 導入環境を Meeting Server に接続する場合は、Lync フロントエンド サーバが信頼する同じ認証局（CA）を使用することを推奨します。CA の詳細、および Meeting Server と Lync の統合のサポートについては、Lync アドバイザにお問い合わせください。

すべての証明書で、次の 3 つの手順を実行する必要があります。

1. 特定の Meeting Server コンポーネントの秘密キーと証明書署名要求（.csr）ファイルを生成します。

   注：公開キーが作成され、.csr ファイル内に保持されます。

2. .csr ファイルを署名のために CA（パブリック CA または内部 CA）に提出します。
3. SFTP を使用して、署名付き証明書および中間 CA バンドル（存在する場合）を Meeting Server にインストールします。

この章の後半で手順 1 および 2 の例を示します。手順 3 については第 4 章で取り上げます。

注：Meeting Server の MMP コマンドを使用して自己署名証明書を生成する手順は、第 6 章で説明します。自己署名証明書はラボで構成をテストする場合に役立ちます。ただし、実稼働環境では、認証局（CA）により署名された証明書を使用することを推奨します。

注：Meeting Server は SHA1 および SHA2 アルゴリズムを使用して署名された証明書をサポートしています。Meeting Server で証明書署名要求を作成すると、CA が現在準拠しているルールに従い、署名には SHA256 が使用されます。

3.1 秘密キーと証明書署名要求（.csr ファイル）の生成

ここでは、Meeting Server の MMP **`pki`** コマンドを使用した公開キーと .csr ファイルの作成について説明します。サードパーティ製ツールを使用する場合は、サードパーティが指定する手順に従って公開キーとファイルを作成してください。その後は、本ガイドの 3.2 項の手順を実行します。OpenSSL を使用して秘密キーと .csr ファイルを作成する場合は、実行する手順の概要が付録 A に記載されています。

`pki csr <key/cert basename>` コマンドを使用して、秘密キー <basename>.key と証明書署名要求 <basename>.csr の 2 つのファイルを生成できます。これらのファイルは、Meeting Server で SFTP を使用してすぐに取得できます。

注：basename に「.」を含めることはできません。たとえば、**`pki csr basename`** は有効ですが、**`pki csr base.name`** は使用できません。

秘密キーと証明書署名要求ファイルを生成する手順は次のとおりです。

1. MMP にログインします。
2. 次の構文を使用して **pki csr** コマンドを入力します。

   pki csr <key/cert basename> <CN:value> [OU:<value>] [O:<value>] [ST:<value>] [C:<value>] [subjectAltName:<value>]

   値は次のとおりです。

   <key/cert basename> は新しいキーと CSR を識別する文字列です。英数字、ハイフン、または下線を使用できます。

   CN、**OU**、**O**、**ST**、**C**、**subjectAltName** については、表 4 で説明します。pki csr コマンドを使用してローカル CA の署名を求める証明書要求ファイルを作成する場合は、これらの任意の項目を省略することができます。パブリック認証局の署名を求める証明書要求ファイルを作成する場合は、すべての属性を指定することを推奨します。

表 4：.csr ファイル内の属性

属性		説明	任意/必須
CN	共通名	ドメイン ネーム システム（DNS）内のサーバの正確な場所を指定する完全修飾ドメイン名（FQDN）です。たとえば、ホスト名が webBridge1 で親ドメインが example.com のコンポーネントの場合、完全修飾ドメイン名は webBridge1.example.com です。FQDN によって、このコンポーネントは他のドメインの webBridge1 という他のコンポーネントと一意に区別されます。	必須（下記の注を参照）
O	組織名またはビジネス名	通常は会社の正式な法人名です。Ltd.、Inc.、Corp. などのサフィックスも含める必要があります。複数の単語を含む属性は“”で囲んでください（例：“Example Inc.”）。	任意
OU	組織部門または部署名	サポート、IT、エンジニアリング、財務など。複数の単語を含む属性は“”で囲んでください（例：“Human Resources”）。	任意
L	所在地	市区町村。ロンドン、ボストン、ミラノ、ベルリンなど。	任意
ST	都道府県	バッキンガムシャー、カリフォルニアなど。短縮形を使用しないでください。複数の単語を含む属性は“”で囲んでください（例：“New Jersey”）。	任意
C	国	組織の所在地の国を表す 2 文字の ISO コードです。US、GB、FR など。	任意
電子メールアドレス		組織に連絡するための電子メールアドレスです。通常は、証明書の管理者または IT 部門の電子メールアドレスを使用します。	任意
SAN	サブジェクト代替名	X509 バージョン 3（RFC 2459）以降、SSL 証明書では証明書に一致する名前を複数指定できるようになりました。このフィールドを使うと、生成された証明書で複数のドメインをカバーできます。カンマで区切った IP アドレス、ドメイン名、電子メールアドレス、通常の DNS ホスト名などを指定できます。このリストを指定する場合は、CN もリストに含める必要があります。このフィールドは省略できますが、XMPP クライアントが証明書を受け入れるようにするには SAN フィールドを入力してください。省略した場合は XMPP クライアントに証明書エラーが表示されます。	XMPP サーバの証明書である場合、または複数のコンポーネントで 1 つの証明書を使用する場合は必須です。下記の注意を参照してください。

注意点：

- Web Bridge で専用証明書を使用する場合、CN フィールドには Web Bridge の DNS A レコードで定義されている FQDN を指定します。FQDN を指定しないと、ブラウザ証明書のエラーが発生する可能性があります。
- XMPP サーバで専用証明書を使用する場合、CN フィールドには XMPP サーバの DNS SRV レコードで定義されている FQDN を指定します。**subjectAltName** フィールドでは、XMPP サーバのドメイン名と XMPP サーバの DNS SRV レコードを指定します。
- Web Bridge、XMPP サーバ、Call Bridge、TURN サーバなどの複数のコンポーネントで同じ証明書を使用する場合は、CN フィールドにドメイン名（DN）を指定し、SAN フィールドにドメイン名（DN）、および証明書を使用する各コンポーネントの FQDN を指定します。
- SAN フィールドでは、リストの「,」デリミタと項目の間にスペースを挿入しないでください。

次に例を示します。

CN=**example.com**

SAN=

callbridge1.example.com,callbridge2.example.com,callbridge3.example.com,xmppserver.example.com,webbridge.example.com,example.com

pki csr コマンドを使用する場合の構文は次のとおりです。

pki csr <key/cert basename> <CN:value> [OU:<value>] [O:<value>] [ST:<value>] [C:<value>] [<subjectAltName:value>]

コマンドの例を示します。

pki csr one-cert CN:example.com subjectAltName:callbridge1.example.com,callbridge2.example.com,callbridge3.example.com,xmppserver.example.com,webbridge.example.com

注：**pki** コマンドを使用する場合は、SAN リストに CN が自動的に追加されるため、上記の例に示すように SAN リストに CN を含めないでください。

3.1.1 Web Bridge 用の CSR

Web ブラウザは、**CN** フィールドを確認して Web Bridge の FQDN を特定します。Web ブラウザの証明書エラーを回避するには、次の推奨事項に従ってください。

- Web Bridge で専用証明書を使用する場合：**CN** フィールドには Web Bridge の DNS A レコードで定義されている FQDN を指定します。FQDN を指定しないと、ブラウザ証明書のエラーが発生する可能性があります。**subjectAltName** フィールドを使用する場合は、**CN** フィールドで指定した FQDN を **subjectAltName** フィールドに含める必要があります（自動的に追加されない場合）。注：**SAN** リストが存在する場合、**pki csr** によって **SAN** リストに **CN** が自動的に追加されます。

- 複数のコンポーネント（Web Bridge、XMPP サーバ、Call Bridge、TURN サーバ）で同じ証明書を使用する場合：**CN** フィールドにドメイン名（DN）を指定し、**SAN** フィールドにドメイン名（DN）、および証明書を使用する各コンポーネントの FQDN を指定します。

次に例を示します。

```
pki csr webbridge CN:www.example.com O:"Example Inc."
```

または

```
pki csr webbridge CN:www.example.com O:"Example Inc."
subjectAltName:guest.example.com
```

この例では、webbridge.key、webbridge.csr という 2 つのファイルが生成されます。ファイルは Meeting Server で SFTP を使用してすぐに取得できます。.csr ファイルを署名のためにパブリック CA に提出します。3.2 項を参照してください。

4.6 項で Web Bridge の証明書のアップロードについて説明します。

3.1.2 XMPP サーバ用の CSR

ネイティブ シスコ ミーティング アプリケーションは、**subjectAltName** フィールドを確認して XMPP サーバのドメインを特定します。クライアント証明書エラーを回避するには、XMPP サーバの CSR で以下のように指定してください。

- **CN** フィールドと **subjectAltName** フィールドに XMPP サーバの DNS レコードを指定する。
- **subjectAltName** フィールドに XMPP サーバのドメイン名を指定する。

たとえば、XMPP ドメインを example.com に設定し、DNS が xmpp.example.com である場合、CN に xmpp.example.com を指定して SAN リストに xmpp.example.com と example.com を追加する必要があります。注：**SAN** リストが存在する場合は、**pki csr** によって **SAN** リストに **CN** が自動的に追加されます。

```
pki csr xmppserver CN:xmpp.example.com O:"Example Inc."
subjectAltName:example.com
```

ワイルドカードを使用する場合は次のとおりです。

```
pki csr xmppserver CN:*.example.com O:"Example Inc."
subjectAltName:example.com
```

ドメイン example.com の XMPP サーバ用に、xmppserver.key、xmppserver.csr という 2 つのファイルが生成されます。.csr ファイルを署名のためにパブリック CA に提出します。3.2 項を参照してください。ファイルは Meeting Server で SFTP を使用してすぐに取得できます。

4.4 項で XMPP サーバの証明書のアップロードについて説明します。

3.1.3 トランク/ロード バランサ ペア用の CSR

複数の Edge サーバを使ったスケーラブルな導入環境では、XMPP サービスをホストする Core サーバが Edge サーバ上の各ロード バランサへのトランクを必要とします。トランクとロード バランサごとに秘密キー/証明書ペアを作成する必要があります。トランクとロード バランサは同じ秘密キー/証明書ペアを使用できませんが、それぞれの証明書は内部 CA で署名できます。

1. Edge サーバで有効になっているロード バランサごとに秘密キー/証明書ペアを作成します。

   次に例を示します。

   pki csr edge1 CN:www.example.com

   edge1.key と edge1.csr が作成されます。

2. XMPP サービスが有効になっている Core サーバごとに秘密キー/証明書ペアを作成します。

   次に例を示します。

   pki csr core1 CN:www.example.com

   core1.key と core1.csr が作成されます。

3. 内部 CA を使用して edge1.csr および core1.csr 証明書署名要求ファイルに署名し、これらに対応する edge1.crt および core1.crt 証明書と内部 CA 証明書（バンドル）を取得します。3.3 項を参照してください。

4.5 項でトランクとロード バランサの証明書のアップロードについて説明します。

3.1.4 データベース クラスタリング用の CSR

データベース クラスタリング用の手順は次のとおりです。

1. データベース サーバの秘密キーと証明書要求ファイルを作成します。データベース クラスタ内のすべてのサーバで同じ証明書を使用できます。この場合は、CN フィールドにいずれかのサーバの FQDN を指定し、SAN フィールドに他のサーバの FQDN を指定します。「キーの拡張用途」を使用する場合は、「サーバ認証」がデータベース サーバに対して許可されていることを確認してください。

   次に例を示します。

   pki csr dbcluster_server CN:www.example.com

   dbcluster_server.csr という名前の CSR ファイルと dbcluster_ server.key という名前の秘密キーが生成されます。

2. データベース クライアントの秘密キーと証明書要求ファイルを作成します。データベース クライアントの CommonName（CN）は「postgres」である必要があります。「キーの拡張用途」を使用する場合は、「クライアント認証」がデータベース クライアントに対して許可されていることを確認してください。

   次に例を示します。

   pki csr dbcluster_client CN:postgres

   dbcluster_client.csr という名前の CSR ファイルと dbcluster_client.key という名前の秘密キーが生成されます。

3. 内部 CA を使用して dbcluster_server.csr および dbcluster_client.csr 証明書署名要求ファイルに署名し、これらに対応する dbcluster_server.crt および dbcluster_ client.crt 証明書と内部 CA 証明書（バンドル）を取得します。3.3 項を参照してください。

4.8 項でデータベース クラスタリングの証明書のアップロードについて説明します。

3.1.5 TURN サーバ用の CSR

TURN サーバで TLS を使用する場合、WebRTC クライアントが接続を信頼できるように、TURN サーバには Web Bridge 用に作成されるような証明書/キー ペアが必要です。証明書は、Web Bridge 証明書に使用した同じ認証局によって署名される必要があります。

次に例を示します。

```
pki csr turnserver CN:www.example.com O:"Example Inc."
```

この例では、turn.key、turn.csr という 2 つのファイルが生成されます。ファイルは Meeting Server で SFTP を使用してすぐに取得できます。.csr ファイルを署名のためにパブリック CA に提出します。3.2 項を参照してください。

4.9 項で TURN サーバの証明書のアップロードについて説明します。

3.2 パブリック認証局を使用した CSR の署名

Meeting Server に必要なパブリック CA 署名付き証明書の一覧については、2.2 項を参照してください。

パブリック CA 署名付き証明書を取得するには、生成された .csr ファイルを希望する認証局（Verisign など）に送信します。CA は ID を確認して署名付き証明書を発行します。証明書ファイルの拡張子は、.pem、.cer、または .crt です。付録 B では、証明書ファイルに使用されるファイル拡張子の概要を示します。

署名付き証明書と秘密キーを Meeting Server に転送する前に、証明書ファイルを確認してください。CA によって証明書チェーンが発行された場合は、チェーンから証明書を抽出する必要があります。証明書ファイルを開き、特定の証明書の BEGIN CERTIFICATE および END CERTIFICATE 行を含むテキストをコピーして、テキスト ファイルに貼り付けます。このファイルを .crt、.cer、または .pem 拡張子で証明書として保存します。残りの証明書チェーンをコピーして別のファイルに貼り付けます。中間証明書チェーンであることがわかる明確な名前を付けて、同じ拡張子（.crt、.cer、または .pem）を使用してください。中間証明書チェーンは、チェーンを発行した CA の証明書が最初でルート CA の証明書がチェーンの最後になる順番で並べる必要があります。

Meeting Server での署名付き証明書と秘密キーのインストールについては、第 4 章を参照してください。

3.3 内部認証局を使用した CSR の署名

Meeting Server に必要な内部 CA 署名付き証明書の一覧については、2.2 項を参照してください。

この項の内容は、内部 CA として AD を使用する場合を対象としています。別の内部 CA を使用する場合は、該当する手順を実行した後、本ガイドの第 4 章を参照してください。

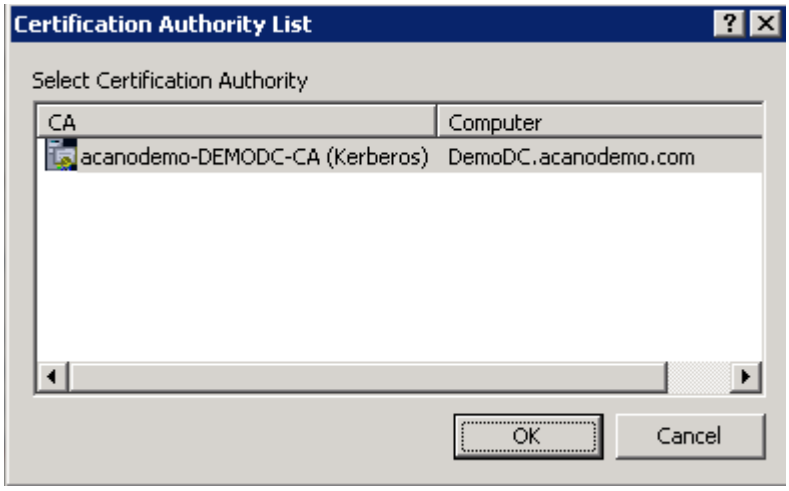
内部 CA 署名付き証明書を取得するには、次の手順を実行します。

1. Active Directory Certificate Services Role がインストールされている Active Directory サーバなどの CA に、生成された .csr ファイルを転送します。
2. CA サーバ上のコマンド ライン管理シェルで次のコマンドを発行します。パスと CSR ファイル名は実際の情報に置き換えてください。

   `certreq -submit -attrib "CertificateTemplate:WebServer" <path\csr_filename>`

   次に例を示します。

   `certreq -submit -attrib "CertificateTemplate:WebServer" C:\Users\Administrator\Desktop\example.csr`
3. このコマンドを入力すると、次のような CA 選択リストが表示されます。適切な CA を選択し、[OK] をクリックします。



証明書を発行する権限がある Windows アカウントを使用している場合は、生成された証明書の保存を求めるプロンプトが表示されます。ファイルを .crt、.cer、または .pem 拡張子で保存します（例：example.crt）。手順 4 に進みます。証明書ファイルの拡張子の概要については、付録 B を参照してください。

生成された証明書を発行するためのプロンプトが表示されずに、「証明書の要求は保留中です：提出済みです（Certificate request is pending: taken under submission）」というメッセージと要求 ID のリストがコマンド プロンプト ウィンドウに表示された場合は、要求 ID を書き留めてください。

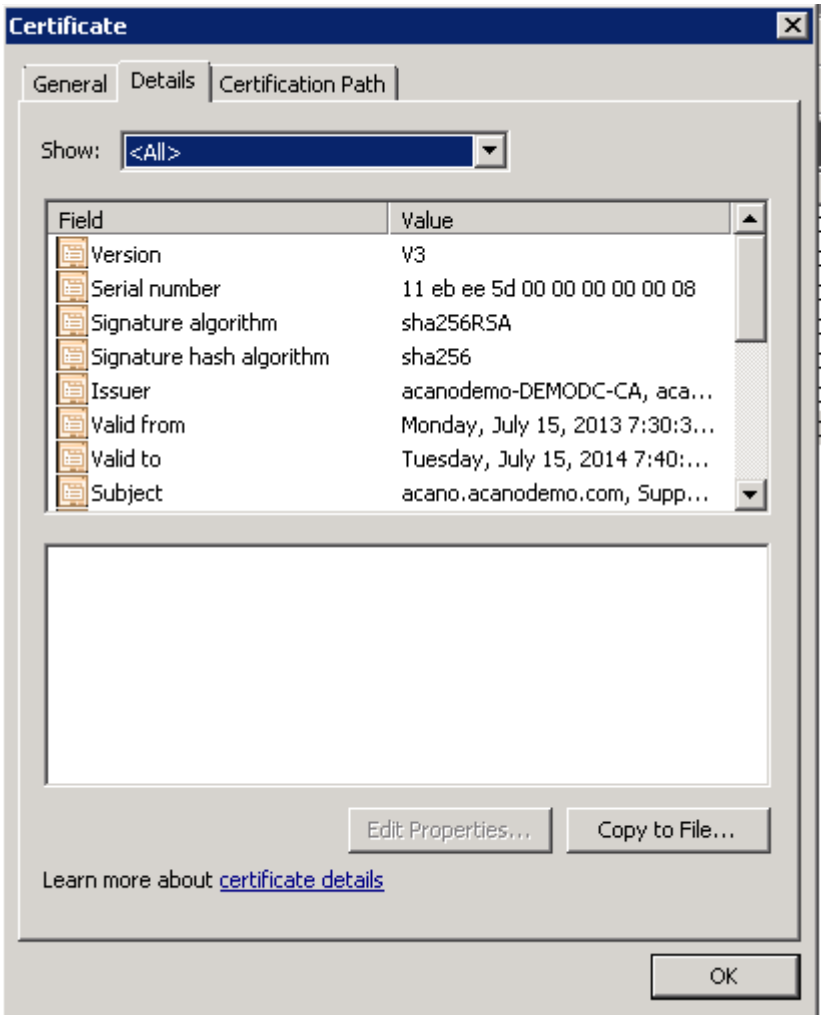
```
C:\Users\Administrator>certreq -submit -attrib "CertificateTemplate:WebServer" C
:\Users\Administrator\Desktop\demokitcsr.pem
Active Directory Enrollment Policy
  {0BD5D0B7-591F-4C77-AFEC-3C0E470F77D5}
  ldap:
RequestId: 8
RequestId: "8"
Certificate request is pending: Taken Under Submission (0)

C:\Users\Administrator>_
```

次の手順に従って、発行された証明書を取得します。

a. CA の [サーバ マネージャ（Server Manager）] ページで、CA のロールの下にある **Pending Requests** フォルダを見つけます。

b. **cmd** ウィンドウに表示された要求 ID と一致する保留中の要求を右クリックし、[すべてのタスク（All Tasks）] > [発行（Issue）] を選択します。

c. 発行された署名付き証明書が [発行した証明書（Issued Certificates）] フォルダに保存されます。証明書をダブルクリックして開き、[詳細（Details）] タブを開きます。



d. [ファイルにコピー（Copy to File）] をクリックすると、証明書エクスポート ウィザードが開始されます。

e. Base-64 encoded X.509（.CER）を選択して、[次へ（Next）] をクリックします。

f. 証明書の保存先を開き、**xmpp** などの名前を入力して、[次へ（Next）] をクリックします。

g. 生成された証明書を .crt、.cer、または .pem 拡張子で保存します（例：**xmpp.crt**）。

4. Meeting Server での署名付き証明書と秘密キーのインストールについては、第 4 章を参照してください。

4 Meeting Server での署名付き証明書と秘密キーのインストール

第 2 章で説明したように、スケーラブルで復元力のある Meeting Server 導入環境には、以下を対象としたパブリック CA 署名付き証明書が必要です。

- **Web bridge**（Web RTC クライアントをエンドユーザが使用できるようにする場合）。Web RTC クライアントが接続を信頼するには、Web Bridge のパブリック CA 署名付き証明書が必要です。
- **TURN サーバ**（セキュア通信に TLS 接続を使用する場合）。
- **XMPP サーバ**（ネイティブ シスコ ミーティング アプリケーション（PC、Mac、iOS）をエンドユーザが使用する場合）。ネイティブ シスコ ミーティング アプリケーションが接続を信頼するには、XMPP サーバのパブリック CA 署名付き証明書が必要です。
- **Call Bridge**（パブリック ネットワークを介した Lync とのダイレクト フェデレーションが必要な場合）。Lync Edge サーバが接続を信頼するには、Call Bridge のパブリック CA 署名付き証明書が必要です。

以下を対象とした内部 CA 署名付き証明書も必要です。

- **Web 管理**。Meeting Server API は Web 管理インターフェイス経由でルーティングされます。したがって、Web 管理インターフェイスではなく API を使用して Call Bridge を設定した場合も証明書が必要です。
- **Call Bridge**。Web Bridge は Call Bridge の証明書を要求します。Active Directory サーバも Call Bridge の証明書を要求します。また、導入環境に SIP トランクが存在する場合は、Call Bridge が SIP コール制御デバイスと相互認証する際に証明書が必要になります。
- **ロード バランサ**（分割サーバ導入で、ネイティブ シスコ ミーティング アプリケーション（PC、Mac、iOS）をエンドユーザが使用する場合）。Core サーバと Edge サーバ間のトランクは、ロード バランサによって提示される証明書を認証（信頼）する必要があります。これは、XMPP サーバが接続を信頼できるように、ロード バランサとトランクの間で設定する必要がある相互認証の一部です。ロード バランサとトランクで同じ証明書を使用することはできません。注：外部のネイティブ シスコ ミーティング アプリケーションはロード バランサ証明書を確認しません。
- **トランク**（分割サーバ導入で、ネイティブ シスコ ミーティング アプリケーション（PC、Mac、iOS）をエンドユーザが使用する場合）。Edge サーバのロード バランサは、トランクによって提示される証明書を認証（信頼）する必要があります。これも、ロード バランサとトランクの間で設定する必要がある相互認証の一部です。トランクとロード バランサで同じ証明書を使用することはできません。

- **データベース クラスタリング**（各データベース サーバとデータベース クライアント（データベースと同じ場所に配置されていない Call Bridge を含む）に、同じ認証局によって署名された秘密キーと証明書が必要な場合）。

注：本ガイドは、『Meeting Server Installation Guide』の説明に従って、Web 管理インターフェイスの秘密キー/証明書ペアがインストールされていることを前提としています。セットアップされていない場合は、ここでセットアップします。

4.1 秘密キーと証明書の再使用

証明書ごとに異なる秘密キー/証明書ペアをインストールする必要はありません。場合によっては、秘密キーと証明書をコピーして複数のサービスで再使用できます。秘密キー/証明書ペアを再使用する場合の推奨事項は次のとおりです。

- Lync 導入環境を Meeting Server に接続する場合は、その Lync 導入環境で信頼されている認証局（CA）を使用する。
- 証明書と秘密キーのファイル名には、使用される場所を反映した名前を使用する（例：**`webadmin.crt`** と **`webadmin.key`**）。
- トランクとロード バランサには異なる秘密キー/証明書ペアが必要なため、両者の間で同じ秘密キー/証明書ペアを使用しない（3.1.3 項を参照）。

4.1.1 秘密キーと証明書の再使用の例

2 つの Edge サーバ上でロード バランサが有効になっていて、単一の Core サーバ上で XMPP サービスを実行している導入環境では、edge1.key、edge1.crt、edge2.key、edge2.crt、core.key、core.crt を生成して、両方のトランクで core.key と core.crt を再使用できます。

4.2 秘密キーと証明書の MMP へのアップロード

1. MMP に SSH でログインします。
2. SFTP を使用して、各秘密キー/証明書ペアと証明書バンドルをアップロードします。
3. MMP PKI コマンドの **`pki list`** を使用して、アップロードされたファイルを確認します。**`pki list`** では、MMP にアップロードされた SSH キーと CSR ファイルも表示されます。

注：秘密キーと証明書のファイル名には、ファイル拡張子の直前以外に「.」を使用しないでください。たとえば callbridge.key は有効ですが、call.bridge.key は使用できません。

4.3 ファイル タイプの検証および証明書と秘密キーの一致の確認

Meeting Server で秘密キー/証明書ペアをインストールする前に、インストールするファイルが適切であることを確認します。ここでは、MMP コマンドの pki inspect、pki match、pki verify を使用して、インストールするファイルの ID を確認する方法について説明します。

ファイルを検証して、現在も有効であるかどうか（失効日）を特定するには、次のように指定します。

pki inspect <filename>

証明書が秘密キーと一致することを確認するには、次のように指定します。

pki match <keyfile> <certificatefile>

証明書が CA によって署名されており、証明書バンドルを使用してこの証明書をアサートできることを確認するには、次のように指定します。

pki verify <cert> <certbundle/CAcert>

次に例を示します。

1. MMP に SSH でログインします。
2. 次のコマンドを入力します。

   pki inspect xmppserver.crt

   証明書がまだ有効であるかどうかなど、ファイルの内容を検証します。
3. 次のコマンドを入力します。

   pki match xmppserver.key xmppserver.crt

   ファイル **xmppserver.key** がファイル xmppserver.crt に一致すること、および 2 つのファイルが 1 つの使用可能な ID を形成することを確認します。
4. 次のコマンドを入力します。

   pki verify xmppserver.crt xmppbundle.crt

   xmppserver.crt が信頼できる CA によって署名されていて、xmppbundle.crt 内の中間証明書のチェーンにより信頼チェーンが確立されていることを確認します。

4.4 XMPP サーバの証明書と秘密キーのインストール

Meeting Server でエンドユーザが PC、Mac、および iOS デバイス向けのネイティブ シスコ ミーティング アプリケーションを使用できる場合は、XMPP サーバのパブリック CA 署名付き証明書をインストールする必要があります。ミーティング アプリケーションは、接続の初期設定時にこの証明書を使用して XMPP サーバとの接続が信頼できるかどうかを特定します。

次の手順は、XMPP サーバがリッスンに使用するネットワーク インターフェイスがすでに設定されていることを前提としています。証明書を割り当てる前に、**listen** MMP コマンドを使用したインターフェイスの設定について、『Scalable and Resilient Server Deployment Guide』を参照してください。

1. MMP に SSH でログインします。
2. 証明書を割り当てる前に、XMPP サーバ インターフェイスを無効にします。

   xmpp disable
3. 次のコマンドを使用して秘密キー/証明書ペアを割り当てます。

   xmpp certs <keyfile> <certificatefile> [<cert-bundle]

   keyfile と **certificatefile** は、それぞれ対応する秘密キーと証明書のファイル名です。CA により証明書バンドルが提供されている場合は、バンドルも個別のファイルとして証明書に含めます。

   次に例を示します。

   xmpp certs xmppserver.key xmppserver.crt xmppserverbundle.crt
4. XMPP サーバ インターフェイスを再度有効化します。

   xmpp enable

証明書が XMPP サーバに正常にインストールされると、次のメッセージが表示されます。

```
SUCCESS: Domain configured
SUCCESS: Key and certificate pair match
SUCCESS: license file present
SUCCESS: XMPP server enabled
```

証明書のインストールに失敗した場合は、次のエラー メッセージが表示されます。

```
FAILURE: Key and certificate problem: certificate and key do not
match
FAILURE: XMPP server configuration not complete
```

証明書バンドルに問題がある場合は、次のようなエラー メッセージが表示されます。

```
SUCCESS: Domain configured
SUCCESS: Key and certificate pair match
FAILURE: certificate verification error: depth=x issuer= x = xx, ST =
xxxxxxx, L = xxxxxx, O = xxxxx, OU = xxxxxx, CN = xxxxxxxxx,
emailAddress = xxxxxxxx@xxxxx.xxx Verification error: unable to get
issuer certificate Failed cert:
Certificate:
    Data:
    Version: xx
    Serial Number: xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=x, ST= xxxxxxx, L= xxxxxx, O= xxxxx, OU= xxxxxx, CN=
xxxxxxxxx, emailAddress = xxxxxxxx@xxxxx.xxx
    Validity
      Not Before: <month><time><year>
      Not After: <month><time><year>
        Subject: C=xx, O=xxxxx, OU=xxxxxxxxxxx, CN=xxxxxxxxxxxxx
        Subject Public Key Info:
```

```
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
SUCCESS: license file present
FAILURE: XMPP server configuration not complete
```

証明書バンドルを調べて証明書チェーンが破損していないことを確認してください。

注：XMPP のライセンス キー ファイル（license.dat）も Meeting Server にインストールする必要があります。『Scalable and Resilient Server Deployment Guide』の手順に従ってください。

4.5 Core と Edge 間のトランクの証明書と秘密キーのインストール

この項の手順は、Meeting Server をネイティブ シスコ ミーティング アプリケーション（PC クライアント、Mac クライアント、または iOS クライアント）に接続し、Core サーバと Edge サーバを分割導入している場合にのみ実行する必要があります。

注：分割サーバによる導入環境では、各 Core サーバは複数の Edge サーバに対して複数のトランクを作成できます。各 Edge サーバは、複数の Core サーバからの複数のトランクを受け入れることができます。

Edge サーバのロード バランサと Core サーバの XMPP サービスの間で相互認証を確立するには、次を実行する必要があります。

- ロード バランサの秘密キー/証明書ペアとトランクの証明書をロード バランサにインストールする。
- トランクの秘密キー/証明書ペアとロード バランサの証明書をトランクにインストールする。

注：ロード バランサ証明書をトランクで再使用しないでください（逆も同様です）。ロード バランサとトランクには別々の証明書を作成する必要があります。

これによってトランクとロード バランサが互いの証明書を認証し、ロード バランサと XMPP サーバの間にセキュアな TLS トランクが作成されます。

導入環境に配置された XMPP サーバが 1 台でも、Edge サーバが複数ある場合は、各 Edge サーバにロード バランサが存在することがあります。この場合、Core サーバ（XMPP サービスをホストしている）は各ロード バランサへのトランクを作成する必要があります。トランクとロード バランサのペアごとに、次の手順 1 ～ 12 を実行します。秘密キー/証明書ペアを再使用する場合の推奨事項については、4.1 項を参照してください。

注：本ガイドは、Meeting Server でロード バランサ用のネットワーク インターフェイスとトランク インターフェイスがすでに設定されていることを前提としています。まだ設定していない場合は、証明書を割り当てる前に、『Scalable and Resilient Server Deployment Guide』の手順を実行してください。

1. Edge サーバの MMP に SSH でログインします。
2. 次のコマンドを使用して、Edge インスタンスを作成します。

loadbalancer create <tag>
たとえば Edge サーバのタグが「Edge1to LB」の場合は、次のように入力します。

loadbalancer create Edge1toLB

3. 次のコマンドを使用して、ロード バランサに秘密キー/証明書ペアとトランクの証明書を割り当てます。

   loadbalancer auth <tag> <keyfile> <certificatefile> <trust-bundle>
   keyfile と **certificatefile** は、ロード バランサの対応する秘密キー/証明書ペアのファイル名です。<trust-bundle> はトランクの証明書です。

   次に例を示します。

   loadbalancer auth Edge1toLB edge1.key edge1.crt core1.crt

   注：トランクの証明書 **core1.crt** は、Edge サーバに「信頼バンドル」として追加されます。

4. 次のコマンドを使用して、トランク インターフェイスおよびポートを設定します。

   loadbalancer trunk <tag> <iface>:<port>
   たとえば、インターフェイス A のポート 4999 でトランク接続を許可する場合は、次のように入力します。

   loadbalancer trunk Edge1toLB a:4999

5. 次のコマンドを使用して、パブリック インターフェイスおよびポート（クライアント接続を受け入れる）を設定します。

   loadbalancer public <tag> <iface:port whitelist>
   たとえば、B のポート 5222 でクライアント接続を許可する場合は、次のように入力します。
   loadbalancer public Edge1toLB b:5222

6. 一般的な Edge サーバ導入環境では、Web Bridge も有効化され、トランクを使用する必要があります。これを許可するには、次のようにパブリック インターフェイスとしてループバックを設定します。
   loadbalancer public Edge1toLB b:5222 lo:5222

7. 次のコマンドを使用して、トランクを有効にします。

   loadbalancer enable <tag>
   例：
   loadbalancer enable Edge1toLB

   注：パブリック ポートは接続を処理するトランクが存在しない限り開かれません。

8. Core サーバの MMP に SSH でログインします。

9. Core サーバと Edge サーバの間に xmpp トラフィック用のトランクを作成します。

   trunk create <tag> <port/service name>
   次に例を示します。
   trunk create trunktoEdge1 xmpp

10. 次のコマンドを使用して、トランクに秘密キー/証明書ペアとロード バランサの証明書を割り当てます。

    trunk auth <tag> <key-file> <cert-file> <trust-bundle>

    keyfile と **certificatefile** は、トランクの対応する秘密キー/証明書ペアのファイル名です。**<trust-bundle>** はロード バランサの証明書です。

    次に例を示します。

    trunk auth trunktoEdge1 core1.key core1.crt edge1.crt

    注：ロード バランサ証明書 **edge1.crt** は、Core サーバに「信頼バンドル」として追加されます。

11. 次のコマンドを使用して、このトランクが接続する Edge サーバを設定します。

    trunk edge <tag> <edge name/ip address> [<default port>]

    たとえば、Edge サーバの名前が edge1.example.com で、ポート 4999 を使用する場合は、次のように入力します。

    trunk edge trunktoEdge1 edge1.example.com 4999

    注：ドメイン名が複数の IP アドレスに解決されると、そのすべてに対して接続が試行されます。

12. トランク インターフェイスを有効にします。

    trunk enable <tag>

    次に例を示します。

    trunk enable trunktoEdge1

    注：ロード バランサとトランクのコマンドの詳細なリストを確認するには、『Cisco Meeting Server MMP Command Reference』を参照してください。

4.6 Web Bridge の証明書と秘密キーのインストール

Meeting Server でエンドユーザが WebRTC アプリケーションを使用できる場合は、Web Bridge のパブリック CA 署名付き証明書をインストールする必要があります。ブラウザは、この証明書を使用して Web Bridge との接続が信頼できるかどうかを特定します。

次の手順は、DNS レコードおよび Web Bridge がリッスンに使用するネットワーク インターフェイスがすでに設定されていることを前提としています。証明書を割り当てる前に、**listen** MMP コマンドを使用したインターフェイスの設定について、『Scalable and Resilient Server Deployment Guide』を参照してください。

Web Bridge ごとに次の手順を実行します。

1. MMP に SSH でログインします。
2. 証明書を割り当てる前に、Web Bridge インターフェイスを無効にします。
   webbridge disable
3. 次のコマンドを使用して秘密キー/証明書ペアを割り当てます。

   webbridge certs <keyfile> <certificatefile> [<cert-bundle>]

   keyfile と **certificatefile** は、それぞれ対応する秘密キーと証明書のファイル名です。CA により証明書バンドルが提供されている場合は、バンドルも個別のファイルとして証明書に含めます。

   次に例を示します。

   webbridge certs webbridge.key webbridge.crt webbridgebundle.crt
4. Web Bridge インターフェイスを再度有効化します。
   webbridge enable

証明書が Web Bridge に正常にインストールされると、次のメッセージが表示されます。

```
SUCCESS: Key and certificate pair match
SUCCESS: Webbridge enabled
```

証明書のインストールに失敗した場合は、次のエラー メッセージが表示されます。

```
FAILURE: Key and certificate problem: certificate and key do not match
FAILURE: Webbridge configuration not complete
```

注：Web Bridge から証明書設定を削除するには、MMP コマンド **webbridge certs none** を使用します。

4.7 Call Bridge の証明書と秘密キーのインストール

導入環境での各 Call Bridge の用途によっては、秘密キー/証明書ペアが必要になることがあります。

- Web Bridge との通信を確立するために使用する場合、導入環境のセキュリティを確保するには、信頼できる Call Bridge からの設定のみが受け入れられるようにすることが重要です。
- SIP コール制御デバイスとの TLS 接続を確立するために使用する場合。
- Lync フロントエンド（FE）サーバとの TLS 接続を確立するために使用する場合、証明書が Lync FE サーバによって信頼されるようにするには、次の条件を満たす必要があります。
  - 証明書の **CN** は、Lync FE サーバで Meeting Server を信頼できるアプリケーションおよびスタティック ルートとして設定する際に追加した FQDN と同じである必要があります。
  - 証明書に **subjectAltName** リストが含まれる場合は、このリストに FQDN も追加する必要があります。
  - Lync FE サーバの証明書を発行した CA など、信頼できる CA サーバを使用して証明書に署名します。

次の手順は、各 Call Bridge がリッスンに使用するネットワーク インターフェイスがすでに設定されていることを前提としています。証明書を割り当てる前に、**listen** MMP コマンドを使用したインターフェイスの設定について、『Scalable and Resilient Server Deployment Guide』を参照してください。

Call Bridge ごとに次の手順を実行します。

1. MMP に SSH でログインします。
2. 次のコマンドを使用して秘密キー/証明書ペアを割り当てます。

   callbridge certs <keyfile> <certificatefile>[<cert-bundle>]

   keyfile と **certificatefile** は、それぞれ対応する秘密キーと証明書のファイル名です。CA により証明書バンドルが提供されている場合は、バンドルも個別のファイルとして証明書に含めます。

   次に例を示します。

   callbridge certs callbridge.key callbridge.crt callbridgebundle.crt
3. 変更を適用するには、Call Bridge インターフェイスを再起動します。

   callbridge restart

証明書が Call Bridge に正常にインストールされると、次のメッセージが表示されます。

```
SUCCESS: listen interface configured
SUCCESS: Key and certificate pair match
```

証明書のインストールに失敗した場合は、次のエラー メッセージが表示されます。

```
FAILURE: Key and certificate problem: certificate and key do not match
```

注：Web Bridge の設定後、すべての Web Bridge の信頼ストアに Call Bridge 証明書を追加する必要があります。4.7.1 項を参照してください。

注：Call Bridge から証明書設定を削除するには、MMP コマンド **callbridge certs none** を使用します。

4.7.1 Call Bridge と Web Bridge 間の信頼の確立

Call Bridge から Web Bridge にゲスト ログインおよびイメージ カスタマイゼーションの設定をプッシュできます（『Customization Guidelines』を参照してください）。導入環境のセキュリティを確保するには、信頼できる Call Bridge からの設定のみが受け入れられるようにすることが重要です。

Call Bridge と Web Bridge 間の信頼は、Call Bridge のパブリック証明書を Web Bridge に提供することで確立されます。Web Bridge はこれを使用して、証明書の所有者であることを暗号化方式で証明するよう、Call Bridge にチャレンジを実行できます。Call Bridge が信頼できるいずれかの証明書の所有者であることを証明できなければ、Web Bridge は設定を受け入れません。

注：分割導入の場合は、Edge サーバで **webbridge trust <callbridge_cert>** コマンドを使用する前に、Call Bridge 証明書を Core サーバから Edge サーバにコピーする必要があります。Call Bridge を実行している Core サーバで次のコマンドを実行します。

```
cms>callbridge
Listening interfaces : a
Key file             : callbridge.key
Certificate file     : callbridge.cer
```

SFTP を使用して、Web Bridge を実行している Edge サーバに Core サーバから証明書およびキー ファイルをコピーします。次に、Edge サーバ上の Web Bridge 信頼ストアに証明書を追加します。

Call Bridge 証明書を Web Bridge 信頼ストアに追加するには、次の手順を実行します。

1. **callbridge** コマンドを発行して、Call Bridge が使用している証明書を確認します。
2. Web Bridge を無効にします。
3. 次のコマンドを使用して、Call Bridge 証明書を信頼ストアに追加します。

   webbridge trust <callbridgecert|cert-bundle>
   次の中から導入環境に最適な方法を使用します。

   a. Call Bridge が 1 つのみで Web Bridge が複数ある場合は、各 Web Bridge にログインして、Call Bridge 証明書を各 Web Bridge の信頼ストアに追加します。cms の例を示します。

   ```
   cms>webbridge disable
   cms>webbridge trust callbridge.crt
   cms>webbridge enable
   SUCCESS: Key and certificate pair match
   SUCCESS: webbridge enabled
   ```

   b. 同じ Call Bridge 証明書を使用する Call Bridge が複数ある場合は、この Call Bridge 証明書を各 Web Bridge の信頼ストアに追加します。次に例を示します。

   ```
   cms>webbridge disable
   cms>webbridge trust callbridge.crt
   cms>webbridge enable
   SUCCESS: Key and certificate pair match
   SUCCESS: webbridge enabled
   ```

   c. 異なる証明書ファイルを持つ複数の Call Bridge がある場合は、次を実行します。

      i. 次のいずれかの方法で、すべての証明書を 1 つの信頼できる証明書バンドルにまとめます。

         - Linux または Unix 系のオペレーティング システム

           ```
           cat cert1.crt cert2.crt cert3.crt >
           combinedcallbridgecerts.crt
           ```

- Windows または DOS

  ```
  copy cert1.crt + cert2.crt + cert3.crt
  combinedcallbridgecert.crt
  ```

- メモ帳またはメモ帳 ++ を使用して手動で証明書をまとめます。最初の証明書の「END CERTIFICATE」行と 2 つ目（および以降）の証明書の「BEGIN CERTIFICATE」行の間にスペースを挿入しないでください。ただし、ファイルの最後には改行が必要です。また、Base64 エンコード形式である必要があります。

ii. その後、次のコマンドを使用して各 Web Bridge でこの証明書バンドルを展開します。

```
webbridge trust combinedcallbridgecert.crt
```

4. Web Bridge を再度有効化します。
5. Web Bridge の信頼ストアに Call Bridge 証明書が追加されていることを確認するには、次のコマンドを実行します。

```
cms>webbridge
Enabled                 : true
Interface whitelist     : a:443
Key file                : webbridge.key
Certificate file        : webbridge.crt
Trust bundle            : callbridge.crt
HTTP redirect           : Enabled
```

4.8 データベース クラスタリングの証明書と秘密キーのインストール

注意：次の手順は、無効化したデータベース クラスタでのみ実行できます。すでにデータベース クラスタを設定済みである場合は、クラスタ内のすべてのサーバで **database cluster remove** コマンドを実行してから、クラスタを再作成する前にこの項のコマンドを実行してください。

データベース クライアントの証明書では、CN を「postgres」に設定する必要があります。証明書が適切であることを確認するには、**pki inspect** コマンドを使用します。次に例を示します。

```
cms> pki inspect dbclient.crt
Checking ssh public keys...not found
Checking user configured certificates and keys...found
File contains a PEM encoded certificate
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:

58:00:00:00:1c:3b:92:8a:95:d2:21:89:58:00:00:00:00:00:1c
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=support, CN=support-DC2-CA
        Validity
            Not Before: Sep 13 13:32:38 2015 GMT
            Not After : Sep 13 13:42:38 2017 GMT
        Subject: CN=postgres
```

データベース クライアント証明書の CN が postgres 以外に設定されていると、「エラー：クライアント証明書の共通名が正しくありません（ERROR: Client certificate common name incorrect）」というエラー メッセージが表示されます。

1. Call Bridge と同じ場所に配置されているかどうかに関係なく、すべてのデータベースのホスト サーバで次の手順を実行します。
   a. ホスト サーバに次の証明書とキーを SFTP で送信します。
      - dbcluster_server.key
      - dbcluster_server.crt
      - dbcluster_client.key
      - dbcluster_client.crt
      - 内部 CA によって提供された証明書バンドルも送信します。付録 A の手順を使用した場合、ファイルは dbcluster_ca.crt です。
   b. データベース クラスタの作成時に使用する証明書を指定します。次に例を示します。

```
cms>database cluster certs dbcluster_server.key dbcluster_
server.crt dbcluster_client.key dbcluster_client.crt dbcluster_
cert-bundle.crt
Certificates updated
cms> database cluster status
Status                    : Disabled
Interface                 : a
Certificates
  Server Key              : dbcluster_server.key
  Server Certificate      : dbcluster_server.crt
  Client Key              : dbcluster_client.key
  Client Certificate      : dbcluster_client.crt
  CA Certificate          : dbcluster_cert-bundle.crt
```

2. データベースと異なる場所に配置されたすべての Call Bridge に対して、次の手順を実行します。
   a. Call Bridge サーバに次の証明書とキーを SFTP で送信します。
      - dbcluster_client.key
      - dbcluster_client.crt
      - 内部 CA によって提供された証明書バンドルも送信します。付録 A の手順を使用した場合、ファイルは dbcluster_ca.crt です。
   b. これらの証明書を使用するようにデータベース クラスタを設定します。

```
cms> database cluster certs dbcluster_client.key dbcluster_
client.crt dbcluster_cert-bundle.crt
Certificates updated
cms> database cluster status
Status                    : Disabled
Interface                 : a
```

```
Certificates
  Client Key              : dbcluster_client.key
  Client Certificate      : dbcluster_client.crt
  CA Certificate          : dbcluster_cert-bundle.crt
```

3. マスター データベースの選択とデータベース クラスタの作成の詳細については、ここで再度『Scalable and Resilient Server Deployment Guide』を参照してください。

4.9 TURN サーバの証明書と秘密キーのインストール

セキュア通信に TLS を使用する場合は、Web Bridge と同じ CA を使用して署名した、TURN サーバの署名付き証明書をインストールする必要があります。ブラウザは、この証明書を使用して Meeting Server との接続が信頼できるかどうかを特定します。

次の手順は、TURN サーバがリッスンに使用するネットワーク インターフェイスがすでに設定されていることを前提としています。証明書を割り当てる前に、**listen** MMP コマンドを使用したインターフェイスの設定について、『Scalable and Resilient Server Deployment Guide』を参照してください。

TURN サーバごとに次の手順を実行します。

1. ホスト サーバの MMP に SSH でログインします。
2. 証明書を指定する前に、TURN サーバ インターフェイスを無効にします。

   turn disable
3. 署名付き証明書と中間 CA バンドル（存在する場合）を、SFTP を使用して Meeting Server にアップロードします。
4. 証明書（および証明書バンドル）と秘密キーが一致していることを確認します。

   pki verify <certicate> <cert bundle/CA cert> [<CA cert>]
5. 証明書（および証明書バンドル）と秘密キーのペアを TURN サーバに割り合てます。

   turn certs <keyfile> <certificatefile> [<cert-bundle>]

   keyfile と certificatefile は、それぞれ対応する秘密キーと証明書のファイル名です。CA により証明書バンドルが提供されている場合は、バンドルも個別のファイルとして証明書に含めます。

   次に例を示します。

   turn certs turn.key turn.crt turnbundle.crt
6. TURN サーバを再度有効化します。

   turn enable

4.10 TLS 証明書の検証

リモートの証明書が信頼されていることを検証するために、SIP および LDAP の相互認証を有効にできます。有効にすると、Call Bridge は（どちら側が接続を開始したかに関係なく）常にリモートの証明書を要求し、サーバでアップロードおよび定義された信頼ストアに対して提示された証明書を比較します。

使用可能な MMP コマンドは次のとおりです。

- 現在の設定を表示する。

  `tls <sip|ldap>`
- 信頼できる認証局を定義する。

  `tls <sip|ldap> trust <crt bundle>`
- 証明書の検証を有効または無効にする、または OCSP を検証に使用するかどうかを指定する。

  `tls <sip|ldap> verify enable|disable|ocsp`

詳細については、『MMP Command Reference』を参照してください。

5 証明書に関する問題のトラブルシューティング

ここでは、いくつかのよくある問題のトラブルシューティングについて説明します。証明書に関する Meeting Server Knowledgebase for further Frequently Asked Questions を参照してください。

5.1 サービスが信頼できないことを示す警告メッセージ

次の場合にメッセージが表示されます。

- 信頼ストアに含まれていない内部 CA を使用している。
- パブリック CA または内部 CA の署名付き証明書が必要なケースで自己署名証明書を使用している（証明書を再発行して、信頼できる CA に署名を要求してください。このコンポーネントへのパブリック アクセスが不要な場合は、内部 CA を使用することもできます）。

5.2 クライアント証明書のエラー

クライアントは XMPP サーバと通信するために接続を信頼する必要があります。クライアントが接続を信頼し、クライアント証明書のエラーが発生しないようにするには、XMPP サーバの証明書に次の情報が含まれている必要があります。

- XMPP サーバの DNS レコード（**CN** フィールドおよび **subjectAltName** フィールド）
- XMPP ドメイン名（**subjectAltName** フィールド）

たとえば、クライアントが「firstname.surname@example.com」を使用している場合、XMPP ドメインは example.com、DNS は xmpp.example.com です。CN に xmpp.example.com を指定して、SAN リスト内に xmpp.example.com と example.com を追加する必要があります。

```
pki csr xmppserver CN:xmpp.example.com O:"Example Inc."
subjectAltName:xmpp.example.com,example.com
```

ワイルドカードを使用する場合は次のとおりです。

```
pki csr xmppserver CN:*.example.com O:"Example Inc."
subjectAltName:*.example.com,example.com
```

5.3 ブラウザ証明書のエラー

WebRTC クライアントは Web Bridge と通信するために接続を信頼する必要があります。クライアントが接続を信頼するには、Web Bridge の証明書にドメイン ネーム システム（DNS）内のサーバの正確な場所を指定する完全修飾ドメイン名（FQDN）を含める必要があります。これは Web Bridge サービスの DNS A レコードで定義されます。Web Bridge 証明書に FQDN が指定されていないと、ブラウザ証明書のエラーが発生します。

また、TLS を Meeting Server に設定する場合は、Web Bridge に割り当てられた証明書/キーペアに類似した証明書が TURN サーバに必要です。TURN サーバに適切な証明書/キー ペアが割り当てられていないと、クライアントが接続を信頼せずに、ブラウザに証明書エラーが表示される可能性があります。

5.4 Call Bridge が Web Bridge に接続できない

Web Bridge の信頼ストアに Call Bridge の証明書がないか、証明書の有効期限が切れています（認証エラー）。

Web Bridge の信頼バンドルに含まれる証明書を表示するには、**`webbridge`** を入力します。

証明書の有効性を確認するには、**`pki inspect <certificate name>`** を入力します。

5.5 トランクとロード バランサ間で同じ秘密キー/証明書ペアを使用する

秘密キー/証明書ペアがトランクとロード バランサで同一であるため、両者が互いを認証できません（信頼バンドルを使用した場合も同様）。トランクとロード バランサ用に異なる秘密キー/証明書ペアを作成してください。

5.6 Lync フロントエンド サーバへの接続の問題

Call Bridge 証明書に署名した CA が Lync フロントエンド サーバの証明書の署名に使用された CA と同じであることを確認します。Call Bridge 証明書が Lync フロントエンド サーバの証明書の署名に使用された同じ CA によって署名されていない場合は、Call Bridge の信頼できる CA 証明書を Lync サーバのルート信頼ストアにアップロードして、Lync サーバが Call Bridge 証明書を信頼できるようにしてください。

Lync に追加された FQDN が Call Bridge の証明書の CN でもあることを確認します。

6 テスト環境での証明書の作成と使用

pki selfsigned コマンドを使用して、Meeting Server で秘密キーと自己署名証明書を作成できます。

自己署名証明書は、「クラスタ キー」（CA 証明書を取得できない）または Lync 認証（CA が信頼できる機関ではない）には使用できません。ただし、Web 管理、トランク/ロード バランサ、および Call Bridge と Web Bridge 間の相互認証に自己署名証明書を使用できます（この場合もブラウザに証明書エラーは表示されます）。自己署名証明書は、実稼働環境ではなくテスト環境で使用することを強く推奨します。

Meeting Server でローカル秘密キーおよび自己署名証明書を生成する手順は次のとおりです。

1. MMP にログインして次のコマンドを入力します。

   pki selfsigned <key/cert basename>

   <key/cert basename> は、生成するキーと証明書を識別します。

   次に例を示します。

   pki selfsigned callbridge

   callbridge.key という名前のローカル秘密キーと **callbridge.crt** という名前の自己署名証明書が作成されます。

付録 A　証明書生成用の OpenSSL コマンド

第 3 章で説明されている MMP **pki** コマンドの代わりに、OpenSSL を使用して秘密キー、証明書署名要求、および証明書を生成できます。この付録では、OpenSSL コマンドの使用について詳しく説明します。ここに記載された例は、OpenSSL を Windows 上で実行することを前提としていますが、OpenSSL は他のプラットフォームで使用することもできます。

注：OpenSSL は管理者モードで実行してください。

A.1　RSA 秘密キーと CSR ファイルの生成

コンピュータ上で OpenSSL ツールキットを使用します。

新しい RSA 秘密キーと CSR ファイルを生成するには、次のコマンドを使用します。

```
openssl req -out <certname>.csr -new -newkey rsa:2048 -nodes -keyout
<keyname>.key
```

次に例を示します。

```
openssl req -out webbridge.csr -new -newkey rsa:2048 -nodes -keyout
webbridge.key
```

webbridge.csr という名前の CSR ファイルと webbridge.key という名前の RSA 2048 ビットの秘密キーが生成されます。

注：keyname と certname に「.」を含めることはできません。たとえば、webbridge は有効ですが、web.bridge は使用できません。

既存の秘密キーの証明書署名要求（CSR）を生成するには、次のコマンドを使用します。

```
openssl req -out <certname>.csr -key <keyname>.key -new
```

次に例を示します。

```
openssl req -out xmppserver.csr -key xmppserver.key -new
```

xmppserver.key という名前の既存の秘密キーに基づいた、xmppserver.csr という名前の CSR ファイルが生成されます。

OpenSSL を使用して証明書に自己署名する場合は、中間 CSR ファイルは必要ありません。次の項に進んでください。

A.2　CSR ファイルの署名

パブリック CA を使用して CSR に署名する場合は、3.2 項の手順を実行します。

内部 CA を使用して CSR に署名する場合は、3.3 項の手順を実行します。

証明書に自己署名する場合は、次の OpenSSL コマンドを使用します。

```
openssl req -x509 -nodes -days 100 -newkey rsa:2048 -keyout
<keyname>.key -out <certname>.crt
```

次に例を示します。

```
openssl req -x509 -nodes -days 100 -newkey rsa:2048 -keyout
callbridge.key -out callbridge.crt
```

callbridge.key という名前の新しい秘密キーと callbridge.crt という名前の（最終）証明書が生成されます。

A.3 データベース クラスタリングの証明書の作成

ここでは、OpenSSL コマンドを使用してデータベース クラスタリングの証明書を作成する方法について説明します。

データベース クラスタリング用に作成された証明書は、同じ認証局（CA）によって署名される必要があります。データベース クラスタリングにユーザがアクセスすることはできないため、CA キーと証明書は OpenSSL を使用して内部で生成できます。

注：OpenSSL は管理者権限を使って実行してください。

注：keyname と certname に「.」を含めることはできません。たとえば、**dbcluster_ca** は有効ですが、**dbcluster.ca** は使用できません。

手順は次のとおりです。

1. CA を定義して、CA の秘密/公開キー ペアと証明書を作成します。

   a. 定義した CA の秘密キーと証明書要求（.csr）のペアを生成します。次の OpenSSL 構文を使用します。

   ```
   openssl req -new -text -nodes -keyout <keyname>.key -out
   <certname>.csr -subj /C=<country>/ST=<state>/L=<location>
   /O=<organization>/OU=<organizational unit>/CN=<authorityname>
   ```

   次に例を示します。

   ```
   openssl req -new -text -nodes -keyout dbcluster_ca.key -out
   dbcluster_ca.csr -subj /C=UK/ST=London/L=London/O=Example
   /OU=/CN=example
   ```

   -subj の後の属性で定義された CA 用に、秘密キー **dbcluster_ca.key** と証明書署名要求ファイル **dbcluster_ca.csr** が作成されます。

   b. 手順 1a で生成された秘密キーと証明書要求（.csr）を使用して、CA の証明書を作成します。

   ```
   openssl req -x509 -text -in dbcluster_ca.csr -key dbcluster_
   ca.key -out dbcluster_ca.crt -days 3650
   ```

   証明書 **dbcluster_ca.crt** が作成されます。

2. 手順 1 で生成された CA のクレデンシャルを使用して、データベース サーバおよびデータベース クライアントの秘密キーと署名付き証明書を出力します。

   a. データベース サーバの秘密キーと証明書要求（.csr）を生成します。

      openssl req -new -nodes -keyout <keyname>.key -out <certname>.csr -subj /C=<country>/ST=<state>/L=<locality> /O=<organization> /OU=<organizational unit>/CN=<nodename>

      nodename には、データベースをホストするサーバの実際の名前を指定します。次に例を示します。

      openssl req -new -nodes -keyout dbcluster_server.key -out dbcluster_server.csr -subj /C=UK/ST=London/L=London/O=Example /OU=/CN=server1

      キー **dbcluster_server.key** と証明書署名要求ファイル **dbcluster_server.csr** が作成されます。

   b. データベース用の CA 署名付き証明書を生成します。次に例を示します。

      openssl x509 -req -CAcreateserial -in dbcluster_server.csr -CA dbcluster_ca.crt -CAkey dbcluster_ca.key -out dbcluster_ server.crt -days 3650

      証明書 **dbcluster_server.crt** が作成されます。

   c. データベースの秘密キーと証明書要求（.csr）を生成します。データベース クライアントの CommonName（CN）は「postgres」である必要があります。

      openssl req -new -nodes -keyout <keyname>.key -out <certname>.csr -subj /C=<country>/ST=<state>/L=<locality>/O=<organization> /OU=<organizational unit>/CN=postgres

      次に例を示します。

      openssl req -new -nodes -keyout dbcluster_client.key -out dbcluster_client.csr -subj /C=UK/ST=London/L=London/O=Example /OU=/CN=postgres

      キー **dbcluster_client.key** と証明書署名要求ファイル **dbcluster_client.csr** が作成されます。

   d. データベース クライアント用の CA 署名付き証明書を生成します。次に例を示します。

      openssl x509 -req -CAcreateserial -in dbcluster_client.csr -CA dbcluster_ca.crt -CAkey dbcluster_ca.key -out dbcluster_ client.crt -days 3650

      証明書 **dbcluster_client.crt** が作成されます。

3. 4.8 項の手順に従って、データベースの証明書と秘密キーをアップロードして割り当てます。

   a. データベースをホストする各サーバに、次のキーと証明書をアップロードします。

      - データベース クラスタのサーバ証明書（手順 2 で生成）
      - データベース クラスタのサーバ キー（手順 2 で生成）
      - データベース クラスタのクライアント証明書（手順 2 で生成）

- データベース クラスタのクライアント キー（手順 2 で生成）
- データベース クラスタの CA 証明書バンドル（手順 1 で生成）

b. データベースと異なる場所に配置された各 Call Bridge には、次のキーと証明書をアップロードする必要があります。

- データベース クラスタのクライアント証明書（手順 2 で生成）
- データベース クラスタのクライアント キー（手順 2 で生成）
- データベース クラスタの CA 証明書バンドル（手順 1 で生成）

A.4　証明書と秘密キーのペアのインストール

Meeting Server での証明書と秘密キーのペアのインストールについては、第 4 章の手順を参照してください。

付録 B　証明書ファイルおよび秘密キーに使用できる拡張子

次の表に、証明書ファイルと秘密キーに使用できるファイル拡張子を示します。

表 5：証明書ファイルに使用できる拡張子

拡張子	ファイル タイプの情報
.pem	PEM はエンコーディング（ASCII base64）を表し、ファイル拡張子としても使用されます。通常は UNIX ベースの Apache Web サーバからインポートされ、OpenSSL アプリケーションと互換性があります。 PEM 証明書ファイルは自動生成されます。一部のセキュアな Web サイトは、ユーザの ID を認証するために PEM ファイル（電子メールで送信される場合が多い）のアップロードを要求する場合があります。
.der	識別符号化規則（DER）はエンコーディングを表し、ファイル拡張子としても使用されます。DER 形式で作成された証明書のバイナリ表現が含まれています。パブリック暗号方式で X.509 証明書を保存する際によく使用されます。
.cer	Web サイトの信頼性を検証するサードパーティの認証局（VeriSign や Thwate など）によって提供されるセキュリティ ファイルです。サーバでホストされた特定の Web サイトの有効性を確立するために、Web サーバにインストールされます。証明書はバイナリ DER または ASCII（Base64）PEM として符号化されています。
.crt	セキュアな Web サイト（「https://」で始まる）が信頼性を証明するために使用する証明書です。Verisign や Thawte などの企業から配布されます。証明書はバイナリ DER または ASCII（Base64）PEM として符号化されています。 証明書ファイルは、ユーザがセキュアなサイトにアクセスしたときに Web ブラウザで自動的に認識されます。証明書に保存されている情報は、ブラウザウィンドウ内のロック アイコンをクリックすると表示できます。

表 6：秘密キー ファイルに使用できる拡張子

拡張子	ファイル タイプの情報
.key	PKCS#8 の公開キーと秘密キーの両方に使用されます。キーはバイナリ DER または ASCII PEM として符号化されています。
.pem	キーが PEM（ASCII base64）を使用して符号化されていることを示します。
.der	キーがバイナリ DER を使用して符号化されていることを示します。

付録 C　MMP PKI コマンド

表 7：MMP pki コマンドのリスト

コマンド/例	説明/注意事項
pki	現在の PKI の使用状況を表示します。
pki list	PKI ファイル、つまり秘密キー、証明書、および証明書署名要求（CSR）を一覧表示します。
pki inspect <filename>	ファイルを確認し、ファイルが秘密キー、証明書、CSR、または不明のいずれであるかを表示します。証明書の場合はさまざまな詳細情報が表示されます。ファイルに証明書のバンドルが含まれている場合は、バンドルの各要素に関する情報が表示されます。 PEM と DER の両方のファイル形式が対象です。
pki match <key> <certificate>	このコマンドは、システム上の指定したキーと証明書が一致するかどうかを確認します。たとえば、秘密キーと証明書を XMPP などのサービスに使用する場合は、これら 2 つが 1 つの使用可能な ID を構成し、一致している必要があります。
pki verify <cert> <cert bundle/CA cert> **pki verify server.crt bundle.crt**	このコマンドは、信頼チェーンを特定する証明書バンドルを使用して、指定した証明書がルート CA によって署名されているかどうかを確認します。
pki unlock <key>	多くの場合、秘密キーはパスワードで保護された状態で提供されます。Meeting Server で使用するには、キーをロック解除する必要があります。 このコマンドを実行すると、対象ファイルをロック解除するパスワードの入力を求められます。 ロックされた名前が、同じ名前のロック解除されたキーに置き換えられます。
pki csr <key/cert basename> [<attribute>:<value>] **pki csr example CN:www.example.com OU:"My Desk" O:"My Office" L:"Round the corner" ST:California C:US subjectAltName:example.com, example@example.com**	シスコが秘密キー情報の生成に関する要件を満たしていることをユーザが十分に信頼できるように、**pki csr** コマンドを使用して秘密キーおよび関連する証明書署名要求を作成できます。 <key/cert basename> は、新しいキーと CSR を識別する文字列です（たとえば、「new」を指定すると、「new.key」ファイルと「new.csr」ファイルが作成されます）。 CSR の属性は、コロン（「:」）で区切った属性名と値のペアで指定します。複数の単語を含む属性は“”で囲んでください。例：**O: “Example Inc.”** *次のページに続く*

コマンド/例	説明/注意事項
	前のページから続く 属性は次のとおりです。 CN:　証明書に必要な commonName。commonName は、システムの DNS 名である必要があります。 OU:　組織部門 O:　組織 L:　地域 ST:　都道府県 C:　国 電子メール アドレス subjectAltName: 注：複数の単語を含む属性は""で囲んでください。例：**O:"Example Inc."** CSR ファイルは SFTP を使ってダウンロードし、署名のために認証局（CA）に提出することができます。返却時は SFTP 経由でアップロードされます。これを証明書として使用できます。 **pki csr <key/cert basename> [<attribute>:<value>]** で、属性として subjectAltName を使用できます。subjectAltName では、IP アドレスとドメイン名をカンマで区切ったリストがサポートされます。 次に例を示します。 **pki csr test1 CN:example.exampledemo.com subjectAltName:exampledemo.com** **pki csr test1 CN:example.exampledemo.com C:US L:Purcellville O:Example OU:Support ST:Virginia subjectAltName:exampledemo.com** **pki csr test3 CN:example.exampledemo.com C:US L:Purcellville O:Example OU:Support ST:Virginia subjectAltName:exampledemo.com,192.168.1.25,xmpp.exampledemo.com,server.exampledemo.com,join.exampledemo.com,test.exampledemo.com** CN と subjectAltName にワイルドカードを使用することができます。次に例を示します。 **pki csr test4 CN:*.exampledemo.com C:US L:Purcellville O:Example subjectAltName:*,exampledemo.com,exampledemo.com** TLS ハンドシェイクのラウンド トリップ時間が長くならないように、証明書のサイズとチェーン内の証明書の数は最小限に抑えてください。
pki selfsigned <key/cert basename>	簡単なテストやデバッグ用に自己署名証明書を生成できます。 <key/cert basename> は、生成するキーと証明書を識別します。たとえば、「pki selfsigned new」を指定すると、new.key と new.crt（自己署名証明書）が生成されます。

コマンド/例	説明/注意事項
pki pkcs12-to-ssh <username> **pki pkcs12-to-ssh john**	PKCS#12 ファイルに保存されている公開 SSH キーを使用できますが、先に処理する必要があります。このコマンドは、<username>.pub という名前でアップロードされた PKCS#12 ファイルから使用可能な公開キーを抽出します。pkcs#12 ファイルのパスワードを入力するように要求されます。入力後、pkcs#12 ファイルはパスワード保護なしで使用できるキーに置き換えられます。 注：pkcs#12 ファイル内の他のデータは失われます。 john というユーザ用にアップロードされた PKCS#12 ファイル john.pub のキーは、このコマンドを実行すると使用可能になります。

シスコの法的情報

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハード コピーおよびソフト コピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

シスコは世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号は当社の Web サイト（www.cisco.com/go/offices）をご覧ください。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

FCC クラス A 準拠装置に関する記述：この装置はテスト済みであり、FCC ルール Part 15 に規定された仕様のクラス A デジタル装置の制限に準拠していることが確認済みです。これらの制限は、商業環境で装置を使用したときに、干渉を防止する適切な保護を規定しています。この装置は、無線周波エネルギーを生成、使用、または放射する可能性があり、この装置のマ

ニュアルに記載された指示に従って設置および使用しなかった場合、ラジオおよびテレビの受信障害が起こることがあります。住宅地でこの装置を使用すると、干渉を引き起こす可能性があります。その場合には、ユーザ側の負担で干渉防止措置を講じる必要があります。

FCC クラス B 準拠装置に関する記述：この装置はテスト済みであり、FCC ルール Part 15 に規定された仕様のクラス B デジタル装置の制限に準拠していることが確認済みです。これらの制限は、住宅地で使用したときに、干渉を防止する適切な保護を規定しています。本機器は、無線周波数エネルギーを生成、使用、または放射する可能性があり、指示に従って設置および使用しなかった場合、無線通信障害を引き起こす場合があります。ただし、特定の設置条件において干渉が起きないことを保証するものではありません。装置がラジオまたはテレビ受信に干渉する場合には、次の方法で干渉が起きないようにしてください。干渉しているかどうかは、装置の電源のオン/オフによって判断できます。

- 受信アンテナの向きを変えるか、場所を移動します。
- 機器と受信機との距離を離します。
- 受信機と別の回路にあるコンセントに機器を接続します。
- 販売業者またはラジオやテレビの専門技術者に連絡します。

シスコでは、この製品の変更または改造を認めていません。変更または改造した場合には、FCC 認定が無効になり、さらに製品を操作する権限を失うことになります。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved.

Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハード コピーおよびソフト コピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

シスコは世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号は当社の Web サイト（www.cisco.com/go/offices）をご覧ください。