Cisco Meeting Server

Cisco Meeting Server リリース 3.8

証明書のガイドライン

2023 年 9 月 7 日

目次

変更履歴

日付	変更点
2023 年 9 月 7 日	Cisco Meeting Server バージョン 3.8 用に更新。
2023 年 3 月 16 日	Cisco Meeting Server バージョン 3.7 用に更新。
2022 年 8 月 23 日	Cisco Meeting Server バージョン 3.6 用に更新。
2022 年 3 月 1 日	Call Bridge クラスタの検証中に実行される証明書名の検証に関するドキュメントを更新しました
2020 年 12 月 8 日	Cisco Meeting Server バージョン 3.1 用に更新。 証明書ガイドラインの 3 つの展開バリアントすべてが 1 つのガイドに統合されました。
2020 年 9 月 3 日	Cisco Meeting Server バージョン 3.0 用に更新
2020 年 6 月 2 日	マイナー修正。
2020 年 5 月 16 日	Cisco Meeting Server バージョン 2.9 用に更新
2019 年 9 月 30 日	軽微な修正。
2019 年 9 月 27 日	軽微な修正。
2019 年 8 月 12 日	Meeting Server バージョン 2.7 用にリリースされました。

1 はじめに

Cisco Meeting Server ソフトウェアは、シスコ ユニファイド コンピューティング サーバー（UCS）技術に基づく特定のサーバー、または仕様に基づく VM サーバーでホストできます。本書では、Cisco Meeting Server を Meeting Server と呼びます。

注：Cisco Meeting Server ソフトウェアバージョン 3.0 以降では、X シリーズサーバーをサポートしません。

Cisco Meeting Server は非常に安全で、サーバ上で実行されるほとんどのサービスと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 を使用すると、通信する当事者は X.509 証明書と公開キーを交換して相手を認証し、暗号化アルゴリズムを交換して当事者間で送信されるデータを暗号化できます。

この証明書ガイドラインドキュメントでは、拡張可能で復元力のある展開のために証明書を作成およびインストールする方法について説明します。他の展開（つまり、単一分割または単一結合）では違いがある場合、これらの違いは強調されます。

注：このガイドでは、Cisco Meeting Server ソフトウェアを Meeting Server と呼びます。

1.1 本ガイドの使用方法

この章の残りの部分では、Meeting Server 展開全体に証明書を展開するために理解する必要がある概念について説明します。PKI、証明書、および信頼ストアにすでに精通している場合は、これをスキップしてください。

第 2 章では、スケーラブルで回復力のあるサーバモデル内で証明書が必要な場所と、必要な証明書のタイプについて詳しく説明します。

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 へ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、証明書に関連する一般的な問題のトラブルシューティング情報を提供します。

第 6 章では、自己署名証明書をすばやく作成する方法について説明します。

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

付録 B では、証明書ファイルと秘密キーに許可されているファイル名拡張子の概要を述べます。

付録 C では、MMP pki コマンドをリストします。

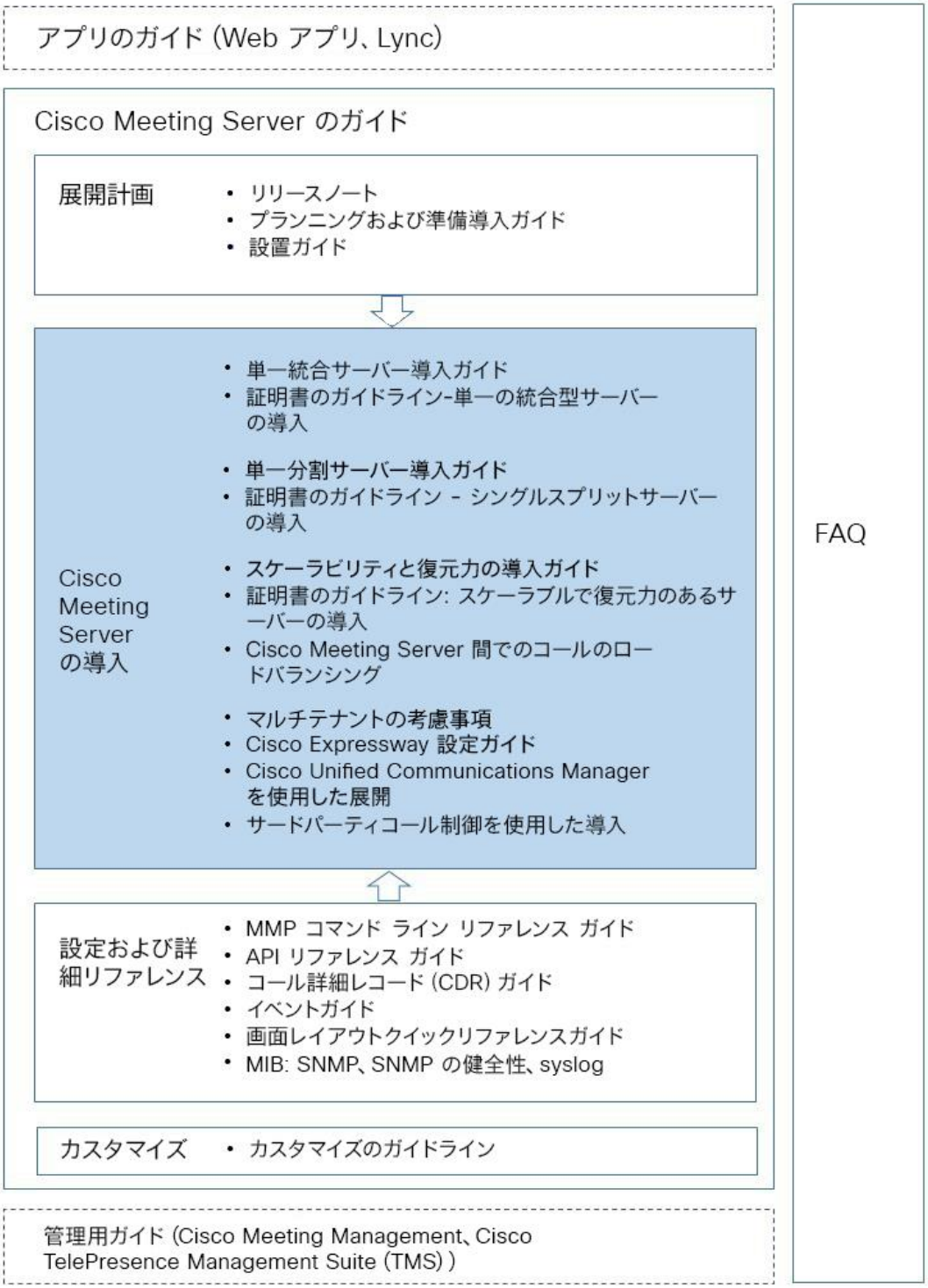
付録 D では、Web Bridge 3 を展開して Cisco Meeting Server Web アプリケーションを使用する場合の証明書と設定情報を提供します。

注：WebRTC 用 Cisco ミーティング アプリケーション（Web Bridge 2）は、Cisco Meeting Server バージョン 3.0 から削除されています。ソフトウェアバージョン 3.0 以降を使用する場合は、WebRTC 用 Cisco ミーティング アプリケーションの代わりに、Cisco Meeting Server Web アプリケーションを使用する必要があります。それには、Web Bridge 3 を展開する必要があります。Web Bridge 3 の展開と設定の詳細については、『導入ガイド（バージョン 3.0 以降）』を参照してください。

重要な情報：バージョン 3.0 から、XMPP サーバ、ロードバランサ、SIP エッジ、および H.323 ゲートウェイコンポーネントが Cisco Meeting Server ソフトウェアから削除されました。さらに、新しい SIP Recorder および Streamer コンポーネントが、サーバソフトウェアから削除された以前の XMPP クライアントバージョンの Recorder および Streamer に置き換わります。TURN サーバはバージョン 3.0 ソフトウェアではそのままであり、ブラウザベースの Cisco Meeting Server Web アプリケーションを Meeting Server 会議に接続するために使用できます。バージョン 3.0 では、ネイティブおよびブラウザベースの Cisco Meeting App クライアントはサポートされません。

このガイドは、Meeting Server のドキュメントセット（図 1 を参照）の一部です。

図 1：Cisco Meeting Server を網羅したガイドの概要



これらのドキュメントは、 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/秘密キーの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 公開キー/秘密キーのペア

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（秘密に保たれなければならないキー）でのみ復号化でき、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あり、公開キーの所有者に関する情報を提供します。通常、証明書が発行されるエンティティの名前、所有者の連絡先の詳細、有効期限（証明書が有効な期間）、および発行者（証明書を発行した機関）が含まれます。証明書は、所有者が本人であることを検証できる信頼できる機関によって署名される必要があります。認証局（CA）は、ネットワーク上の個人、組織、およびコンピュータのアイデンティティを証明する信頼できる機関です。

エンティティが証明書を必要とする場合、最初に公開キー/秘密キーのペアを生成します。次に、エンティティの公開キーとエンティティを識別する情報を含む証明書署名要求（.csr）ファイルを作成します（表 1 を参照）。 エンティティは、秘密キーを使用して .csr ファイルに署名し、.csr ファイルを処理のために CA に送信します。エンティティは、必要な検証のレベルに応じて、.csr ファイルを Verisign などのパブリック CA に送信するか、内部 CA、たとえば、Active Directory 証明書サービスロールがインストールされている Active Directory サーバを使用します。

CA は、.csr ファイルと公開キーを使用して、エンティティのアイデンティティを検証します。検証が成功した場合、CA はエンティティにデジタル ID 証明書を発行します。これは、証明書に記載されている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します。エンティティはデジタル ID 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、公開キーが実際に秘密キーの所有者に属していることを高レベルで保証します。

表 1：.csr ファイル内の情報

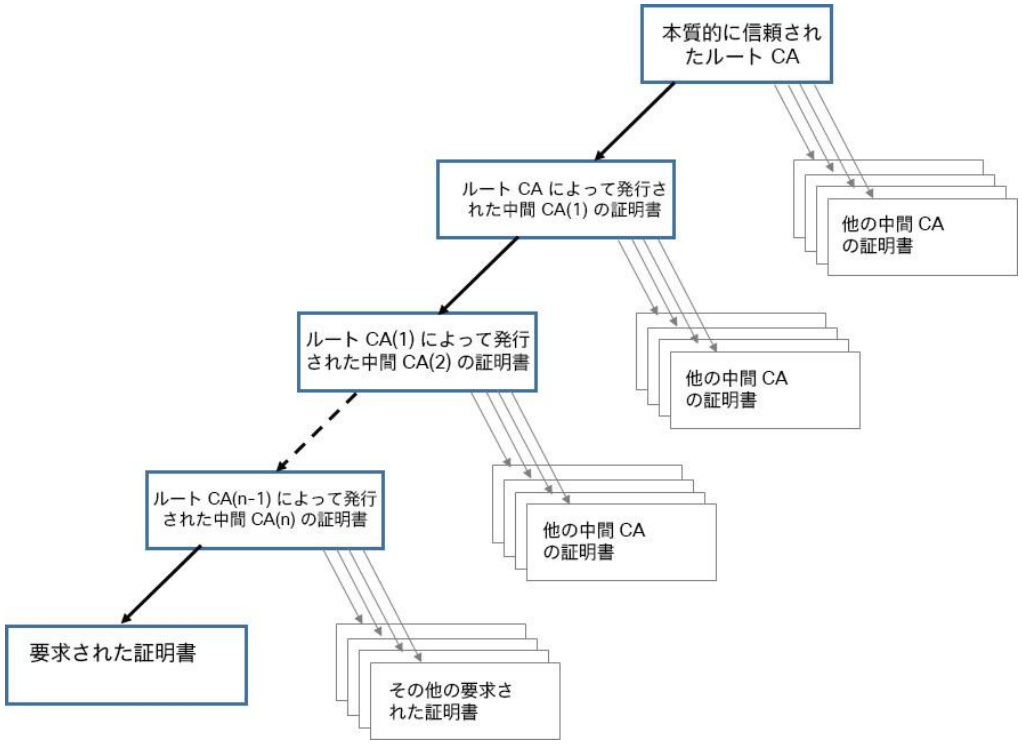
情報	説明
通常名（CN）	これは、保護したい完全修飾ドメイン名です。 例えば、「www.example.com」など。
組織名またはビジネス名（O）	通常、会社の正式な法人名。Ltd.、Inc.、Corp. などのサフィックスを含める必要があります。
組織単位または部署名（OU）	たとえば、Support、IT、Engineering、Finance など。

情報	説明
場所（L）	市区町村名。たとえば、London、Boston、Milan、Berlin など。
州、地域、郡または州（ST）	たとえば、ニュージャージー州バッキンガムシャー。省略しないでください。
国（C）	組織の所在地の国を表す 2 文字の ISO コード。たとえば、US、GB、FR などです。
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。通常は、証明書管理者または IT 部門の電子メールアドレスです。
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、証明書が一致する必要がある複数の名前を指定できます。subjectAltName（SAN）には、電子メールアドレス、IP アドレス、通常の DNS ホスト名などを含めることができます。

1.2.3 信頼の連鎖

あるエンティティが別のエンティティから認証用の証明書を提供するように要求（チャレンジ）された場合、そのエンティティは、自分自身の証明書と、チャレンジ側当事者が信頼する認証局（通常はルート認証局と呼ばれる）へのリンクを確立する一連の他の証明書を提示する必要があります。エンティティの証明書をルート CA にリンクするこの証明書の階層は、「信頼の連鎖」と呼ばれます。ルート CA が別の認証局（中間 CA と呼ばれる）の証明書に署名し、その認証局がエンティティの証明書に署名することはよくあります。その場合、エンティティは、自身の証明書と、ルート CA によって発行されたこの中間 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、信頼できるルート CA へのリンクを確立していない場合、チャレンジ側当事者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をルート CA にリンクする一連の証明書は、中間 CA に発行されるため、「中間証明書」と呼ばれます。

図 2：証明書の信頼の連鎖



接続デバイスが信頼の連鎖を検証できるようにするために、すべての証明書には「発行先」と「発行者」という 2 つのフィールドが含まれています。中間 CA は、この 2 つのフィールドに異なる情報を表示し、必要に応じて、信頼を確立するためにチェックを続行する場所を接続デバイスに示します。ルート CA 証明書は「発行先」および「発行者」自体であるため、これ以上のチェックはできません。

たとえば、エンティティ A（Web サーバ www.example.com）がエンティティ B（Web クライアント）によって認証を要求された場合、エンティティ A はその証明書と証明書チェーンをエンティティ B に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：**example.com、**発行者：中間** CA 1
証明書 2 - **発行先：中間** CA 1、**発行者：中間** CA 2
証明書 3 - **発行先：中間** CA 2、**発行者：ルート** CA

エンティティ B の信頼ストアにルート CA の証明書を提供すると、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を確立できます。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。エンティティ A だけが秘密キーにアクセスできるため、エンティティ A だけが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」と呼ばれることがあります。

1.2.4 証明書バンドル

証明書バンドルは、ルート CA の証明書とチェーン内のすべての中間証明書のコピーを保持する 1 つのファイル（拡張子が .pem、.cer、または .crt）です。証明書は、証明書バンドルの最後にあるルート CA の証明書と連続している必要があります。外部クライアント（Web ブラウザなど）では、セキュアな接続を設定するときに、証明書と証明書バンドルが Web Bridge 3 によって提示される必要があります。Call Bridge が SIP ピアへの TLS トランクを確立する場合、Call Bridge はその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。

CSV データファイルは、メモ帳などのテキストエディタを使用して作成できます。-----BEGIN CERTIFICATE----- および -----END CERTIFICATE----- タグを含むすべての文字をドキュメントに挿入する必要があります。証明書の間にスペースを入れないでください。たとえば、証明書 1 の -----END CERTIFICATE と証明書 2 の -----BEGIN CERTIFICATE----- の間にスペースや余分な行を入れないでください。証明書 1 は ----------END CERTIFICATE----- で終わり、次の行には証明書 2 の -----BEGIN CERTIFICATE が含まれます。ファイルの最後には、1 行追加する必要があります。.pem、.cer、または .crt の拡張子を付けてファイルを保存します。

注：Web Bridge 3 では、すべての証明書定義で証明書のバンドル、つまり完全なチェーンファイルを使用する必要があります。これは、Web Bridge 2 の証明書の実装とは異なります。

1.2.5 信頼ストア

Web ブラウザやその他のクライアントは、信頼できる署名機関のリストを保持しているため、「信頼の連鎖」によって信頼できるサーバのリストを保持します。これらの信頼できる CA は、クライアントの「信頼ストア」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、失効リスト内のエンティティをストアから削除します。

接続するクライアント（またはデバイス）が証明書を信頼するために、クライアントは、証明書の CA がクライアントの信頼ストアに保持されているかどうかを確認します。証明書が信頼できる CA によって発行されていない場合、接続するクライアントは、発行 CA の証明書が信頼できる CA によって発行されたかどうかを確認します。これは、信頼できる CA が見つかるか、信頼できる CA がなくなるまで、チェーンを上って繰り返されます。信頼できる CA が見つかった場合、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 が見つからない場合、接続するクライアントは通常、エラーメッセージを表示します。

バージョン 3.0 では、Web Bridge 3 での C2W 接続用の証明書が変更され、2.9 では必要であった、信頼ストアでルート証明書が不要になりました。これにより、管理者は、どの証明書を信頼するかをより柔軟に選択できます。たとえば、管理者が社内のポリシーに従って C2W 接続を保護するために公開証明書を使用する必要がある場合でも、その公開 CA によって署名されたすべての証明書を信頼することなく、接続先で使用されるクライアントまたはサーバの C2W 証明書だけを信頼することができます。これは、証明書のピンニングと呼ばれます。

2 展開に必要な証明書

セクション 2.1 では、展開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と、必要な証明書のタイプ（パブリック CA または内部 CA によって署名されている）について説明しています。

Cisco Meeting Server 3.0 **ソフトウェアから削除されたコンポーネントに関する注意：**次のコンポーネントがソフトウェアバージョン 3.0 から削除されました。Web Bridge 2、H.323 ゲートウェイ、SIP Edge、XMPP サーバ、およびロードバランサ。

2.1 パブリックまたは内部 CA によって署名された証明書

外部デバイスに接続する Meeting Server 上のアプリケーションは、外部デバイスから信頼される必要があり、パブリック CA によって署名された証明書を必要とします。Meeting Server 内で内部的にインターフェイスするアプリケーションは、パブリック CA または内部 CA によって署名された証明書を使用できます。内部 CA によって署名された証明書は、ローカルまたは組織の認証局によって生成できます。たとえば、Active Directory 証明書サービスロールがインストールされている Active Directory サーバなどです。セクション 3 を参照してください。

パブリック CA によって署名された証明書を必要とするアプリケーションを表 2 に示します。内部 CA によって署名された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その他の証明書関連の FAQ については、このリンクにアクセスしてください。

注：WebRTC コールを使用する展開では、Call Bridge 証明書に、digitalSignature ビットが設定された KeyUsage 拡張を含める必要があります。そうしないと、Call Bridge と Web アプリクライアント間のメディアの DTLS ネゴシエーションが失敗する可能性があります。この KeyUsage は、通常、クライアントおよび/またはサーバ証明書の CA 設定を使用するときに含まれます。

表 2：パブリック CA によって署名された証明書（スケーラブルで回復力のあるサーバモデル）

パブリック CA によって署名された証明書を必要とするアプリケーション	証明書の使用	理由
Web Bridge 3（Web アプリを使用する場合のみ）		Web ブラウザには、パブリック CA によって署名された証明書が必要です
Call Bridge（Meeting Server が直接 Lync フェデレーションのためにパブリックネットワークに接続されている場合のみ）	callbridge - TLS Web サーバ認証、TLS Web クライアント認証	直接フェデレーションを行う場合、Lync Edge サーバは、Call Bridge からのパブリック CA によって署名された証明書を必要とします。
TURN サーバー	TURN - TLS Web サーバー認証	TURN サーバで TLS を構成する場合、TURN サーバは、WebRTC クライアントが接続を信頼するように、Web Bridge 用に作成されたものと同様の証明書/キーペアを必要とします。証明書は、Web Bridge で使用されるものと同じ CA によって署名される必要があります。

表 3：内部 CA によって署名された証明書（スケーラブルで回復力のあるサーバモデル）

内部 CA によって署名された証明書を使用できるアプリケーション	証明書の使用	理由
Web 管理	WebAdmin - TLS Web サーバー認証	Meeting Server は Web Admin のインターフェイスへの HTTPS 接続のみを許可するため、Web Admin には証明書が必要です。 注：Meeting Server API は Web Admin のインターフェイスを介してルーティングされるため、Web Admin インターフェイスではなく API を介して Call Bridge を設定する場合でも証明書が必要です。 さらに、クラスタ内の Call Bridge は、Web Admin を介して HTTPS 経由で相互に接続します。バージョン 2.4 から、Call Bridge 信頼ストアを使用してクラスタ内の Call Bridge を検証することにより、Call Bridge クラスタのセキュリティを向上させることができます。 セクション 4.9 を参照してください。

内部 CA によって署名された証明書を使用できるアプリケーション	証明書の使用	理由
Call Bridge	TLS Web クライアント認証と TLS Web サーバー認証	Web Bridge 3 の c2w 接続では、Call Bridge からの証明書が必要であり、信頼する必要があります。 Active Directory サーバも、Call Bridge からの証明書を信頼する必要があります。 さらに、展開に TLS を使用する SIP トランクがある場合、Call Bridge には、SIP 呼制御デバイスとの相互認証のための証明書が必要です。
データベース クライアント	データベース - TLS Web クライアント認証	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。セクション 3.1.4 を参照してください。
データベースサーバー	データベース - TLS Web サーバ認証	
レコーダー	レコーダー - TLS Web サーバ認証	Meeting Server でレコーダーを有効にする場合、Call Bridge はレコーダーからの署名付き証明書を必要とし、レコーダーは Call Bridge からの証明書を必要とし、信頼する必要があります。
ストリーマ	ストリーマ - TLS Web サーバ認証	Meeting Server でストリーマを有効にする場合、Call Bridge はストリーマからの署名付き証明書を必要とし、ストリーマは Call Bridge からの証明書を必要とし、信頼する必要があります。
c2w	webbridge 3 c2w - TLS Web サーバー認証	C2W 証明書は、Call Bridge と Web Bridge 3 の間の接続に使用されます。

注：証明書が CA によって署名されるときに、証明書のプロパティを指定するのに役立つように、証明書の用途が記載されます。ExtendedKeyUsages が有効になっていない場合、証明書はすべての用途に対して有効です。いずれかが有効になっている場合、証明書の ExtendedKeyUsages には、少なくとも表で定義されている用途が含まれている必要があります。

注：Call Bridge 証明書の KeyUsage ビットには、KeyUsage digitalSigning が含まれている必要があり、さもないと、Call Bridge と Web アプリクライアント間のメディアの DTLS ネゴシエーションが失敗する可能性があります。この KeyUsage は、通常、クライアントおよび/またはサーバ証明書の CA 設定を使用するときに含まれます。

2.2 Meeting Server が証明書の拡張を検証する方法

Meeting Server は、以下に説明するように証明書の拡張を検証します。

クライアント接続の場合：

- ExtendedKeyUsage 拡張が存在する場合、TLS Web クライアント認証ビットがセットされている必要があります。
- KeyUsage 拡張が存在する場合、digitalSignature および keyAgreement ビットの少なくとも 1 つがセットされている必要があります。
- Netscape 拡張機能が存在する場合は、SSL クライアントビットがセットされている必要があります。

サーバー接続の場合：

- ExtendedKeyUsage 拡張が存在する場合、TLS Web サーバ認証ビットがセットされている必要があります。
- KeyUsage 拡張が存在する場合、digitalSignature、keyEncipherment、および keyAgreement ビットの少なくとも 1 つがセットされている必要があります。
- Netscape 拡張機能が存在する場合、SSL サーバビットがセットされている必要があります。

さらに、証明書のチェーンを検証する場合、リーフ証明書を除くすべての証明書は、次のオプションの少なくとも 1 つを使用して CA としてマークされている必要があります。

- KeyUsage 拡張が存在する場合、keyCertSign ビットがセットされている必要があります。
- 基本制約拡張が存在する場合、CA ビットがセットされている必要があります。
- Netscape 拡張機能が存在する場合、SSL CA ビットがセットされている必要があります。

3 証明書の取得

セクション 2.1 では、展開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と、必要な証明書のタイプ（パブリック CA または内部 CA によって署名されている）について説明しています。この章では、さまざまな種類の証明書を取得する方法に焦点を当て、第 4 章ではそれらをインストールする場所について説明します。

注：Lync 展開を Meeting Server に接続する場合は、Lync Front End サーバによって信頼されているのと同じ認証局（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 間の統合のサポートについては、Lync アドバイザーにお問い合わせください。

すべての証明書は、3 ステップのプロセスに従う必要があります。

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：公開キーは .csr ファイル内に作成され、保持されます。

   ---

2. 署名のために .csr ファイルを CA（パブリック CA または内部 CA）に送信します。
3. CA からの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。

この章の残りの部分では、ステップ 1 と 2 の例を示します。第 4 章では、ステップ 3 について説明します。

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については、セクション 6 で説明します。これらは、ラボで設定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）によって署名された証明書を使用することをお勧めします。

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 が証明書署名要求を作成するときには、CA の現在の運営ルールに従って、SHA256 を使用して署名されます。

3.1 秘密キーと証明書署名要求（.csr ファイル）の生成

このセクションでは、Meeting Server MMP pkiコマンドを使用して公開キーと .csr ファイルを作成する方法について説明します。サードパーティ製ツールを使用してこれを行う場合は、サードパーティの指示に従って、このガイドの セクション 3.2 から再開してください。OpenSSL を使用して秘密キーと .csr ファイルを作成する場合は、付録 A の手順の概要に従ってください。

pki csr<key/cert basename> コマンドを使用して、次の 2 つのファイルを生成します。private key <basename>.key および certificate signing request file <basename>.csr。これらのファイルはSFTP を使用して、Meeting Server からすぐに取得できます。

注：basename には「.」または「_」を含めないでください。たとえば、 pki csr basenameは有効ですが、**pki csr base.name**または**pki csr base_name**は許可されません。

秘密キーと証明書署名要求ファイルを生成するには：

1. MMP にログインします。
2. 次のシンタックスを使用して **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>]

   ここで

   <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、または下線文字のみを含めることができます。

   CN、OU、O、ST、C、subjectAltNameは、表 4 で説明されています。オプションのマークが付いているものは、ローカル CA による署名のために pki csr コマンドを使用して証明書要求ファイルを作成している場合は省略できます。公的認証局が署名す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お勧めします。

表 4：.csr ファイルの属性

属性		説明	オプション/必須
CN	共通名	これは、ドメインネームシステム（DNS）で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完全修飾ドメイン名は webBridge1.example.com です。FQDN は、コンポーネントを他のドメインの webBridge1 と呼ばれる他のコンポーネントから一意に区別します。	必須、以下の注記を参照
O	組織名またはビジネス名	通常、会社の正式な法人名。Ltd.、Inc.、Corp. などのサフィックスを含める必要があります。たとえば、“Example Inc.” のように複数の単語を使用する場合は、属性の前後に “” を使用します。	オプション
OU	組織単位または部署名	たとえば、Support、IT、Engineering、Finance など。“Human Resources” など、複数の単語がある場合は属性の前後に “” を使用します。	オプション
L	位置	市区町村名。たとえば、London、Boston、Milan、Berlin など。	オプション
ST	州、地域、郡または州	たとえば、Buckinghamshire、California など。省略しないでください。“New Jersey” など、複数の単語がある場合は属性を “” で囲みます。	オプション
C	国	組織の所在地の国を表す 2 文字の ISO コード。たとえば、US、GB、FR などです。	オプション
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。通常は、証明書管理者または IT 部門の電子メールアドレスです。	オプション
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、証明書が一致する必要がある複数の名前を指定できます。このオプションのフィールドにより、生成された証明書が複数のドメインをカバーできるようになります。IP アドレス、ドメイン名、電子メールアドレス、通常の DNS ホスト名などをコンマで区切って含めることができます。このリストを指定する場合は、このリストに CN も含める必要があります。	必須、または単一の証明書を複数のコンポーネントで使用する場合。下記の注を参照

注意点：

- Web Bridge 3 に専用の証明書を使用する予定の場合は、Web Bridge3 の DNS A レコードで定義されている FQDN を CN フィールドで指定します。FQDN を指定しなかった場合は、ブラウザ証明書のエラーが発生します。
- Web Bridge 3、Call Bridge、TURN サーバ、Web Admin、Reorder、Streamer など、複数のコンポーネントで同じ証明書を使用する予定の場合は、CN フィールドでドメイン名（DN）を指定し、SAN フィールドで、証明書を使用する各コンポーネントのドメイン名（DN）と FQDN を指定します。
- SAN フィールドで、「,」区切り文字とリスト内の項目の間にスペースがないことを確認してください。

例：

CN=**example.com**

SAN=
callbridge1.example.com,callbridge2.example.com,callbridge3.example.com,webbridge3.example.com,example.com

pki csr コマンドを使用する場合：

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

コマンドは次のとおりです。

```
pki csr onecert CN:example.com subjectAltName:callbridge1.example.com,callbridge2.example.com,callbridge3.example.com,webbridge3.example.com
```

注：pkiコマンドを使用すると、CN が SAN リストに自動的に追加されます。上の例に示すように、SAN リストに CN をリストしないでください。

3.1.1 Call Bridge の CSR

展開内での各 Call Bridge の使用方法によっては、秘密キーと証明書のペアが必要になる場合があります。

- Web Bridge 3 との通信を確立するため。展開のセキュリティにとって、信頼できる Call Bridge からのみ設定を受け入れることが重要です。
- SIP 呼制御デバイスとの TLS 接続を確立するため。

- Lync フロントエンド（FE）サーバとの TLS 接続を確立するため。証明書が Lync FE サーバによって信頼されるようにするため。
  - 単一分割展開および単一結合展開の場合：証明書の **CN**は、Lync FE サーバ上の信頼できるアプリケーションおよび静的ルートとして Meeting Server を設定するときに追加された FQDN と同じである必要があります。
  - スケーラブルで回復力のある展開の場合：証明書の **CN**は、Lync FE サーバで信頼できるアプリケーションとして Meeting Server を設定するときに追加された FQDN と同じである必要があります。この値を表示するには、Lync Powershell コマンド **get-cstrustedapplicationcomputer** を使用します。
  - 証明書に **subjectAltName** リストがある場合は、FQDN もリストに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。

例：

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

または

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

この例では、callbridge.key と callbridge.csr という 2 つのファイルが生成されます。これらのファイルは、SFTP を使用して Meeting Server から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 に送信します。セクション 3.2 を参照してください。

セクション 4.4 では、Call Bridge の証明書のアップロードについて詳しく説明します。

3.1.2 Web Bridge 3 の CSR

Web ブラウザは CNフィールドを見て、Web Bridge 3 の FQDN を決定します。Web ブラウザの証明書エラーを回避するには、次のアドバイスに従ってください。

- Web Bridge 3 の専用証明書を使用している場合：**CN**フィールドで、Web Bridge 3 の DNS A レコードで定義されている FQDN を指定します。FQDN を指定しなかった場合は、ブラウザ証明書のエラーが発生します。 **subjectAltName** フィールドが使用されている場合、 **CN** フィールドで指定された FQDN は、`subjectAltName` フィールドに自動的に追加されない場合は、 `subjectAltName` フィールドに含める必要があります。注：**SAN**リストが存在する場合、 **pki csr**は CN を **SAN**リストに自動的に追加します。

注：展開環境で Cisco Expressway Web プロキシが Web Bridge に接続する必要がある場合、Web Bridge 証明書の SAN フィールドに、Web Bridge に接続する Expressway-C で使用される A レコードが含まれていることを確認します。含まれていない場合、接続は失敗します。たとえば、Expressway が join.example.com の Web Bridge に接続するように設定されている場合、この FQDN の A レコードが存在する必要があります。また、Web Bridge 証明書の SAN フィールドに join.example.com が含まれている必要があります。

- 複数のコンポーネント（Web Bridge 3、Call Bridge、および TURN サーバ）で同じ証明書を使用する予定の場合：証明書を使用するコンポーネントごとに、 CN フィールドでドメイン名（DN）を指定し、 SANフィールドでドメイン名（DN）と FQDN を指定します。

例：

```
pki csr webbridge3 CN:www.example.com O:"Example Inc."
```

または

```
pki csr webbridge3 CN:www.example.com O:"Example Inc."
subjectAltName:guest.example.com
```

この例では、webbridge3.key と webbridge3.csr という 2 つのファイルが生成されます。これらのファイルは、SFTP を使用して Meeting Server から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 に送信します。セクション 3.2 を参照してください。

セクション 4.5 では、Web Bridge 3 の証明書のアップロードについて詳しく説明します。

3.1.3 MeetingApps の CSR

Web ブラウザは MeetingApp と直接通信するため、ブラウザの信頼できる CA 証明書を使用する必要がありました。内部 CA によって署名された証明書を使用する予定の場合は、Web アプリに使用する各ブラウザで証明書を検証する必要があります。

注：内部 CA によって署名された証明書を使用している場合、ファイル共有機能はすべてのブラウザおよびすべてのオペレーティングシステムで動作しない可能性があります。

MeetingApps 専用の証明書を使用することをお勧めします。Web ブラウザは CN フィールドを調べて、MeetingApps の FQDN を判別します。CN フィールドで、MeetingApps の DNS A レコードで定義されている FQDN を指定します。FQDN を指定しなかった場合は、ブラウザ証明書のエラーが発生します。**subjectAltName** フィールドが使用されている場合、**CN** フィールドで指定された FQDN は、subjectAltName フィールドに自動的に追加されない場合は、**subjectAltName**フィールドに含める必要があります。

注： SANリストが存在する場合、**pki csr**は **CN**を **SAN**リストに自動的に追加します。

例：

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

または

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

この例では、meetingapps.key と meetinapps.csr という 2 つのファイルが生成されます。これらのファイルは、SFTP を使用して Meeting Server から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 に送信します。セクション 3.2 を参照してください。

セクション 4 では、MeetingApps の証明書のアップロードについて詳しく説明します。

3.1.4 データベースクラスタリングの CSR

注：このセクションは、スケーラブルで回復力のある展開にのみ適用されます。

バージョン 2.7 以降、データベースクラスタでは、クラスタ内のデータベースに保持または接続する各 Meeting Server に設定された CA と同じ CA によって署名されたクライアント証明書およびサーバー証明書が必要になります。証明書の使用を必須にすることで、クラスタ全体の機密性と認証の両方が確保されます。

2.7 以降、MMP からすべての証明書を直接生成できるようになりました。データベースクラスタの証明書と秘密キーを SFTP 経由でダウンロードしてから、同じデータベースクラスタに属する各 Meeting Server にアップロードする必要があります。

注意：クラスタを構成するデータベースノードは、正規のノードのみがクラスタに接続できるように、信頼できるルート CA 証明書を使用して設定する必要があります。ノードは、信頼されたルート証明書で終わる証明書チェーンを提供する接続を信頼するようになります。したがって、各データベースクラスタは専用のルート証明書を使用する必要があります。ルート証明書や中間証明書を他の目的で使用することはできません。

引き続き、openssl など、他の方法を使用して証明書を作成することもできます。詳細については、 付録 A を参照してください。

データベースクラスタリングの場合：

1. pki selfsigned コマンドを使用して、dbca 自己署名証明書を作成します。

   以下にその例を示します。

   ```
   pki selfsigned dbca CN:"My company CA"
   ```

   dbca.keyという名前のローカル秘密キーと、dbca.crt に共通名 **CN=My Company CA**を使用した自己署名証明書を作成します。

2. 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 フィールドでいずれかのサーバの FQDN を指定し、SAN フィールドで他のサーバの FQDN を指定します。「拡張キー使用法」を使用している場合は、データベースサーバーに対して「サーバー認証」が許可されていることを確認します。

   例：

   ```
   pki csr dbserver CN:server.db.example.com
   subjectAltName:server02.db.example.com
   ```

   dbserver.csr という名前の CSR ファイルと dbcserver.key という名前の秘密キーを生成します。

3. 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。「拡張キー使用法」を使用している場合は、データベースクライアントに対して「クライアント認証」が許可されていることを確認します。

   例：

   ```
   pki csr dbclient CN:postgres
   ```

   dbclient.csr という名前の CSR ファイルと dbclient.key という名前の秘密キーを生成します。

4. 内部 CA を使用して、dbserver.csr および dbclient.csr 証明書署名要求ファイルに署名し、これらに対応する dbserver.crt および dbclient.crt 証明書と内部 CA 証明書（バンドル）を取得します。

   例：

   ```
   pki sign dbserver dbca
   pki sign dbclient dbca
   ```

5. 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 フィールドでいずれかのサーバの FQDN を指定し、SAN フィールドで他のサーバの FQDN を指定します。「拡張キー使用法」を使用している場合は、データベースサーバーに対して「サーバー認証」が許可されていることを確認します。

   例：
   pki csr db01server CN:www.example.com
   db01server.csr という名前の CSR ファイルと db01server.key という名前の秘密キーを生成します。

6. 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。「拡張キー使用法」を使用している場合は、データベースクライアントに対して「クライアント認証」が許可されていることを確認します。

   例：
   pki csr db01client CN:postgres
   db01client.csr という名前の CSR ファイルと db01client.key という名前の秘密キーを生成します。

7. 内部 CA を使用して、db01server.csr および db01client.csr 証明書署名要求ファイルに署名し、これらに対応する db01server.crt および db01client.crt 証明書と内部 CA 証明書（バンドル）を取得します。セクション 3.3 を参照してください。

セクション 4.4 では、データベースクラスタリング用の証明書のアップロードについて詳しく説明します。

3.1.5 TURN サーバの CSR

TURN サーバで TLS を使用する予定の場合、WebRTC クライアントが接続を信頼できるように、TURN サーバは Web Bridge 3 用に作成されたものと同様の証明書/キーペアを必要とします。証明書は、Web Bridge 3 の証明書で使用されるものと同じ認証局によって署名される必要があります。

例：
pki csr turnserver CN:www.example.com O:"Example Inc."
この例では、turn.key と turn.csr という 2 つのファイルが生成されます。これらのファイルは、SFTP を使用して Meeting Server から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 に送信します。セクション 3.2 を参照してください。

セクション 4.6 では、TURN サーバの証明書のアップロードについて詳しく説明します。

3.2 パブリック認証局を使用した CSR への署名

Meeting Server に必要なパブリック CA 署名付き証明書のリストについては、セクション 2.1 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 ファイルを優先する認証局（Verisign など）に送信します。CA はあなたの身元を確認し、Meeting Server とその使用要件に必要なパブリック CA 署名付き証明書のリストに対して署名付き証明書を発行します。証明書ファイルには、.pem、.cer、または .crt の拡張子が付けられます。 付録 B に、証明書ファイルに使用されるファイル拡張子の簡単な概要を示します。

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 が証明書チェーンを発行した場合は、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 行と 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 拡張子を付けて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ように順に並べる必要があります。

Meeting Server に署名付き証明書と秘密キーをインストールする方法については、第 4 章を参照してください。

注：証明書を展開する前に、**`pki inspect`**コマンドまたは openssl ツールを使用して、すべての証明書に正しい CN、SAN、KeyUsage、および ExtendedKeyUsage の値があることを確認することをお勧めします。

3.3 内部認証局を使用して CSR に署名する

Meeting Server とその使用要件に必要な内部 CA 署名付き証明書のリストについては、セクション 2.1 を参照してください。

注：Meeting Server が Cisco Unified Communications Manager にトランクされている展開では、Cisco Unified Communications Manager は、TLS Web クライアント認証と TLS Web サーバ認証の両方を含む拡張キー使用を許可するテンプレートを使用して、Call Bridge 証明書に署名する必要があります。Microsoft Active Directory 証明書サービスは、このタイプの証明書を発行できます。

注：証明書を展開する前に、**`pki inspect`**コマンドまたは openssl ツールを使用して、すべての証明書に正しい CN、SAN、KeyUsage、および ExtendedKeyUsage の値があることを確認することをお勧めします。

このセクションは、Microsoft Active Directory を内部 CA として使用している場合に適用されます。別の内部 CA を使用している場合は、対応する指示に従って、このガイドの第 4 章から再開してください。

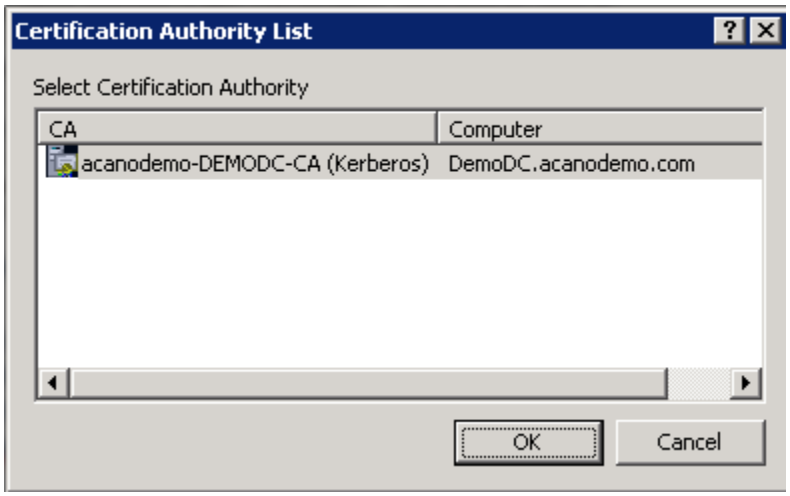
内部 CA 署名付き証明書を取得するには、次の手順に従います。

1. 生成された .csr ファイルを CA（たとえば、Active Directory 証明書サービスロールがインストールされている Active Directory サーバ）に転送します。
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、パスと CSR ファイル名をお客様の情報に置き換えて発行します。

   `certreq -submit -attrib "CertificateTemplate:Computer" <path\csrfilename>`

   例：

   `certreq -submit -attrib "CertificateTemplate:Computer" 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。正しい CA を選択して、[OK] をクリックします。



証明書発行許可を持つ Windows アカウントを使用している場合は、生成された証明書を保存するよう求めるプロンプトが表示されます。.crt、.cer、または .pem の拡張子を付けてファイルを保存します（例：example.crt）。ステップ 4 に進みます。証明書ファイル拡張子の簡単な概要については、付録 B を参照してください。

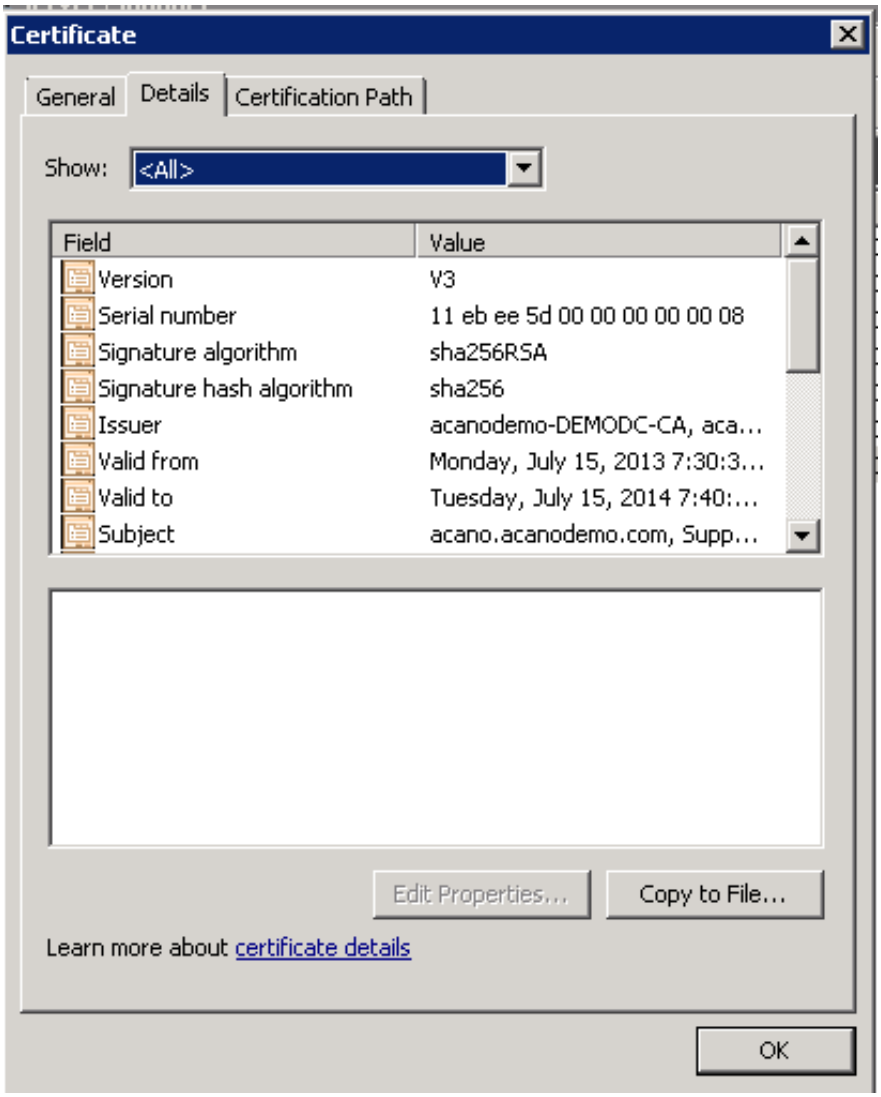
生成された証明書を発行するためのプロンプトが表示されず、代わりに次のようにコマンド プロンプト ウィンドウに「**証明書の要求は保留中です：提出済みです**（Certificate request is pending: taken under submission）」というメッセージが表示された場合は、要求 ID（Request ID）をメモしておきます。

以下のステップに従って、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（CA Role）] 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リクエスト ID に一致する保留中のリクエストを右クリックして、[**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 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（Certificate Export Wizard）] が起動します。

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、たとえば、
`callbridge` などの名前を入力して、[**次へ**（Next）] をクリックします。

g. .crt、.cer、または .pem の拡張子を付けてファイルを保存します（例：
`callbridge.crt`）。

4. Meeting Server に署名付き証明書と秘密キーをインストールする方法については、第 4 章を参照してください。

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

スケーラブルで回復力のある Meeting Server の展開には、次のパブリック CA 署名付き証明書が必要です。

- セキュア通信のために TLS 接続を使用する場合は、TURN **サーバー**。
- パブリックネットワークを介した直接の Lync フェデレーションが必要な場合は、Call Bridge。Lync Edge サーバは、接続を信頼するために、Call Bridge からのパブリック CA 署名付き証明書を必要とします。
- Web Bridge 3 では、Web アプリケーションを使用するために Web Bridge 3 を展開する場合、接続を信頼するために、Call Bridge からのパブリック CA 署名付き証明書が必要です。

また、以下のパブリックまたは内部 CA 署名付き証明書：

- Web Admin。Meeting Server API は Web Admin のインターフェイスを介してルーティングされるため、Web Admin インターフェイスではなく API を介して Call Bridge を設定する場合でも証明書が必要です。

  注：このガイドでは、『Meeting Server 設置ガイド』で説明されているように、Web Admin インターフェイスの秘密キー/証明書のペアがすでに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。

- Call Bridge。Web Bridge 3 の c2w 接続では、Call Bridge からの証明書が必要であり、信頼する必要があります。Active Directory サーバも、Call Bridge からの証明書を必要とします。さらに、展開に SIP トランクがある場合、Call Bridge には、SIP 呼制御デバイスとの相互認証のための証明書が必要です。
- **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）は、同じ認証局によって署名された秘密キーと証明書を必要とします。（スケーラブルで復元力のある展開のみ。）
- Meeting Server で Recorder を有効にする場合、Call Bridge は Recorder からの証明書を必要とし、Recorder は Call Bridge からの証明書を必要とし、信頼する必要があります。証明書のアップロードと Recorder の設定の詳細については、『Cisco Meeting Server のスケーラブルで復元力のある導入ガイド』の「Recorder」セクションを参照してください。

- Meeting Server で Streamer を有効にする場合、Call Bridge は Streamer からの証明書を必要とし、Streamer は Call Bridge からの証明書を必要とし、信頼する必要があります。証明書のアップロードと Streamer の設定の詳細については、『Cisco Meeting Server のスケーラブルで復元力のある導入ガイド』の「Streamer」セクションを参照してください。
- Web Bridge 3 を有効にした場合、C2W 証明書は、Call Bridge と Web Bridge 3 の間の接続に使用されます。Call Bridge が Web Bridge 3 との C2W 接続を確立するには、証明書を検証する C2W 信頼ストアを指定する必要があります。
- MeetingApps に使用される証明書は、MMP コマンドを使用して MeetingApps を設定するときに割り当てる必要があります。MeetingApps に内部 CA 署名付き証明書を使用している場合は、すべてのブラウザで MeetingApps アドレスに接続して、証明書を検証する必要があります。

4.1 秘密キーと証明書の再利用

証明書のインストールごとに異なる秘密キー/証明書のペアを用意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、複数のサービスに再利用できます。秘密キー/証明書のペアを再利用する場合のアドバイスは次のとおりです。

- Lync 展開を Meeting Server に接続している場合は、Lync 展開によって信頼されている認証局（CA）を使用することをお勧めします。
- 証明書と秘密キーには、それらが使用される場所を反映するファイル名を使用します（例：**`webadmin.crt`**および **`webadmin.key`**）。

4.2 MMP への秘密キーと証明書のアップロード

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：**`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`** は、MMP にアップロードされた SSH キーと CSR ファイルも一覧表示します。

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を除き、「.」を含めないでください。 たとえば、callbridge.key は有効ですが、call.bridge.key は許可されません。

4.3 ファイルの種類を検査し、証明書と秘密キーが一致することを確認する

Meeting Server に秘密キー/証明書のペアをインストールする前に、インストールする正しいファイルがあることを確認してください。このセクションでは、MMP コマンド（pki inspect、pki match、pki verify）を使用して、インストールする予定のファイルの ID を確認する方法について簡単に説明します。

ファイルを検査して、ファイルがまだ有効かどうか（有効期限）を判断するには：

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：

pki match <keyfile> <certificatefile>

証明書が CA によって署名されていることと、証明書バンドルを使用してこれを表明できることを確認するには：

pki verify <cert> <certbundle/CAcert>

例：

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 callbridge.crt

   たとえば、証明書がまだ有効かどうかを確認するために、ファイルの内容を検査します。
3. 次のコマンドを入力します。

   pki match callbridge.key callbridge.crt

   ファイル **callbridge.key**がファイル **callbridge.crt**に一致し、それらが一緒になって 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 callbridge.crt callbridgebundle.crt

   callbridge.crt が信頼できる CA によって署名されていることを確認するには、callbridge.crt の中間証明書のチェーンを通じて確立された信頼のチェーンを使用します。

4.4 Call Bridge の証明書と秘密キーのインストール

セクション 1.1.1 で説明されているように、展開内で各 Call Bridge がどのように使用されているかに応じて、秘密キー/証明書のペアが必要になる場合があります。

以下のステップでは、各 Call Bridge がリッスンするため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、MMP コマンド **listen**を使用してインターフェイスを設定する方法については、『Scalable and Resilient Server Deployment Guide』を参照してください。

各 Call Bridge について：

1. Meeting Server の MMP に SSH で接続します。
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>]

   ここで、 **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 によって証明書バンドルが提供された場合は、バンドルも個別のファイルとして証明書に含めます。

   例：

   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、次のように表示されます。

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

証明書のインストールに失敗すると、次のエラーメッセージが表示されます。

FAILURE：キーと証明書に問題があります：証明書とキーが一致しません

注：Web Bridge 3 を設定した後、すべての Web Bridge 3 の信頼ストアに Call Bridge 証明書を追加する必要があります。

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。

4.4.1 Call Bridge と Web Bridge 3 間の信頼の確立

Web Bridge 3 では、ゲストログインの設定とイメージのカスタマイズを Call Bridge からプッシュできます（カスタマイズガイドラインを参照）。展開のセキュリティにとって、信頼できる Call Bridge からのみ設定を受け入れることが重要です。

C2W 証明書は、Call Bridge と Web Bridge 3 の間の接続に使用されます。Call Bridge が Web Bridge 3 との C2W 接続を確立するには、証明書を検証する C2W 信頼ストアを指定する必要があります。詳細については、セクション 4.5 および 付録 D を参照してください。

4.5 Web Bridge 3 の証明書と秘密キーのインストール

Web Bridge 3 を展開して Cisco Meeting Server Web アプリを使用するための証明書設定の詳細については、付録 D を参照してください。 Web アプリケーションを使用できるように Web Bridge 3 を設定するのに役立つ情報を以下に示します。

- Web Bridge 3 では、すべての証明書定義で証明書のバンドル、つまり完全なチェーンファイルを使用する必要があります。これは、Web Bridge 2 の証明書の導入方法とは異なります。
- 「Call Bridge to Web Bridge」（C2W）プロトコルは、callbridge と webbridge3 の間のリンクです。
- Call Bridge が Web Bridge 3 に接続できるようにするには、（**`webbridge3 c2w listen`** を使用して）インターフェイス上でポートを開く必要があります（Web Bridge 3 がそのポートをリッスンします）。この理由から、この Web Bridge 3 について Call Bridge に情報を伝えるための API リクエストを実行するときに、このポートを指定したアドレスを使用する必要があります。この接続は、証明書を使用してセキュリティで保護する必要があります。
- 開いたポートを外部アクセスから保護することを推奨します。つまり、Call Bridge からのみポートに到達可能にする必要があります。
- Call Bridge は **`callbridge certs`** を使用して設定された証明書を使用し、Web Bridge は **`webbridge3 c2w certs`** を使用して設定された証明書を使用します。
- Web Bridge は、**`webbridge3 c2w trust`** によって設定された信頼ストアに含まれる、いずれかによって署名された、Call Bridge の証明書を信頼します。

- Call Bridge は、**`callbridge trust c2w`** によって設定された信頼ストアに含まれるいずれかによって署名された証明書を持つ Web Bridge 3 を信頼します。
- 公的機関が署名した証明書は必要ありません。MMP で作成した自己署名証明書を使用できます。
- SAN/CN は、Call Bridge API で Web Bridge 3 に登録するために使用された c2w:// の URL で使用されている FQDN または IP アドレスに一致する必要があります。（これが一致しない場合、Call Bridge は TLS ネゴシエーションに失敗し、Web Bridge 3 によって提示された証明書を拒否するため、Web Bridge 3 に接続できません）。
- Call Bridge および Web Bridge 3 証明書の拡張要件の詳細については、第 2.1 章 を参照してください。

4.6 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 を使用する予定の場合は、TURN サーバ用の署名付き証明書をインストールする必要があります。その際、Web Bridge で使用したものと同じ CA を使用して証明書に署名します。証明書は、Meeting Server との接続を信頼するかどうかを判断するときにブラウザによって使用されます。

以下のステップでは、TURN サーバがリッスンするため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 コマンドを使用してインターフェイスを設定する方法については、『Scalable and Resilient Server Deployment Guide』を参照してください。

各 TURN サーバについて：

1. ホストサーバーの MMP に SSH で接続します。
2. 証明書を割り当てる前に、TURN サーバインターフェイスを無効にします。

   `turn disable`
3. CA からの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 サーバに割り当てます。

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

ここで、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 によって証明書バンドルが提供された場合は、バンドルも個別のファイルとして証明書に含めます。

例

```
turn certs turn.key turn.crt turnbundle.crt
```

6. TURN サーバを再度有効化します。

```
turn enable
```

4.7 MeetingApps の証明書のインストール

参加者が Web アプリ会議でファイルを共有するためには、MeetingApp の証明書を設定する必要があります。公的に署名され、ブラウザが信頼する CA 証明書を使用することをお勧めします。ただし、内部 CA 署名付き証明書を使用している場合、Web アプリは、会議への参加に使用する各ブラウザで、これらの証明書を検証するように求めます。

証明書を割り当てる前に、MeetingApp が通信に使用するインターフェイスとポートを設定する必要があります。MeetingApps インターフェイスとポートの設定の詳細については、『*Cisco Meeting Server 導入ガイド*』を参照してください。

MeetingApps 証明書を設定するには、次の手順を実行します。

1. ホストサーバーの MMP に SSH で接続します。
2. 証明書を割り当てる前に MeetingApps を無効にします。

   ```
   meetingapps disable
   ```
3. 次のコマンドを使用して、MeetingApps の証明書とキーのペアを割り当てます。

   ```
   meetingapps https certs <key-file> <crt-fullchain-file>
   ```
4. MeetingApp を有効にします。

   ```
   meetingapps enable
   ```

4.7.1 MeetingApps 証明書の検証

接続を信頼するには、MeetingApps に使用される内部 CA 署名付き証明書を、Web アプリ会議ミーティングに使用する各ブラウザで検証する必要があります。内部 CA 署名付き証明書が検証されておらず、Web アプリが MeetingApps 証明書の検証に使用できるリンクとともに参加者にプロンプトを表示した場合、参加者は会議でファイルを共有できません。

証明書を検証するには、ブラウザで新しいタブを開き、MeetingApps アドレスに続けて「:」とポート番号を入力します。プロンプトが表示されたら、証明書を受け入れ、MeetingApps へのアクセスを続行する必要があります。MeetingApps に使用されるアドレスとポートは、**meetingapps** コマンドを使用して取得できます。

注：Web アプリ会議に使用するすべてのブラウザで、このプロセスを繰り返す必要があります。

MeetingApps 証明書を検証した後、Web アプリ会議でファイルを共有できます。

4.7.2 信頼されたルート認証局への証明書の追加

または、内部 CA 署名付き証明書を、Web アプリ会議に使用するすべてのブラウザの信頼されたルート認証局ストアにインストールすることもできます。

内部 CA 署名付き証明書を信頼されたルート認証局ストアに追加するには：

1. MeetingApps が使用する中間証明書とルート証明書をダウンロードします。
2. すべての内部 CA 署名付き証明書を信頼されたルート認証局にインポートします。

証明書を信頼ストアにインポートしたら、Web アプリ会議に参加してファイルを共有できます。

4.8 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、サーバー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には

  tls <sip|ldap>
- 信頼できる認証局を定義するには

  tls <sip|ldap> trust <crt bundle>
- 証明書の検証を有効または無効にするか、または OCSP が検証に使用されるかどうかを指定します。

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

詳細については、『MMP コマンドリファレンスガイド』を参照してください。

4.9 Call Bridge クラスタの検証

注：このセクションは、スケーラブルで回復力のある展開にのみ適用されます。

Call Bridge 信頼ストアを使用してクラスタ内の Call Bridge を検証することにより、Call Bridge クラスタのセキュリティを向上させることができます。Call Bridge は、Web 管理が前面にある HTTPS を介して相互に接続するため、クラスタ化された Call Bridge の Web 管理証明書を保持する証明書バンドルを作成し、証明書バンドルをクラスタの各 Call Bridge の信頼ストアにアップロードする必要があります。次の MMP コマンドを使用します。

`callbridge trust cluster <bundle name>`

Call Bridge がクラスタ内の別の Call Bridge に接続するときには、信頼ストアにある証明書のバンドルをチェックして、接続先の Call Bridge の ID を検証します。これにより、Call Bridge が安全でない Meeting Server に接続するリスクがなくなります。証明書バンドルは、証明書チェーンまたは信頼できる証明書の許可リストのいずれかです。

注：バージョン 3.4 から、「**`callbridge trust cluster`**」が有効になっているときには証明書名の検証が導入されているため、クラスタリングで設定されたピアは、対応する Web Admin 証明書の正確な FQDN と一致する必要があり、この設定を行わなかった場合、Call Bridge クラスタ障害が発生します。

信頼ストアが使用されていない場合、クラスタ化された Call Bridge 間で証明書の検証は行われず、Call Bridge は引き続きリモート Call Bridge への接続を確立しますが、その ID は検証されません。

Call Bridge クラスタ証明書バンドルを Call Bridge 信頼ストアから削除するには、次の MMP コマンドを使用します。

`callbridge trust cluster none`

5 証明書に関する問題のトラブルシューティング

このセクションでは、いくつかの一般的な問題のトラブルシューティングについて説明します。証明書に関するその他のよくある質問については、 Meeting Server ナレッジベースを参照してください。

5.1 サービスが信頼できないという警告メッセージ

このメッセージは、次のような場合に表示されます。

- 信頼ストアにない内部 CA を使用した場合。
- パブリックまたは内部 CA 署名付き証明書が必要な場所で自己署名証明書を使用した場合。証明書を再発行して、信頼できる CA によって署名してもらいます。このコンポーネントへのパブリックアクセスを予定しているのでない限り、これは内部 CA でもかまいません。

5.2 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に署名するために使用された 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のと同じ 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ことを確認します。

Lync に追加された FQDN が、Call Bridge の証明書の CN としても存在することを確認します。

5.3 証明書が間もなく期限切れになるか、すでに期限切れであるというメッセージ

証明書の更新は、新しい証明書セットの展開と同じプロセスに従います。証明書の取得とインストールについては、以前のセクションを参照してください。

クラスタ環境の場合、証明書の有効期限が切れると、クラスタ内のデータベースノードは相互通信を停止します。コマンド **`database cluster remove`** を使用してクラスタを削除しない限り、Meeting Server データベースクラスタノードで証明書を更新することはできません。したがって、データベースのクラスタ化を解除し、証明書を更新して、クラスタを再び作成します。詳細については、「Cisco Meeting Server を使用して期限切れのデータベースクラスタ証明書を更新する方法」を参照してください。

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、CA 証明書を取得できないため（スケーラブルで回復力のある展開の場合のみ）「クラスタキー」として使用したり、Lync 認証として使用したりすることはできません（CA が信頼できる機関ではないため）。ただし、ブラウザには証明書エラーが表示されますが、自己署名証明書は Web Admin、および Call Bridge と Web Bridge 間の相互認証に使用できます。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お勧めします。

Meeting Server でローカル秘密キーと自己署名証明書を生成するには：

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>

   ここで、**<key/cert basename>** は、生成されるキーと証明書を識別します。以下に

   その例を示します。

   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を作成します。

   callbridge.crt

付録 A 証明書を生成するための OpenSSL コマンド

セクション 1.2 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、使用する OpenSSL コマンドについて詳しく説明します。これらの例では、OpenSSL が Windows で実行されてい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も使用できます。

注：OpenSSL は管理者モードで実行してください。

注：この章の例では、Web Bridge 3 を使用しています。

A.1 RSA 秘密キーと CSR ファイルの生成

コンピュータで 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。次に例を示します。

```
openssl req -out webbridge3.csr -new -newkey rsa:2048 -nodes -keyout
webbridge3.key
```

webbridge3.csr という名前の CSR ファイルと、webbridge3.key という名前の RSA 2048 ビット秘密キーを生成します。

注：キー名と証明書名には「.」または「_」を含めないでください。たとえば、**webbridge3** は有効ですが、**web.bridge 3**または **web_bridge_3**は許可されません。

既存の秘密キーについて証明書署名要求（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
```

例：

```
openssl req -out webbridge3.csr -key webbridge3.key -new
```

webbridge3.key という名前の既存の秘密キーに基づいて、webbridge3.csr という名前の CSR ファイルを生成します。

OpenSSL を使用して証明書に自己署名する場合、中間 CSR ファイルは不要なので、次のセクションに進んでください。

A.2 CSR ファイルへの署名

パブリック CA を使用して CSR に署名するには、 セクション 1.2 の説明に従ってください。

内部 CA を使用して CSR に署名するには、 セクション 1.3 の説明に従

ってください。証明書に自己署名するには、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
```

例：

```
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（最終的な）証明書を生成します。

A.3 データベースクラスタリング用の証明書の作成

このセクションでは、スケーラブルで回復力のある展開のみを対象として、OpenSSL コマンドを使用してデータベースクラスタリング用の証明書を作成する方法について詳しく説明します。

注：データベースクラスタ証明書は 2.7 以降必須となりました。Meeting Server データベースクラスタのセットアップを容易にするために、データベースクラスタの署名付き証明書を作成する pki コマンドがあります。詳細については、セクションを参照してください。

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はユーザはアクセスできないため、CA、キー、および証明書は OpenSSL を使用して内部で生成できます。

データベースクラスタでは、クラスタ内のデータベースに保持または接続する各 Meeting Server で設定されたのと同じ CA によって署名されたクライアント証明書およびサーバ証明書が必要です。証明書の使用を必須にすることで、クラスタ全体の機密性と認証の両方が確保されます。

注意：証明書を必要としない旧バージョンの Meeting Server ソフトウェアを使用して、証明書を使用せずにデータベースクラスタが設定されている場合、バージョン 2.7 にアップグレードすると、データベースは停止し、証明書が設定されてデータベースクラスタが再作成されるまでアクセスできなくなります。

注：管理者権限で OpenSSL を実行していることを確認してください。

注：キー名と証明書名には「.」を含めないでください。または「_」、たとえば **db01_ca** または **db01.ca** は許可されていません。

次の手順に従ってください。

1. CA を定義し、CA の秘密キーと公開キーのペアと証明書を作成します。
   a. ユーザが定義した CA の秘密キーと証明書署名要求 (.csr) のペアを生成します。次の OpenSSL シンタックスを使用します。

      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>

      例：

      openssl req -new -text -nodes -keyout db01ca.key -out db01ca.csr -subj /C=UK/ST=London/L=London/O=Example/OU=/CN=example

      -subj に続く属性で定義された CA の秘密キー **db01ca.key**と証明書署名要求ファイル **db01ca.csr**を作成します。

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

      openssl req -x509 -text -in db01ca.csr -key db01ca.key -out db01ca.crt -days 3650

      証明書 **db01ca.crt** を作成します。

2. ステップ 1 で生成された CA ログイン情報を使用して、データベースサーバと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署名要求（.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>

ここで、 **nodename** は、データベースをホストするサーバの実際の名前です。たとえば次のようなものです。

```
openssl req -new -nodes -keyout db01server.key -out db01server.csr -subj
/C=UK/ST=London/L=London/O=Example/OU=/CN=server1
```

キー **db01server.key**と証明書署名要求ファイル **db01server.csr** を作成します。

b. データベースの CA 署名付き証明書を生成します。例：

```
openssl x509 -req -CAcreateserial -in db01server.csr -CA db01ca.crt -
CAkey db01ca.key -out db01server.crt -days 3650
```

証明書 **db01server.crt** を作成します

c. データベースの秘密キーと証明書署名要求（.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
```

例：

```
openssl req -new -nodes -keyout db01client.key -out db01client.csr -subj
/C=UK/ST=London/L=London/O=Example/OU=/CN=postgres
```

キー **db01client.key** と証明書署名要求ファイル **db01client.csr** を作成します。

d. データベースクライアントの CA 署名付き証明書を生成します。たとえば次のようなものです。

```
openssl x509 -req -CAcreateserial -in db01client.csr -CA db01ca.crt -
CAkey db01ca.key -out db01client.crt -days 3650
```

証明書 **db01client.crt** を作成します

3. セクション 1.8 の手順に従って、データベース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 サーバ証明書（ステップ 2 で生成）
      - データベース クラスタ サーバ キー（ステップ 2 で生成）
      - データベース クラスタ クライアント証明書（ステップ 2 で生成）
      - データベース クラスタ クライアント キー（ステップ 2 で生成）
      - データベースクラスタ CA 証明書バンドル（ステップ 1 で生成）

   b. データベースと同じ場所に配置されていない各 Call Bridge に、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 クライアント証明書（ステップ 2 で生成）
      - データベース クラスタ クライアント キー（ステップ 2 で生成）
      - データベースクラスタ CA 証明書バンドル（ステップ 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 への証明書と秘密キーのペアのインストールの詳細については、第 4 章の説明に従ってください。

付録 B 証明書ファイルと秘密キーで許可されている拡張子

次の表に、証明書ファイルと秘密キーに許可されているファイル拡張子を示します。

表 5：証明書ファイルに許可される拡張子

拡張	ファイルの種類に関する情報
.pem	PEM はエンコーディング（ASCII base64）であり、ファイル拡張子としても使用されます。通常、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的に生成されます。一部の安全な Web サイトでは、アイデンティティを認証するために、ユーザに PEM ファイルのアップロード（おそらく電子メールで送信）を求める場合があります。
DER	Distinguished Encoding Rules（DER）は、エンコーディングであり、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。一般的にはパブリック暗号化における X.509 証明書を保存するために使用されます。
.cer	Web サイトの真正性を検証する、VeriSign や Thwate などのサードパーティの認証局によって提供されるセキュリティファイル。サーバでホストされている特定の 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 としてエンコードできます。
.crt	安全な Web サイト（「https://」で始まる）が真正性を確認するために使用する証明書。Verisign や Thawte などの企業によって配布されています。証明書は、バイナリ DER または ASCII（Base64）PEM としてエンコードできます。 ユーザが安全なサイトにアクセスすると、証明書ファイルは Web ブラウザによって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して表示できます。

表 6：秘密キーファイルに許可される拡張子

拡張	ファイルの種類に関する情報
.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 としてエンコードできます。
.pem	キーが PEM（ASCII base64）を使用してエンコードされたことを示します。
.der	キーがバイナリ DER を使用してエンコードされたことを示します。

付録 C MMP PKI コマンド

以下は、MMP pki コマンドのリストです。

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）をリストします。
pki inspect <filename>	ファイルを検査し、ファイルが秘密キー、証明書、CSR、または不明であるかどうかを示します。証明書の場合は、さまざまな詳細が表示されます。ファイルに証明書のバンドルが含まれている場合、バンドルの各要素に関する情報が表示されます。 PEM および DER の両方の形式のファイルが処理されます。
pki match <key> <certificate>	このコマンドは、指定されたキーとシステム上の証明書が一致するかどうかを確認します。秘密キーと証明書は、1 つの使用可能な ID の 2 つの半分であり、callbridge などのサービスに使用される場合は一致する必要があります。
pki verify <cert> <cert bundle/CA cert> [<CA cert>] **pki verify server.pem bundle.pem rootca.pem** **pki verify server.pem bundle.pem**	証明書は認証局（CA）によって署名される場合があり、CA は中間 CA 証明書の「証明書バンドル」と、場合によっては CA 証明書を独自のファイルで提供します。証明書が CA によって署名されていることと、証明書バンドルを使用してこれを表明できることを確認するには、次のコマンドを使用します。
pki unlock <key>	多くの場合、秘密キーはパスワードで保護されます。Meeting Server で使用するには、キーのロックを解除する必要があります。 このコマンドは、ターゲットファイルのロックを解除するためのパスワードの入力を求めます。ロックされた名前は、同じ名前のロック解除されたキーに置き換えられます

コマンド/例	説明/注意事項
pki csr <key/cert basename> [<attribute>:<value>] **pki csr example CN:www.example.com OU:"My Desk" O:"My Office" L:"San Jose" ST:California C:US**	Cisco が秘密キーマテリアルの生成要件を満たしていることをユーザーが信頼できるように、秘密キーおよび関連する証明書署名要求を生成できます。 <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」と入力すると、「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、属性名と値のペアをコロン（「:」）で区切って指定できます。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: 組織単位 O: 組織 L: 地方 ST: 州 C: 国 emailAddress: 電子メールアドレス CSR ファイルは SFTP でダウンロードして、署名のために認証局（CA）に渡すことができます。（または、CSR ファイルを「pki sign」コマンドで使用して、証明書をローカルで生成することもできます。) 返送時には、SFTP 経由でアップロードする必要があります。その後、証明書として使用できます。 注：1.6.11 から、**pki csr <key/cert basename> [<attribute>:<value>]** は、属性として subjectAltName を取るようになりました。subjectAltName の IP アドレスとドメイン名は、コンマ区切りのリストでサポートされています。たとえば次のようなものです。 **pki csr test1 CN:example.exampledemo.com subjectAltName:exampledemo.com** **pki csr test2 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,exampledemo.com, server.exampledemo.com,join.exampledemo.com, test.exampledemo.com** 証明書のサイズと チェーン内の証明書の数を最小限に抑えてください。そうしないと、TLS ハンドシェイクのラウンドトリップ時間が長くなります。

コマンド/例	説明/注意事項
pki selfsigned <key/cert basename> [<attribute>:<value>]	このコマンドを使用して、自己署名証明書を生成できます。 <key/cert basename> は、生成されるキーと証明書を識別します。たとえば、「pki selfsigned new」と入力すると、new.key と new.crt（自己署名証明書）が生成されます。 CSR の属性は、属性名と値のペアをコロン（「:」）で区切って指定できます。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: 組織単位 O: 組織 L: 地方 ST: 州 C: 国 emailAddress: 電子メールアドレス CSR ファイルは SFTP でダウンロードして、署名のために認証局（CA）に渡すことができます。返送時には、SFTP 経由でアップロードする必要があります。その後、証明書として使用できます。 証明書のサイズとチェーン内の証明書の数を最小限に抑えてください。そうしないと、TLS ハンドシェイクのラウンドトリップ時間が長くなります。
pki sign <csr/cert basename> <CA key/cert basename>	このコマンドは <csr/cert basename> によって識別される CSR に署名し、<CA key/cert basename> によって識別される CA 証明書とキーで署名された、同じベース名を持つ証明書を生成します。 ファイル <csr/cert basename> および <CA key/cert basename> は、コマンド「pki csr」および「pki selfsigned」によってそれぞれ生成されている必要があります。
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、最初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求められます。完了後、pkcs#12 ファイルはパスワード保護なしで使用可能なキーに置き換えられます。 注：pkcs#12 ファイルに含まれるその他のデータはすべて失われます。 このコマンドを実行することで、ユーザー john のアップロードされた PKCS#12 ファイル john.pub のキーを使用可能にすることができます。

付録 D Cisco Meeting Server Web アプリケーションを使用するために Web Bridge 3 を展開するための証明書と設定情報

D.1 Web Bridge 3 を使用するための Meeting Server の構成

Web Bridge 3 では、HTTPS ポートと C2W ポートを設定する必要

があります。Web Bridge3 を使用するように Meeting Server を設

定するには、次の手順を実行します。

1. MMP に SSH でログインします。
2. MMP で **webbridge3**コマンドを使用して、Web Bridge 3 を設定します。
   Web Bridge 3 の使用方法を表示するには、「**help webbridge3**」と入力します。

```
> help

     webbridge3

     Usage:
     webbridge3
     webbridge3 restart
     webbridge3 enable
     webbridge3 disable
     webbridge3 https listen <interface:port allowed list>
     webbridge3 https certs <key-file> <crt-fullchain-file>
     webbridge3 https certs none
     webbridge3 http-redirect (enable [port]|disable)
     webbridge3 c2w listen <interface:port allowed list>
     webbridge3 c2w certs <key-file> <crt-fullchain-
     file> webbridge3 c2w certs none
     webbridge3 c2w trust <crt-bundle>
     webbridge3 c2w trust none
     webbridge3 options <space-separated options>
     webbridge3 options none
     webbridge3 status
```

3. （省略可）HTTP 接続用のポートをセットアップします。このポートは、Web アプリケーションが設定されているすべての Meeting Server インターフェイスに対して開かれます。着信 HTTP 接続は、着信したインターフェイスでの一致する HTTPS ポートに自動的にリダイレクトされます。**webbridge3 http-redirect enable [port]** でポートを指定しない場合、デフォルトのポートは 80 です。

4. HTTPS サービスがリッスンするポートを設定します。インターフェイスのポート 443 でリッスンするように設定するには、次のコマンドを実行します。

   webbridge3 https listen a:443

5. HTTPS 証明書を設定します。これらは Web ブラウザに対して提示される証明書であるため、認証局による署名が必要であり、ホスト名や目的などが一致している必要があります。（証明書ファイルは、エンドエンティティの証明書で始まり、ルート証明書で終わる完全な証明書チェーンです）。次のコマンドを入力します。

   webbridge3 https certs wb3-https.key wb3-https-fullchain.crt

6. C2W 接続を設定します。このアドレスとポートは、Call Bridge からのみアクセス可能にすることを推奨します。次のコマンドを実行すると、インターフェイス a のポート 9999 に設定されます。

   webbridge3 c2w listen a:9999

   ここでは例としてポート 9999 を使用していますが、ネットワーク上の利用可能な任意のポートを使用できます。これは、443 とは異なり、固定ポートではありません。

7. C2W 接続の証明書を設定します。C2W 接続に使用する SSL サーバ証明書を設定する必要があります。（証明書の要件については、下記の「C2W 接続を使用するように Call Bridge を設定する」を参照してください。さらに詳細な情報については、この FAQ [英語] を参照してください）。

   webbridge3 c2w certs wb3-c2w.key wb3-c2w-fullchain.crt

8. Web Bridge 3 C2W サーバは、Call Bridge がクライアント証明書を提示することを想定しており、次のコマンドによって提供される信頼バンドルを使用して、Call Bridge を信頼するかどうかを検証します。

   webbridge3 c2w trust wb3-c2w-trust-bundle.crt

9. Web Bridge 3 を有効にします。

   webbridge3 enable

D.2 C2W 接続を使用するための Call Bridge の構成

C2W 証明書は、Call Bridge と Web Bridge 3 の間の接続に使用されます。Call Bridge が Web Bridge 3 との C2W 接続を確立するには、証明書を検証する C2W 信頼ストアを指定する必要があります。つまり、前述のステップ 7 で設定した、Web Bridge 3 が提示する証明書です。

1. Call Bridge の使用方法を表示するには、MMP で次の **callbridge**コマンドを使用します。「**help callbridge**」と入力すると、次のように表示されます。

   ```
   > help callbridge
   Configure CMS callbridge


   使用方法


       callbridge listen <interface allowed list>
       callbridge prefer <interface>
       callbridge certs <key-file> <crt-file> [<cert-bundle>]
       callbridge certs none
       callbridge trust c2w <bundle>
       callbridge trust c2w none
       callbridge add edge <ip address>:<port>
       callbridge del edge
       callbridge trust edge <trusted edge certificate bundle>
       callbridge trust cluster none
       callbridge trust cluster <trusted cluster certificate bundle>
       callbridge restart
   ```

2. Call Bridge の証明書を設定します。

   ```
   callbridge certs cert.key cert.crt
   ```

3. Web Bridge 3 が提示した SSL サーバ証明書の検証に使用する C2W 信頼ストアを設定します。（詳細については、この FAQ [英語] を参照してください）。

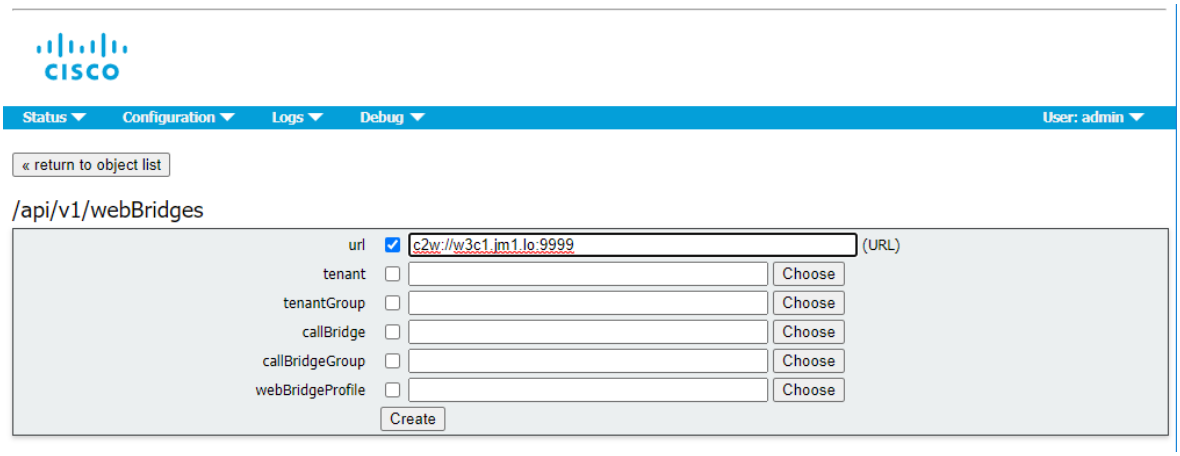
   ```
   callbridge trust c2w c2w-callbrige-trust-store.crt
   ```

4. Call Bridge を再起動します。

   ```
   callbridge restart
   ```

5. Web 管理ユーザインターフェイスに移動し、[**設定（Configuration）**] > [**API**] を選択し、**/api/v1/webBridges** を選択して、実行中の callbridge REST API に Web Bridge 3 URL を以下のように登録します。URL プロトコルは webbridge3 であることを示します。つまり URL で c2w:// プロトコルを指定することで、webbridge3 接続として処理されます。

図 4：Call Bridge API に対する Web Bridge 3 の URL の登録



Cisco 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、すべてユーザー側の責任となります。

対象製品のソフトウェア ライセンスと限定保証は、製品に添付された『Information Packet』に記載されています。ソフトウェアライセンスまたは限定保証書が見つからない場合は、CISCO の代理店に連絡してコピーを入手してください。

Cisco が採用している TCP ヘッダー圧縮機能は、UNIX オペレーティング システムの UCB（University of California, Berkeley）のパブリック ドメイン バージョンとして、UCB が開発したプログラムを採用したものです。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。シスコおよび上記代理店は、商品性、特定目的適合、および非侵害の保証、もしくは取り引き、使用、または商慣行から発生する保証を含み、これらに限定することなく、明示または暗黙のすべての保証を放棄します。

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などの図は、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 IP 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。

この文書の印刷されたハードコピーおよび複製されたソフトコピーは、すべて管理対象外と見なされます。最新バージョンについては、現在のオンラインバージョン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。各オフィスの住所と電話番号は、当社の Web サイト（http://www.cisco.com/jp/go/offices）をご覧ください。

Cisco の商標