サンプル トランザクションおよびパケット交換による SSL の概要

目次

概要

このドキュメントでは、セキュア ソケット レイヤ (SSL) プロトコルの基本概念について説明し、サンプル トランザクションとパケット キャプチャを提供します。

SSL レコードの概要

SSL 内のデータの基本単位がレコードです。 各レコードは 5 バイトのレコード ヘッダーとその後に続くデータで構成されます。

レコード形式

- **タイプ** : uint8 - 値は後述
- **バージョン** : uint16
- **長さ** : uint16

タイプ Version 長さ
T VH VL LH LL

レコード タイプ

SSL には 4 つのレコード タイプがあります。

- **ハンドシェイク**（22、0x16）
- **暗号仕様変更**（20、0x14）
- **アラート**（21、0x15）
- **アプリケーション データ**（23、0x17）

レコード バージョン

レコード バージョンは 16 バイト値で、ネットワーク順序でフォーマットされます。

> **注:** SSL バージョン 3（SSLv3）の場合は、バージョンが 0x0300 になります。 Transport Layer Security バージョン 1（TLSv1）の場合は、バージョンが 0x0301 になります。 Cisco 適応型セキュリティ アプライアンス（ASA）は、バージョン 0x0002 を使用する SSL バージョン 2（SSLv2）、または、TLSv1 以降の任意の TLS バージョンをサポートしません。

レコード長

レコード長は 16 バイト値で、ネットワーク順序でフォーマットされます。

理論上、これは 1 つのレコードを最大 65,535（2^16 -1）バイトの長さにできることを意味します。 TLSv1 RFC2246 では、最大長が 16,383（2^14 -1）と規定されています。 Microsoft 製品（Microsoft Internet Explorer とインターネット インフォメーション サービス）はこの制限を超えていることが知られています。

レコードのタイプ

ここでは、4 種類の SSL レコードについて説明します。

ハンドシェイク レコード

ハンドシェイク レコードには、ハンドシェイクに使用される一連のメッセージが含まれています。 次に、メッセージとその値を示します。

- Hello Request（0、0x00）
- Client Hello（1、0x01）
- Server Hello（2、0x02）
- Certificate（11、0x0B）
- Server Key Exchange（12、0x0C）
- Certificate Request（13、0x0D）
- Server Hello Done（14、0x0E）
- Certificate Verify（15、0x0F）
- Client Key Exchange（16、0x10）

- **Finished**（20、0x14）

シンプルなケースでは、ハンドシェイク レコードが暗号化されません。 ただし、Finished メッセージを含むハンドシェイク レコードは、常に暗号仕様変更（CCS）レコードの後ろに出現するため、必ず暗号化されます。

暗号仕様変更レコード

CCS レコードは暗号化方式の変更を示すために使用されます。 CCS レコード以降のすべてのデータが新しい暗号を使って暗号化されます。 CCS レコードは暗号化される場合とされない場合があります。 1 回のハンドシェイクによるシンプルな接続では、CCS レコードが暗号化されません。

アラート レコード

アラート レコードは、ある状態が発生したことをピアに知らせるために使用されます。 一部のアラートは警告ですが、その他は致命的で、接続が失敗します。 アラートは、暗号化される場合とされない場合があり、ハンドシェイク中やデータ転送中に発生する可能性があります。 アラートには次の 2 種類があります。

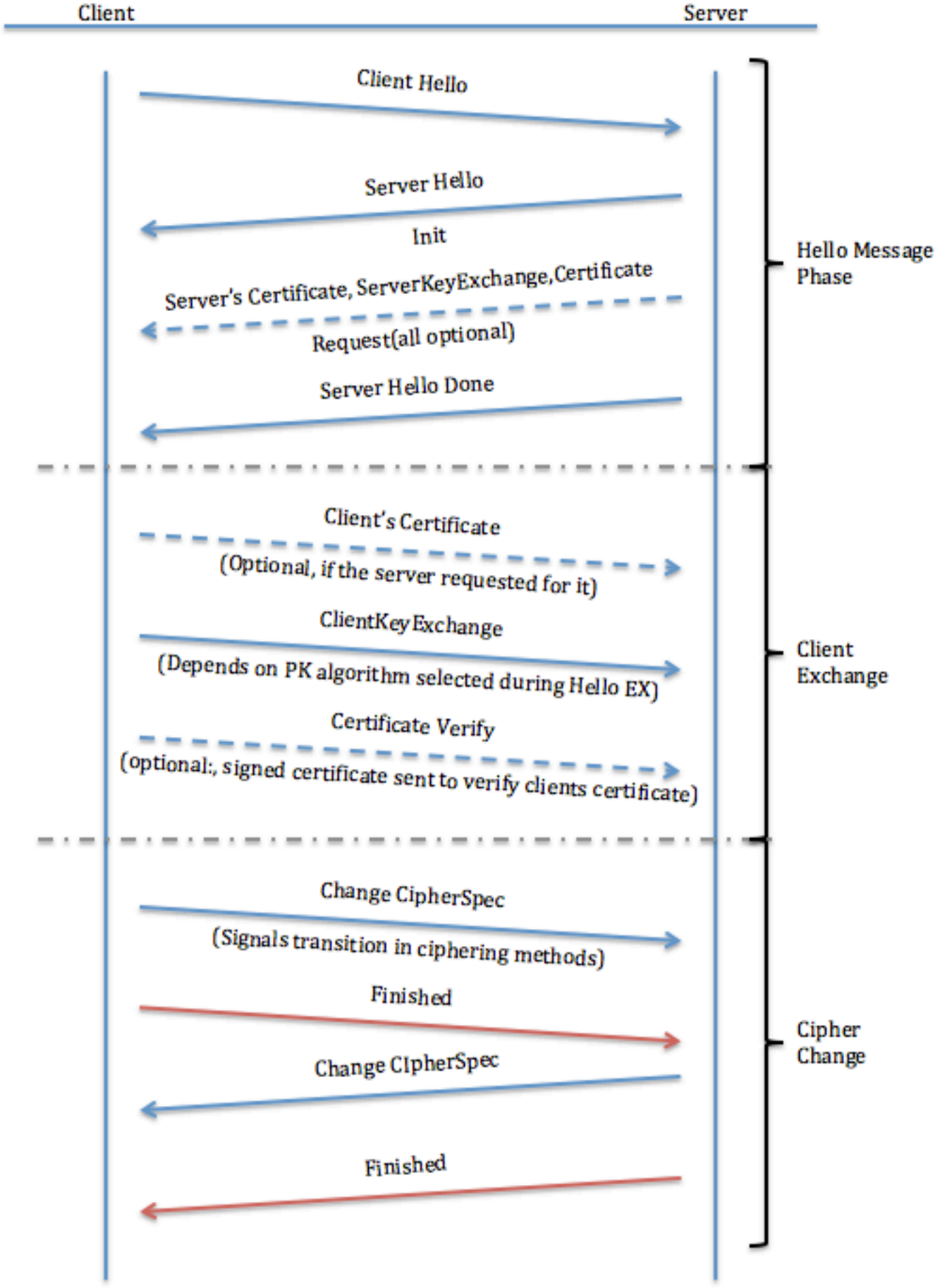
- **クロージャ アラート:** トランケーション攻撃を防ぐためには、クライアントとサーバ間の接続が正しく閉じられている必要があります。 送信者がその接続ではこれ以上メッセージを送信しないことを示す close_notify メッセージが受信者に送信されます。
- **エラー アラート**： エラーが検出されると、検出したパーティが他方にメッセージを送信します。 致命的なアラート メッセージの送受信時は、両方のパーティが即座に接続を閉じます。 エラー アラートの例は次のとおりです。
  - unexpected_message（致命的）
  - decompression_failure
  - handshake_failure

アプリケーション データ レコード

このレコードには実際のアプリケーション データが含まれています。 これらのメッセージは、レコード層で伝送され、現在の接続状態に基づいて、フラグメント化、圧縮、または暗号化されます。

サンプル トランザクション

ここでは、クライアントとサーバ間のサンプル トランザクションについて説明します。


Client
Server
Client Hello
Server Hello
Init
Server's Certificate, ServerKeyExchange,Certificate
Request(all optional)
Server Hello Done
Hello Message Phase
Client's Certificate
(Optional, if the server requested for it)
ClientKeyExchange
(Depends on PK algorithm selected during Hello EX)
Certificate Verify
(optional:, signed certificate sent to verify clients certificate)
Client Exchange
Change CipherSpec
(Signals transition in ciphering methods)
Finished
Change CIpherSpec
Finished
Cipher Change

Hello 交換

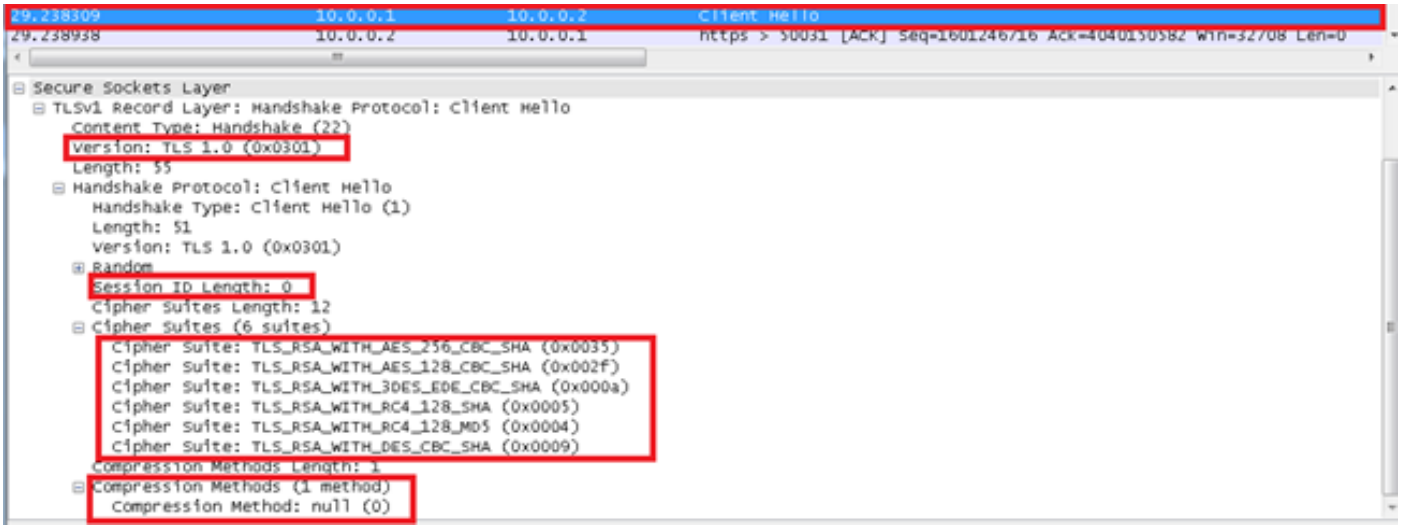
SSL クライアントとサーバが通信を開始すると、プロトコル バージョンに同意して、暗号化アルゴリズムを選択し、オプションで相互に認証し合って、公開キー暗号化テクニックを使って共有秘密を生成します。 これらのプロセスはハンドシェイク プロトコルで実行されます。 要約すると、クライアントが Client Hello メッセージをサーバに送信します。このメッセージには Server Hello メッセージで応答する必要があり、そうしない場合は致命的なエラーが発生して接続が失敗します。 Client Hello と Server Hello は、クライアントとサーバ間でセキュリティ拡張機能を設定するために使用されます。

Client Hello

Client Hello は、次の属性をサーバに送信します。

- **プロトコル バージョン：** クライアントがこのセッション中に通信する SSL プロトコルのバージョン。
- **セッション ID：** クライアントがこの接続に使用するセッションの ID。 交換の最初の Client Hello では、セッション ID が空です (後述する注意の下のパケット キャプチャ画面を参照)。
- **暗号スイート：** これは、Client Hello メッセージでクライアントからサーバに渡されます。これには、クライアントのプリファレンス (最優先選択肢) の順番でクライアントでサポートされる暗号化アルゴリズムの組み合わせが含まれています。 暗号スイートごとに、キー交換アルゴリズムと暗号仕様の両方が定義されます。 サーバは、暗号スイートを選択するか、受け入れ可能な選択肢がない場合は、ハンドシェイク障害アラートを返して、接続を閉じます。
- **圧縮方式：** クライアントでサポートされる圧縮アルゴリズムのリストが含まれています。 サーバがクライアントから送信された方式をサポートしていない場合は、接続が失敗します。圧縮方式は null にすることもできます。

  注: キャプチャ内のサーバ IP アドレスは 10.0.0.2 で、クライアント IP アドレスは 10.0.0.1 です。

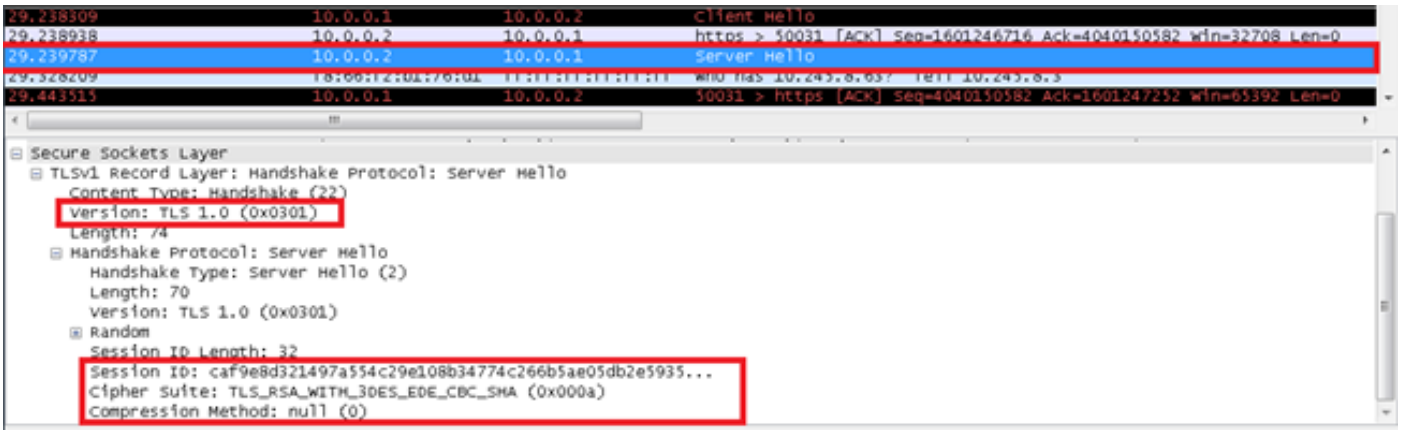


Server Hello

サーバは次の属性をクライアントに送り返します。

- **プロトコル バージョン：** クライアントがサポートする SSL プロトコルの選択されたバージョン。

- **セッション ID：** これは、この接続に対応するセッションの ID です。 Client Hello でクライアントから送信されたセッション ID が空でない場合は、サーバがセッション キャッシュで一致するものを探します。 一致するものが見つかって、サーバが指定されたセッション状態を使用して新しい接続を確立する場合は、クライアントから提供されたものと同じ値で応答します。 これは、セッションが再開されたことを示し、パーティが直接 Finished メッセージに進む必要があることを示します。 そうでない場合は、このフィールドに新しいセッションを識別する別の値が設定されます。 サーバは、空の **session_id** を返すことによって、セッションがキャッシュされないために再開できないことを示す場合があります。
- **暗号スイート：** クライアントから送信されたリストからサーバが選択したもの。
- **圧縮方式：** クライアントから送信されたリストからサーバが選択したもの。
- **証明書要求：** サーバは、クライアントに、設定されているすべての証明書のリストを送信して、認証に使用する証明書の選択を許可します。

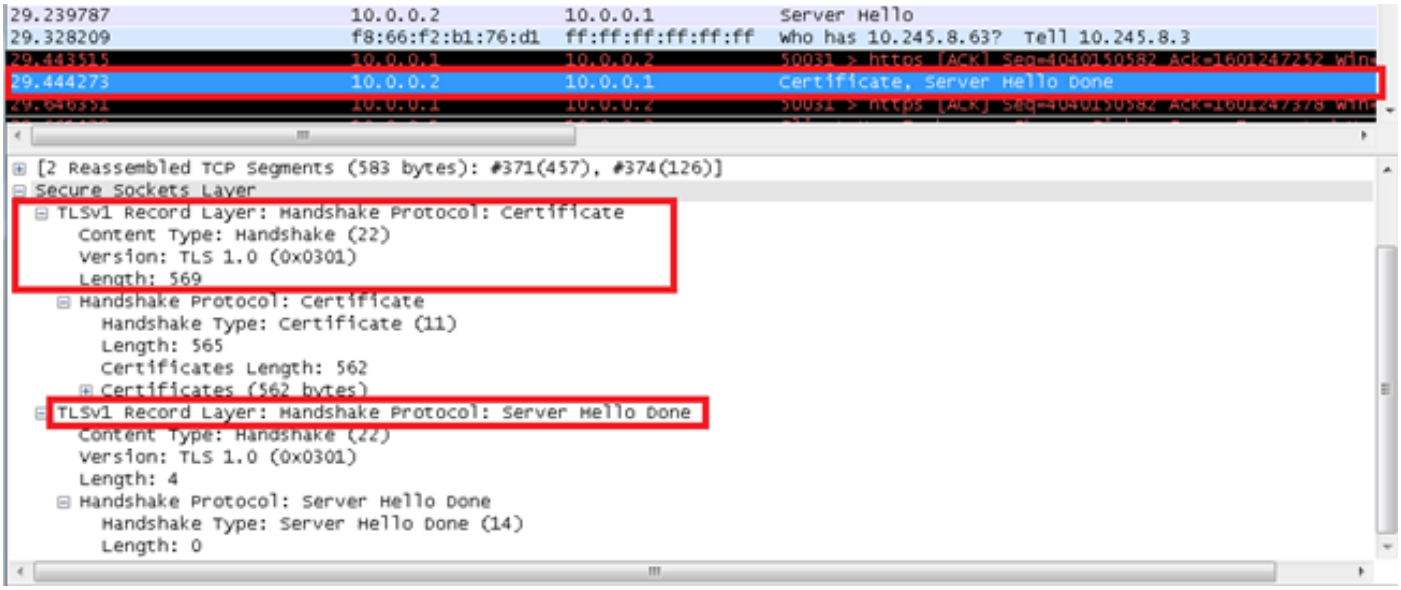


SSL セッション再開要求の場合：

- サーバはクライアントに Hello 要求も送信できます。 これは、都合の良いときに、Client Hello 要求を使って再ネゴシエーションを開始する必要があることをクライアントに知らせるためだけのものです。 クライアントは、ハンドシェイク プロセスがすでに進行中であれば、サーバからの Hello 要求を無視します。
- ハンドシェイク メッセージは、アプリケーション データの伝送より優先されます。 再ネゴシエーションは、最大長アプリケーション データ メッセージの伝送時間の 1 ~ 2 倍以内で開始する必要があります。

Server Hello Done

Server Hello Done メッセージは、Server Hello と関連するメッセージの終わりを示すためにサーバによって送信されます。 このメッセージの送信後に、サーバはクライアントの応答を待ちます。 Server Hello Done メッセージを受信すると、クライアントは、必要に応じて、サーバから有効な証明書が提供されたことを確認し、Server Hello パラメータが受け入れ可能かどうかをチェックします。



Server Certificate、Server Key Exchange、および Certificate Request（オプション）

- **サーバ証明書：** サーバが認証を受ける必要がある場合（よくあるケース）は、Server Hello メッセージの直後にその証明書を送信します。 証明書の種類は、選択された暗号スイート キー交換アルゴリズムに適している必要があり、通常は X.509.v3 証明書が使用されます。
- **Server Key Exchange：** Server Key Exchange メッセージは、サーバ上に証明書が存在しない場合にサーバから送信されます。 Diffie?Hellman（DH）パラメータがサーバ証明書に含まれている場合は、このメッセージが使用されません。
- **証明書要求：** サーバは、オプションで、選択された暗号スイートに適していれば、クライアントに証明書を要求できます。

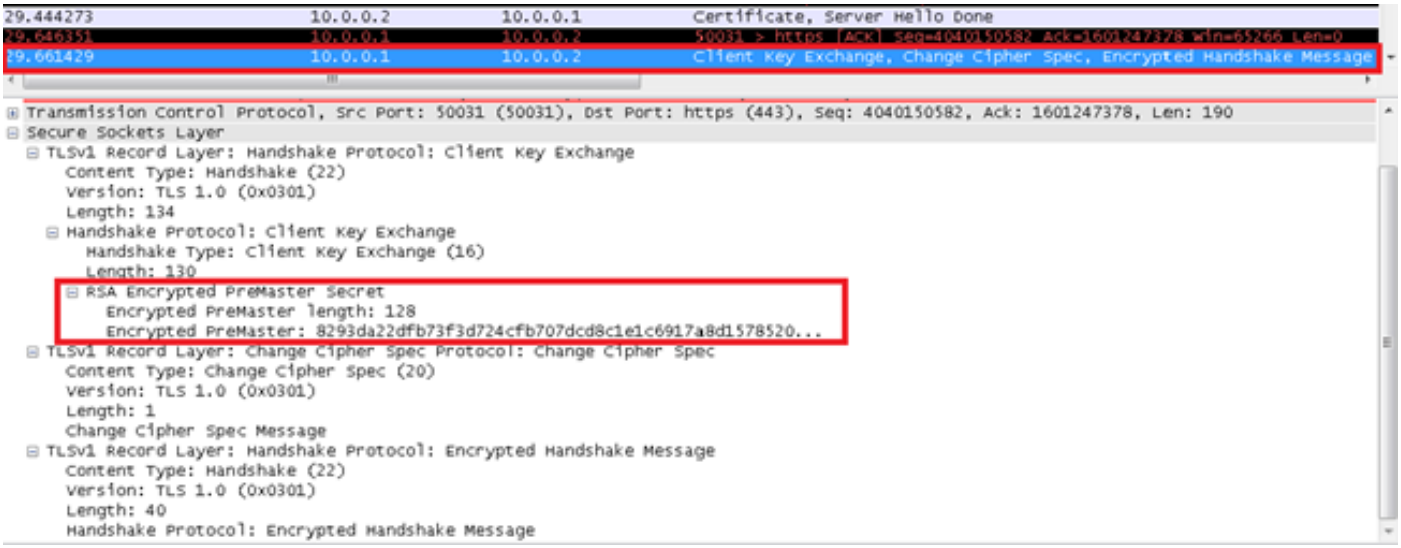
クライアント交換

Client Certificate（オプション）

これは、Server Hello Done メッセージの受信後にクライアントが送信する最初のメッセージです。 このメッセージは、サーバが証明書を要求した場合にだけ送信されます。 適切な証明書が使用できない場合は、クライアントが代わりに **no_certificate** アラートを送信します。 このアラートは単なる警告です。 ただし、クライアント認証が必要な場合は、サーバが致命的なハンドシェイク障害アラートで応答する可能性があります。 クライアント DH 証明書は、サーバ指定の DH パラメータと一致する必要があります。

Client Key Exchange

このメッセージの内容は、Client Hello メッセージと Server Hello メッセージ間で選択された公開キー アルゴリズムによって異なります。 クライアントは、Rivest-Shamir-Addleman（RSA）アルゴリズムで暗号化されたプレマスター キーと、キーの同意と認証用の DH のどちらかを使用します。 RSA がサーバ認証とキー交換に使用される場合は、48 バイトの **pre_master_secret** がクライアントによって生成され、サーバ公開キーに基づいて暗号化され、サーバに送信されます。 サーバは、秘密キーを使用して、**pre_master_secret** を復号化します。 その後で、両方のパーティが **pre_master_secret** を **master_secret** に変換します。



Certificate Verify（オプション）

クライアントが署名可能な証明書を送信すると、デジタル署名された Certificate Verify メッセージが送信され、証明書が明示的に確認されます。

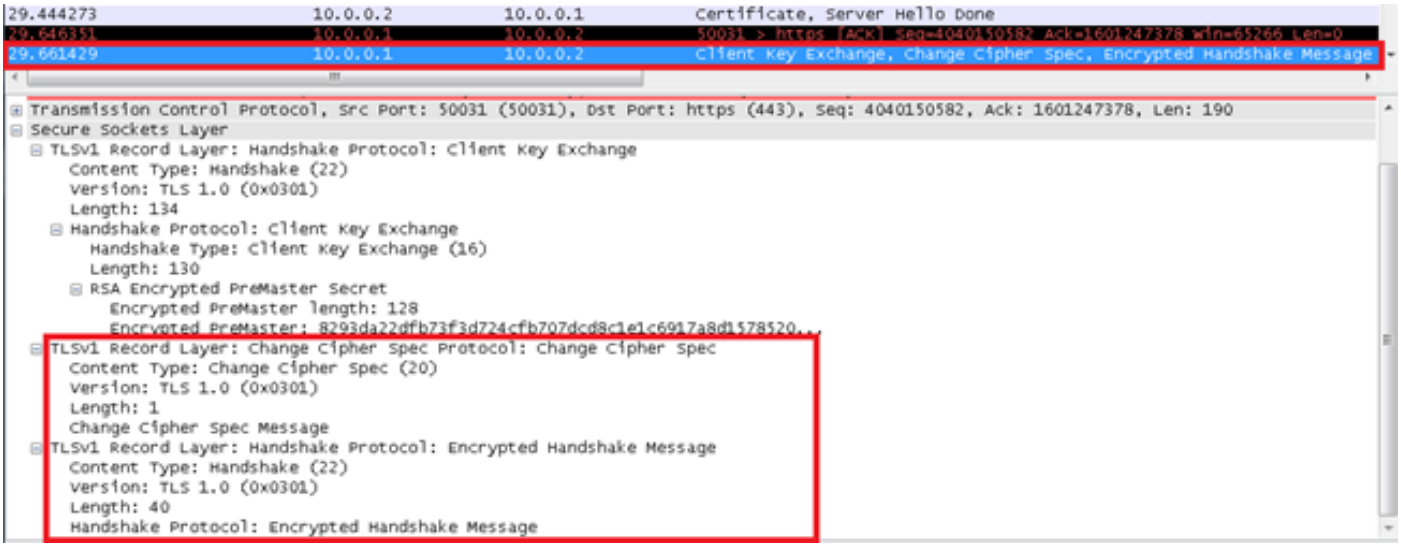
暗号の変更

暗号仕様変更メッセージ

暗号仕様変更メッセージがクライアントによって送信され、クライアントは保留中の暗号仕様（新しい仕様）を現在の暗号仕様（これまで使用されていた仕様）にコピーします。 暗号仕様変更プロトコルは、暗号化方針の変更を知らせるために存在します。 このプロトコルは、現在の（保留になっていない）暗号仕様に基づいて暗号化および圧縮された 1 つのメッセージで構成されます。 このメッセージは、後続のレコードが最新のネゴシエートされた暗号仕様とキーに基づいて保護されていることを受信側パーティに通知するためにクライアントとサーバの両方から送信されます。 このメッセージを受信すると、受信側は "read pending" 状態を "read current" 状態にコピーします。 クライアントはハンドシェイク キー交換の後の暗号仕様変更メッセージと Certificate Verify メッセージ（もしあれば）を送信し、サーバはクライアントから受信したキー交換メッセージを正常に処理してからメッセージを送信します。 1 つ前のセッションが再開されると、暗号仕様変更メッセージが Hello メッセージの後に送信されます。 キャプチャでは、Client Exchange、Change Cipher、および Finished の各メッセージがクライアントから 1 つのメッセージとして送信されています。

Finished メッセージ

Finished メッセージは、キー交換プロセスと認証プロセスが成功したことを確認するために、常に暗号仕様変更メッセージの直後に送信されます。 Finished メッセージは、最新のネゴシエートされたアルゴリズム、キー、および秘密で保護される最初のパケットです。 Finished メッセージの確認応答は必要ありません。 パーティは、Finished メッセージの送信直後に暗号化されたデータの送信を開始できます。 Finished メッセージの受信側は、その内容が正しいことを確認する必要があります。



関連情報

- [RFC 6101 - セキュア ソケット レイヤ プロトコル バージョン 3.0]
- [テクニカルサポートとドキュメント - Cisco Systems]