EAP-TLS の概要

国立研究開発法人 情報通信研究機構

1. 基本情報

✧ 名前

PPP EAP TLS Authentication Protocol

✧ 機能

EAP において TLS を用いて用する暗号アルゴリズムのネゴシエーションと相互認証・鍵交換を実現することを目的としたプロトコル。

✧ 関連する標準

RFC2716 (https://www.ietf.org/rfc/rfc2716.txt)

2. プロトコル仕様

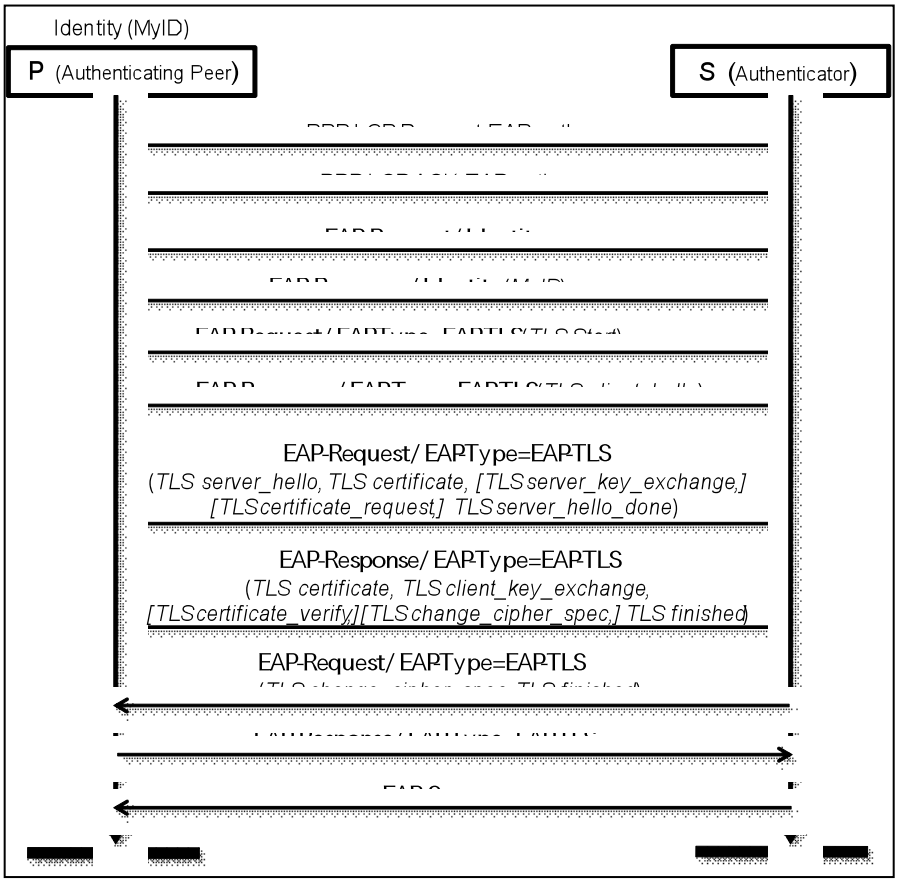

図 1. シーケンス図

2.1. EAP 全般について

EAP パケットは以下の 4 種類のフィールドで構成される。

- ✧ Code：パケットの種類を特定するための定数。Request, Response, Success, Failure の 4 種類がある。
- ✧ Identifier：Request と Response を対応させるためのフィールド。
- ✧ Length：パケット全体の長さ。
- ✧ Data：任意のペイロードであり、RFC3748 の中では規定されない。

Code フィールドが Request もしくは Response の場合、さらに以下のとおり。

- ✧ Type：Request/Response で運ぶペイロードの種類を指定するフィールド。RFC3748 では 8 個の値が指定されており、デフォルトの認証方式は MD5-Challenge もしくは One Time Password である。EAP 上で拡張方式による認証を行う場合には、Type フィールドで認証方式を指定する。
- ✧ Type-Data：Type で指定された型のペイロード。

2.2. EAP-TLS について

EAP-TLS では、EAP の Type フィールドが EAP-TLS に設定される。EAP-TLS ではいくつかの暗号プロトコルを定義しており、バリエーションも多い。本評価では、2. の参考文献の 2.1.1 に記載されている Base Case を評価対象とする。以下、EAP-TLS の Data フィールドのペイロードについて説明する。上記のシーケンス図におけるメッセージのペイロードは以下のとおりである。

- ✧ メッセージ 1：EAP-Request/Identity
  Data フィールドは何も含まない。
- ✧ メッセージ 2：EAP-Response/Identity
  Data フィールドはピアの ID を含む。
- ✧ メッセージ 3：EAP-Request/EAP-TLS (TLS Start)
  Data フィールドは「Start ビット」を含む。
- ✧ メッセージ 4：EAP-Response/EAP-TLS (TLS client_hello)
  - ➢ TLS client_hello：以下のペイロードを含む。
    - ・ クライアントのバージョン
    - ・ 乱数 client_random
    - ・ セッション ID session-id
    - ・ 利用可能な暗号スイートのリスト
    - ・ 利用可能なハッシュ関数のリスト

✧ メッセージ 5：EAP-Request/EAP-TLS (TLS server_hello, TLS certificate, [TLS server_key_exchange], TLS certificate_request, TLS server_hello_done)

- TLS server_hello：以下のペイロードを含む。
  - サーバのバージョン
  - 乱数 server.random
  - セッション ID session-id（生成方法は後述）
  - 選択された暗号スイート cipher_suite
  - 選択されたハッシュ関数 compression_method
- TLS certificate：以下のペイロードを含む。
  - 公開鍵証明書
  - 方式としては、RSA、DHE-DSS、DHE-RSA、DH-DSS、DH-RSA
- TSL server_key_exchange：オプションであり、クライアントが premaster secret を変更するために十分な情報を certificate ペイロードが含まない場合にのみ用いられる。
  - 利用可能な証明書のリスト
  - 利用可能な認証局のリスト

✧ メッセージ 6：EAP-Response/EAP-TLS (TLS certificate, TLS client_key_exchange, TLS certificate_verify, TLS change_cipher_spec, TLS finished)

- TLS certificate：ピアの公開鍵証明書。
- TLS client_key_exchange：premaster secret を共有するためのデータ。RSA を用いる場合にはサーバの公開鍵で暗号化された premaster secret。DH を用いる場合には公開する値 g^x である。
- TLS certificate_verify：メッセージ 3～5 の TLS メッセージに対する署名値。
- TLS change_cipher_spec：OPTIONAL。暗号スイートのスペックを変更するための値。
- TLS finised メッセージは鍵共有が成功したことを確認するための verify_data を含む。
  - Verify_data=PRF(master_secret,
    finished_label, MD5(handshake_message)
    +SHA1-128(handshake_message))
  - finished_label：固定値「client finished.」である。
  - Handshake_message：メッセージ 3～6 の TLS メッセージすべてを結合したデータ。

✧ メッセージ 7：EAP-Request/EAP-TLS (TLS change_cipher_spec, TLS finished)

- TLS finised メッセージは鍵共有が成功したことを確認するための verify_data を含む。
  - Verify_data=PRF(master_secret, finished_label, MD5(handshake_message) +SHA1-128(handshake_message))
  - finished_label：固定値「client finished.」である。
  - Handshake_message:メッセージ 3～7 の TLS メッセージすべてを結合したデータ。

✧ メッセージ 8：EAP-Response/EAP-TLS

Data フィールドは何も含まない。

✧ メッセージ 9：EAP-Success

Data フィールドは何も含まない。

2.3. 鍵階層について

✧ Key_Material
= TLS-PRF-128(master_secret, “client EAP encryption”, client.random||server.random)

✧ MSK = Key_Material[0..63]

✧ EMSK = Key_Material(64,127)

✧ IV = TLS-PRF-64(“”, “cclient_EAP_encryption”, client.random||server.random)

✧ Enc-RECV-Key = MSK(0,31)：ピアからサーバに送るデータの暗号化鍵

✧ Enc-SEND-Key = MSK(32,63)：サーバからピアに送るデータの暗号化鍵

✧ RECV-IV = IV(0,31)：ピアからサーバにデータを送るときの IV

✧ SEND-IV = IV(32,63)：サーバからピアにデータを送るときの IV

✧ Session-Id = 0x0D || client.random || server.random

3. 攻撃者モデル（自然言語による記述）

攻撃者として Dolev-Yao モデルを想定する。すなわち、通信の盗聴、改ざん、遮断、再送が可能とする。

4. セキュリティ要件（自然言語による記述）

RFC3748 では、暗号プロトコルの安全性について記述するためにいくつかの例を示しており、多くの場合これに従ってセキュリティプロパティが記述される。EAP-TLS については、

以下のように記述されている。

- Auth. mechanism: Certificates
- Ciphersuite negotiation: Yes [4]
- Mutual authentication: Yes [1]
- Integrity protection: Yes [1]
- Replay protection: Yes [1]
- Confidentiality: Yes [2]
- Key derivation: Yes
- Key strength: [3]
- Dictionary attack protection: Yes
- Fast reconnect: Yes
- Crypt. binding: N/A
- Session independence: Yes [1]
- Fragmentation: Yes
- Channel binding: No

5. 安全性に関して知られている結果

5.1. 脅威/脆弱性

EAP-TLS について、現時点で知られている脆弱性はない。

5.2. 形式手法に基づく検証

AVISPA による評価結果が http://www.avispa-project.org/library/EAP_TLS.html に掲載されている。

6. 備考

本文書は、**総務省「暗号・認証技術等を用いた安全な通信環境推進事業に関する実証実験の請負　成果報告書**」からの引用である。