EAP-Archie の Scyhter による評価結果

国立研究開発法人 情報通信研究機構

1. 基本情報

- ✧ 名前

  The EAP Archie Protocol

- ✧ 機能

  PPP 接続において事前共有鍵を用いた相互認証・鍵交換プロトコル。

- ✧ 関連する標準

  Internet Draft (Intended status: Informational)

  http://tools.ietf.org/html/draft-jwalker-eap-archie-01

2. Scyther の文法による記述

2.1. プロトコル仕様

```
secret KEK: Function;
secret KCK: Function;
const AES-CBC-MAC: Function;
protocol EAP-Archie(P, S)
{
  role P
  {
    fresh Np: Nonce;
    fresh Binding: Nonce;
    var Ns: Nonce;
    var SessionID: Nonce;

    recv_1(S, P, (S, SessionID));
    send_2(P, S,
          (SessionID, P, {Np}KEK(S, P), Binding,
          AES-CBC-MAC(KCK(S, P), S, P, {Np}KEK(S, P), Binding)));
```

```
    recv_3(S, P,
          (SessionID, {Ns}KEK(S, P), Binding,
          AES-CBC-MAC(KCK(S, P), S, P, {Np}KEK(S, P), SessionID,
          {Ns}KEK(S, P), Binding)));
    send_4(P, S, (SessionID, AES-CBC-MAC(KCK(S, P), SessionID)));
  }

  role S
  {
    fresh Ns: Nonce;
    fresh SessionID: Nonce;
    var Np: Nonce;
    var Binding: Nonce;

    send_1(S, P, (S, SessionID));
    recv_2(P, S,
          (SessionID, P, {Np}KEK(S, P), Binding,
          AES-CBC-MAC(KCK(S, P), S, P, {Np}KEK(S, P), Binding)));
    send_3(S, P,
          (SessionID, {Ns}KEK(S, P), Binding,
          AES-CBC-MAC(KCK(S, P), S, P, {Np}KEK(S, P), SessionID,
{Ns}KEK(S, P), Binding)));
    recv_4(P, S, (SessionID, AES-CBC-MAC(KCK(S, P), SessionID)));
  }
 }
```

2.2. 攻撃者モデル

Scyther はデフォルトで Dolev-Yao モデルを想定しており、特に記載すべき項目はない。

2.3. セキュリティ要件

```
// ロール P のセキュリティ要件
claim_p1(P, Secret, Np);
claim_p2(P, Secret, Ns);
```

```
claim_p3(P, Alive);
claim_p4(P, Weakagree);
claim_p5(P, Niagree);
claim_p6(P, Nisynch);
// ロールSのセキュリティ要件
claim_s1(S, Secret, Np);
claim_s2(S, Secret, Ns);
claim_s3(S, Alive);
claim_s4(S, Weakagree);
claim_s5(S, Niagree);
claim_s6(S, Nisynch);
```

3. Scyther による評価結果

3.1. 出力

Scyther で評価可能なセキュリティ要件についての、評価結果は以下のとおりである。攻撃は発見されなかった。

```
$ time scyther-macro.exe eap-archie_macro.spdl
claim   EAP-Archie,A    Niagree_A1      -       Ok      [no attack
within bounds]
claim   EAP-Archie,A    Nisynch_A2      -       Ok      [no attack
within bounds]
claim    EAP-Archie,A     SKR_A3   prf(kdf(k(A,P),KeySessKey),Na,Np)
Ok      [no attack within bounds]
claim   EAP-Archie,P    Niagree_P1      -       Ok      [proof of
correctness]
claim   EAP-Archie,P    Nisynch_P2      -       Ok      [proof of
correctness]
claim    EAP-Archie,P     SKR_P3   prf(kdf(k(A,P),KeySessKey),Na,Np)
Ok      [proof of correctness]
```

4. 形式化

4.1. 方針

IETF の通信プロトコルでは、ペイロードの階層ごとに、ペイロードタイプ、ペイロード長などが記載されている。しかし、これらの情報は冗長であり、また、Scyther では長さなどの情報をチェックすることはできない。そこで、メッセージのペイロードを特定するための、最低限のメッセージタイプ情報は残し、それ以外の認証に関係ないと思われる情報はモデルから削除した。Reserved フィールドは省略した。Type フィールドの情報は MsgID フィールドで上書きされるので省略した。

EAP プロトコルでは、Identifier フィールドを用いて Request と Response を 1 対 1 対応させている。しかし、Scyther ではメッセージのタイプチェックで不整合があるとエラーを返す。すなわち、ペイロードが異なるメッセージは自動的に判別する。EAP-Archie ではペイロードが同じになるようなメッセージがないため、Identifier フィールドは省略した。

EAP-Archie プロトコルでは､MAC の計算に用いる関数を AES-CBC-MAC-96 を用いているが、Scyther ではアルゴリズム個別の特徴を捉えることができないので関数名を省略し、また、入力についても、鍵とそれ以外の情報を区別していない。

Binding の値はピアとサーバのロール名で代用した。

4.2. 妥当性

抽象化は行っているが、情報が大きく損なわれたり、特殊なデータ変換を用いたりはしていないので、妥当な形式化であると考える。

4.3. 検証ツールとの相性

プロトコル仕様、攻撃者モデルを記述するにあたって、特に制限はなかった。

セキュリティ要件を記述するにあたって、Scyther では、鍵長など数値化された情報を記述することができない。同様に辞書攻撃について評価することはできない。これは暗号プロトコルではなく、暗号プリミティブの安全性として評価されるべき項目である。データの完全性、秘匿性は本評価の対象である認証プロトコルの範囲外であるため、評価を行っていない。

相互認証については、暗号プロトコルが満たすべき性質が記載されていない。しかし、リプレイ攻撃に対する耐性を謳っていること、鍵共有を目的としていることから、injective agreement が達成されるべきセキュリティ要件であるとみなした。

Scyther では、injective な性質について評価ができないため、non-injective agreement よりも強い性質である non-synchronization について評価を行った。また、セッション鍵はナンス Na, Np と事前共有鍵 k(A,P)から生成されるため、これらナンスの共有が成功していれば、セッション鍵は秘密裡に共有できたと考える。これについては、non-injective

agreement が成立していることで確認できる。

4.4. 検証ツール適用時の性能

検証時間は 0.2 秒だった。実行環境は以下のとおり。

- CPU：AMD Phenom X4 9750B (2.4GHz)
- メモリ：1.7GB

5. 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。