PKMのScyther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

PKM (Privacy and Key Management Protocol)

✧ 機能

Wimaｘ通信における端末 (SS/MS) と基地局 (BS) 間の認証・鍵交換プロトコル。

✧ 関連する標準

IEEE Std 802.16e-2005

(http://standards.ieee.org/getieee802/download/802.16e-2005.pdf)

2. Scytherの文法による記述

2.1. プロトコル仕様

```
usertype String;
usertype Time;
hashfunction hmac;
hashfunction SHA;
macro KEK = SHA(AK);
const choice: Function;
const CryptoSuites: Function;
macro Cert-SS = pk(SS);
macro AuthRequest = (Cert-SS, SS); //12.1.1.4.10
macro AuthReply = ({AK, PS}pk(SS), sqn, BS); //12.1.1.4.13
macro KeyRequest = (sqn, hmac(sqn, AK)); //12.1.1.4.11
macro KeyReply-payload = (sqn, IV, {TEK, IV}KEK);
macro KeyReply = (KeyReply-payload, hmac(KeyReply-payload, AK));
//12.1.1.4.15
//////////////////////////////////////////////////////
protocol PKMv1(SS, BS)
```

```
{
  /*************************************************/
  role SS //peer
    {
      /*************************************************/
      // variables
      // AK exchange
      var AK: Nonce; //authentication key
      var sqn: Nonce; //key sequence number
      var PS: Nonce; //random number used in PKCS#1 v2.1

      // TEK exchange
      var TEK: Nonce; //traffic encryption key
      var IV: Nonce; //IV for CBC-encryption mode

      /*************************************************/
      // sequence
      //---- 7.2.1: SS authorization and AK exchange ----
//       send_1(SS,BS, AuthInfo);
      send_2(SS,BS, AuthRequest);
      recv_3(BS,SS, AuthReply);
      //---- 7.2.2: TEK exchange
      send_4(SS,BS, KeyRequest);
      recv_5(BS,SS, KeyReply);

      /*************************************************/
      // security properties
    }
  /*************************************************/
  role BS //server
    {
      /*************************************************/
```

```
      // variables
      // AK exchange
      fresh AK: Nonce; //authentication key
      fresh sqn: Nonce; //key sequence number
      fresh PS: Nonce; //random number used in PKCS#1 v2.1

      // TEK exchange
      fresh TEK: Nonce; //traffic encryption key
      fresh IV: Nonce; //IV for CBC-encryption mode

      /*********************************************/
      // sequence
      //---- 7.2.1: SS authorization and AK exchange ----
//      recv_1(SS,BS, AuthInfo);
      recv_2(SS,BS, AuthRequest);
      send_3(BS,SS, AuthReply);
      //---- 7.2.2: TEK exchange
      recv_4(SS,BS, KeyRequest);
      claim(BS, Running, SS, AK);
      claim(BS, Running, SS, TEK);
      send_5(BS,SS, KeyReply);

      /*********************************************/
      // security properties

    }
}
```

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

本評価では、ロール BS によるロール SS の認証だけではなく、両側認証としてロール SS に関する性質についても評価している。

3. Scyther による評価結果

3.1. 出力

暗号プロトコルの実行セッション数に制限をおいた評価（すなわち bounded）で、weak agreement への攻撃の可能性が指摘されている。

```
claim   PKMv1,SS        SKR_SS1 AK      Fail    [at least 2 attacks]
claim   PKMv1,SS        SKR_SS2 TEK     Fail    [at least 1 attack]
claim   PKMv1,SS        Alive_SS3       -       Fail    [at least 1
attack]
claim   PKMv1,BS        SKR_BS3 AK      Ok      [no attack within bounds]
claim   PKMv1,BS        SKR_BS4 TEK     Ok      [no attack within bounds]
claim   PKMv1,BS        Alive_BS5       -       Ok      [no attack within
bounds]
claim   PKMv1,BS        Weakagree_BS6   -       Fail    [at least 1
attack]
claim   PKMv1,BS        Commit_BS7      (SS,AK) Fail    [at least 1
attack]
claim   PKMv1,BS        Commit_BS8      (SS,TEK)        Fail    [at least
1 attack]
```

3.2. 攻撃の解説

図 1 は PKMv1 がロール BS について weak agreement を満たさない例である。ここで、ロール SS（Bob）はロール BS が Charlie だと想定して暗号プロトコルを実行しており、直接ロール BS と通信しているわけではない。ただし、ペイロードの改ざんがされるわけではないので、攻撃としての意味はない。

4. 形式化

4.1. 方針

AuthInfo メッセージはロール BS が知らない認証局が作成している可能性がある証明書を送付しており、暗号プロトコルとしては無意味であるため省略した。

PKM では、1 回の鍵交換で 2 つの鍵 TEK を交換する（うち 1 つは前回交換したもの)、評価時間が大幅に増加する上、1 回の鍵交換処理では妥当な形式化が難しいため、1 つの鍵 TEK を交換する形式化とした。

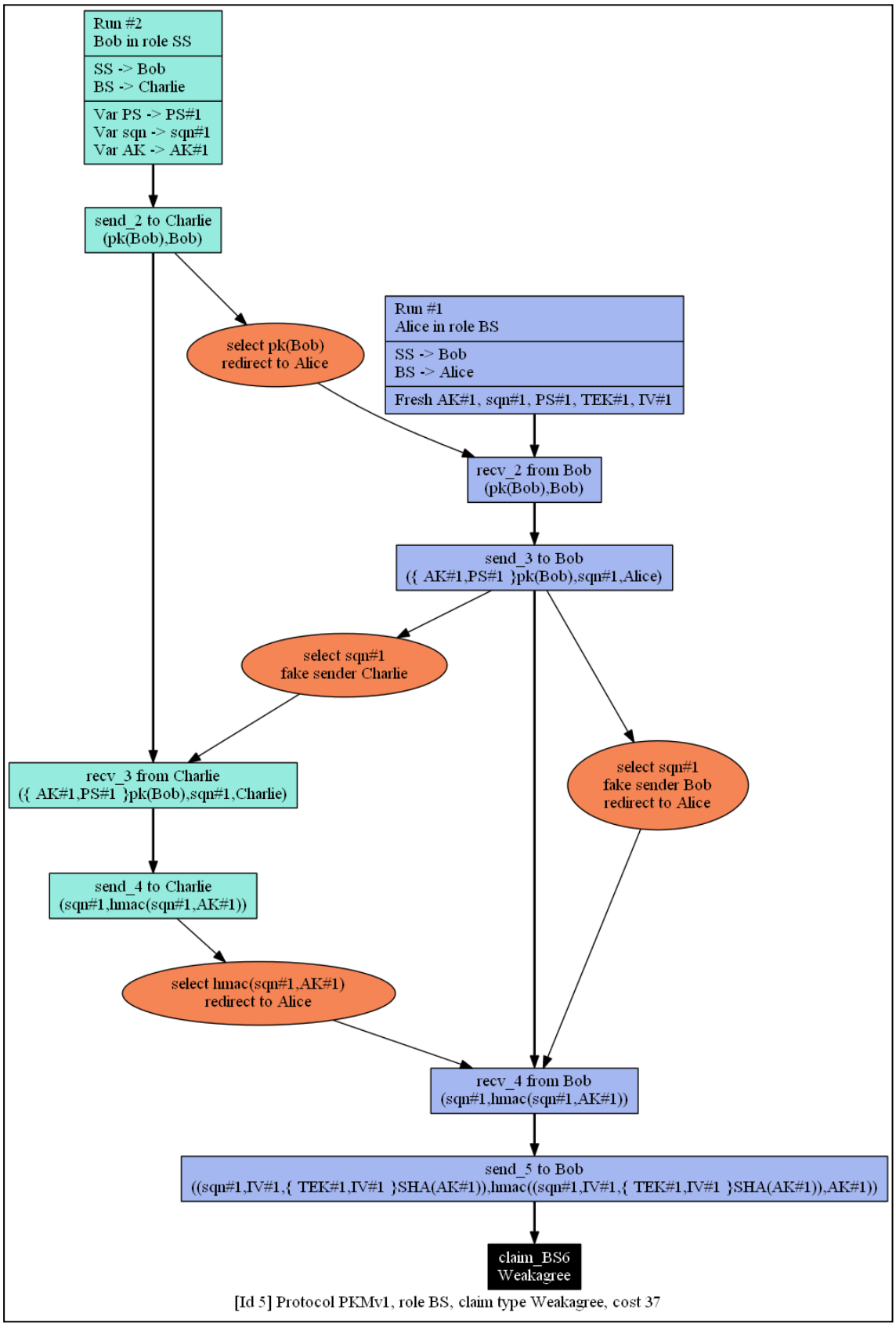
Run #2
Bob in role SS
SS -> Bob
BS -> Charlie
Var PS -> PS#1
Var sqn -> sqn#1
Var AK -> AK#1
send_2 to Charlie
(pk(Bob),Bob)
select pk(Bob)
redirect to Alice
Run #1
Alice in role BS
SS -> Bob
BS -> Alice
Fresh AK#1, sqn#1, PS#1, TEK#1, IV#1
recv_2 from Bob
(pk(Bob),Bob)
send_3 to Bob
({ AK#1,PS#1 }pk(Bob),sqn#1,Alice)
select sqn#1
fake sender Charlie
recv_3 from Charlie
({ AK#1,PS#1 }pk(Bob),sqn#1,Charlie)
select sqn#1
fake sender Bob
redirect to Alice
send_4 to Charlie
(sqn#1,hmac(sqn#1,AK#1))
select hmac(sqn#1,AK#1)
redirect to Alice
recv_4 from Bob
(sqn#1,hmac(sqn#1,AK#1))
send_5 to Bob
((sqn#1,IV#1,{ TEK#1,IV#1 }SHA(AK#1)),hmac((sqn#1,IV#1,{ TEK#1,IV#1 }SHA(AK#1)),AK#1))
claim_BS6
Weakagree
[Id 5] Protocol PKMv1, role BS, claim type Weakagree, cost 37

図 1. 攻撃に関する解説

4.2. 妥当性

交換するセッション鍵を 1 つとしたため、複数のセッションをまたがる場合の暗号プロトコルの安全性については妥当な評価となっていない可能性がある。ただし、PKMv1 では完全ななりすましが可能であり、このモデルの単純化が評価結果に大きな影響を与えることはないと考える。

4.3. 検証ツールとの相性

プロトコル仕様の記述において、Scyther はロール BS が知らない（可能性がある）認証局をモデル化することができない。しかし、これが評価結果に影響を与える可能性は小さい。攻撃者モデル、セキュリティ要件を記述するにあたって、特に制限はなかった。

4.4. 検証ツール適用時の性能

検証時間は 3 分 14 秒だった。実行環境は以下のとおり。

- CPU：Intel Core2Duo E8400 3.0GHz
- メモリ：4GB
- OS：Windows7 32-bit 版

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。