PKMv2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 Version 2)

- ✧ 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 Number;
hashfunction hmac;
//---- message 2: SS->BS
macro Cert-SS = pk(SS);
macro AuthRequest-payload = (Cert-SS, Ns);
macro AuthRequest = (AuthRequest-payload, {AuthRequest-payload}sk(SS));
//---- message3: BS->SS
macro AuthReply-payload = (Ns, Nb, {prePAK}pk(SS), pk(BS));
macro AuthReply = (AuthReply-payload, {AuthReply-payload}sk(SS));
//---- message 4: SS->BS
macro AuthComp = (Nb, {Nb}sk(SS));
//---- key derivation
hashfunction Dot16KDF;
hashfunction SHA;
macro AK = Dot16KDF(prePAK, SS);
```

```
macro KEK = SHA(AK);
//---- message 5: BS->SS
macro TekChallenge-payload = (BSrandom, sqn);
macro TekChallenge = (TekChallenge-payload, hmac(TekChallenge-payload,
AK));
//---- message 6: SS->BS
macro KeyRequest-payload = (SSrandom, BSrandom, sqn);
macro KeyRequest = (KeyRequest-payload, hmac(KeyRequest-payload, AK));
//---- message 7: BS->SS
macro KeyReply-payload = (SSrandom, BSrandom, sqn, {TEK}KEK);
macro KeyReply = (KeyReply-payload, hmac(KeyReply-payload, AK));
////////////////////////////////////////////////////////
protocol PKMv2(SS, BS)
{
  /*************************************************/
  role SS //peer
    {
      /*************************************************/
      // variables
      // AK exchange
      fresh Ns: Nonce;
      var Nb: Nonce;
      var prePAK: Nonce;

      // TEK exchange
      var sqn: Number; //key sequence number
      var BSrandom: Nonce;
      fresh SSrandom: Nonce;
      var TEK: Nonce; //traffic encryption key

      /*************************************************/
      // sequence
```

```
      //---- 7.2.1: SS authorization and AK exchange ----
//      send_1(SS,BS, AuthInfo);
      send_2(SS,BS, AuthRequest);
      recv_3(BS,SS, AuthReply);
      send_4(SS,BS, AuthComp);
      //---- 7.2.2: TEK exchange
      recv_5(BS,SS, TekChallenge);
      claim(SS, Running, BS, AK);
      send_6(SS,BS, KeyRequest);
      recv_7(BS,SS, KeyReply);
      claim(SS, Running, BS, TEK);

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

    }
  /*************************************************/
  role BS //server
    {
      /***********************************************/
      // variables
      // AK exchange
      var Ns: Nonce;
      fresh Nb: Nonce;
      fresh prePAK: Nonce;

      // TEK exchange
      fresh sqn: Number; //key sequence number
      fresh BSrandom: Nonce;
      var SSrandom: Nonce;
      fresh TEK: Nonce; //traffic encryption key
```

```
      /*********************************************/
      // sequence
      //---- 7.2.1: SS authorization and AK exchange ----
//      recv_1(SS,BS, AuthInfo);
      recv_2(SS,BS, AuthRequest);
      send_3(BS,SS, AuthReply);
      recv_4(SS,BS, AuthComp);
      //---- 7.2.2: TEK exchange
      send_5(BS,SS, TekChallenge);
      recv_6(SS,BS, KeyRequest);
      claim(BS, Running, SS, AK);
      claim(BS, Running, SS, TEK);
      send_7(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

```
// ロール SS のセキュリティ要件
claim(SS, SKR, AK);
claim(SS, SKR, TEK);
claim(SS, Alive);
claim(SS, Weakagree);
claim(SS, Commit, BS, AK);
claim(SS, Commit, BS, TEK);
// ロール BS のセキュリティ要件
claim(BS, SKR, AK);
claim(BS, SKR, TEK);
claim(BS, Alive);
```

```
claim(BS, Weakagree);
claim(BS, Commit, SS, AK);
claim(BS, Commit, SS, TEK);
```

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。すなわち、PKMv2 の認証（＋鍵 TEK の共有）は、ロール BS の鍵 TEK に関する agreement を満たさない。

```
claim    PKMv2,SS SKR_SS3  Dot16KDF(prePAK,SS)        Ok [no attack
within bounds]
claim    PKMv2,SS SKR_SS4  TEK       Ok      [no attack within
bounds]
claim    PKMv2,SS Alive_SS5-         Ok      [no attack within
bounds]
claim    PKMv2,SS Weakagree_SS6     -         Ok      [no attack
within bounds]
claim    PKMv2,SS Commit_SS7        (BS,Dot16KDF(prePAK,SS))    Ok
         [no attack within bounds]
claim    PKMv2,SS Commit_SS8        (BS,TEK)  Ok      [no attack
within bounds]
claim    PKMv2,BS SKR_BS3  Dot16KDF(prePAK,SS)         Ok      [no
attack within bounds]
claim    PKMv2,BS SKR_BS4  TEK       Ok      [no attack within
bounds]
claim    PKMv2,BS Alive_BS5-         Ok      [proof of correctness]
claim    PKMv2,BS Weakagree_BS6     -         Ok      [no attack
within bounds]
claim    PKMv2,BS Commit_BS7        (SS,Dot16KDF(prePAK,SS))    Ok
         [no attack within bounds]
claim    PKMv2,BS Commit_BS8        (SS,TEK)  Fail    [at least 3
attacks]
```

3.2. 攻撃の解説

図 1 は PKMv2 がロール BS について weak agreement を満たさない例である。攻撃者、通信路上を流れたメッセージの一部を用いて、以降のメッセージの改ざんを行っている。しかし、ペイロードの値が変わるわけではないので、この攻撃は本質的なものではないと考えられる。実際、すべてのデータに関する agreement を調べる claim（Niagree）で評価を行ったところ、攻撃は見つからなかった。

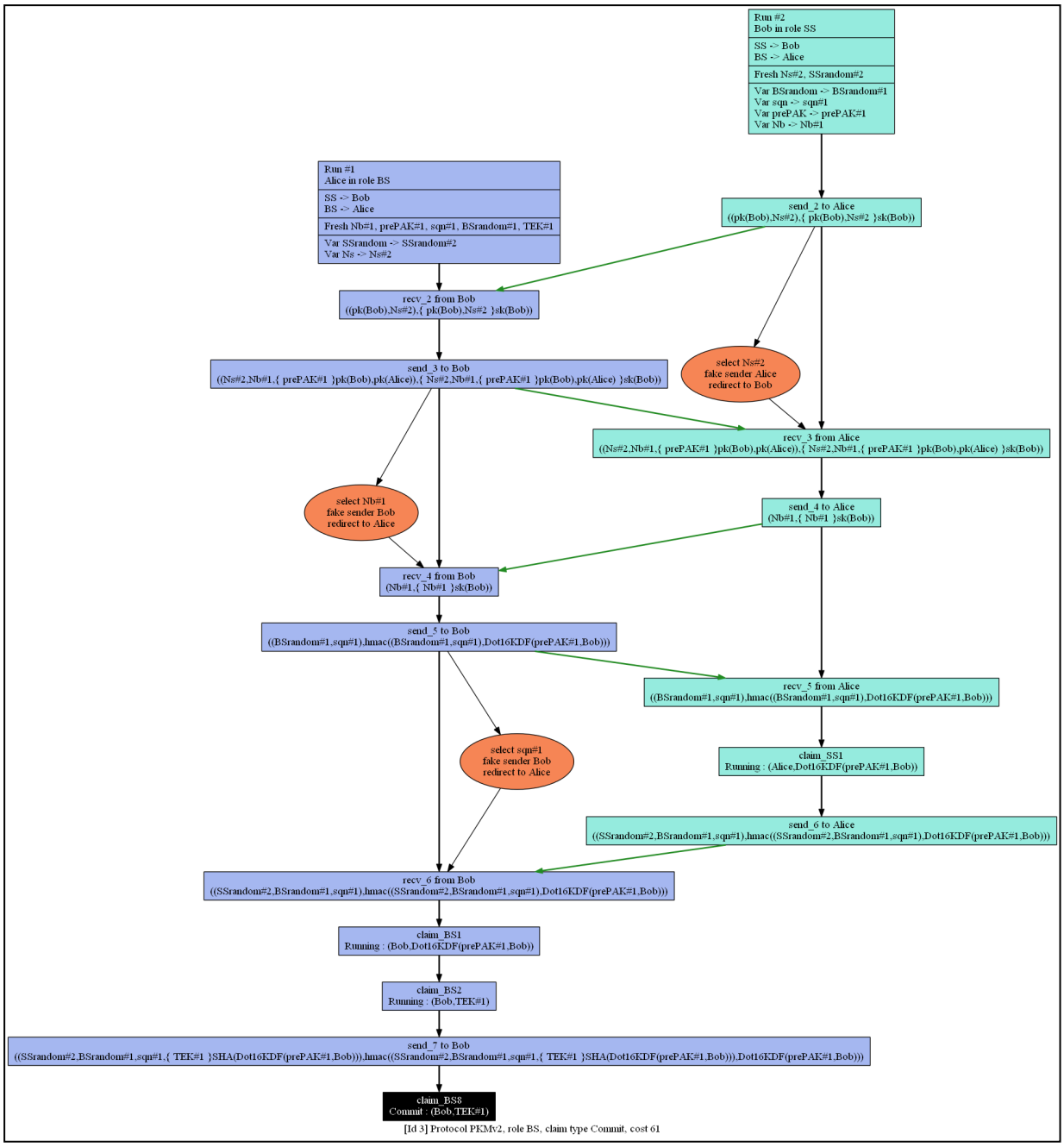

図. 1　攻撃に関する解説

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。

SAID, Capabilities, KeyLifetime は省略した。

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。

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

検証時間は約 4 時間 30 分だった。実行環境は以下のとおり。

- CPU：Intel Xeon E5502 1.87GHz
- メモリ：48GB
- OS：Ubuntu Linux 9 64-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。