PANA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

  Protocol for Carrying Authentication for Network Access (PANA)

- ✧ 機能

  UDP/IP 上でのネットワークアクセス認証・鍵交換プロトコルであり、Extensible Authentication Protocol (EAP) のメッセージを運ぶ。

- ✧ 関連する標準

  RFC5191 (https://tools.ietf.org/html/rfc5191)

2. Scyther の文法による記述

本文書では、EAP-AKA が利用されているとする。

2.1. プロトコル仕様

```
hashfunction prf, kdf, mac;
hashfunction h;
const inc,dec: Function;
inversekeys (inc,dec);
const empty;
usertype EAP-Msg-Code;
usertype EAP-Req-Resp-Type;
const EAP-Request,EAP-Response,EAP-Success,EAP-Failure: EAP-Msg-Code;
const Identity: EAP-Req-Resp-Type;
hashfunction f1, f2, f3, f4, f5;
hashfunction prf1, prf2;
const Add: Function;
const Sub: Function;
inversekeys(Add, Sub);
const aka-method: EAP-Req-Resp-Type;
```

```
usertype PANA-Msg-Code;
usertype PANA-Req-Resp-Type;

const           PANA-Request,PANA-Response,PANA-Success,PANA-Failure:
PANA-Msg-Code;

protocol pana-auth-aka(I, R)
{
 role I {
         fresh rand, sqn: Nonce;
fresh msgnum: Nonce;
fresh PANA-msgnum, PAA-nonce: Nonce;
var PaC-nonce: Nonce;

send_!0p(I,I,PANA-msgnum);
send_1(I,R,PANA-Request,PANA-msgnum);
recv_2(R,I,PANA-Response,PANA-msgnum);
send_3(I,R,PANA-Request,inc(PANA-msgnum),PAA-nonce,EAP-Request,msgnu
m,Identity,I);
recv_4(R,I,PANA-Response,inc(PANA-msgnum),PaC-nonce,EAP-Response,msg
num,Identity,R);
claim(I,Running,R,rand, sqn);

send_5(I,R,PANA-Request,inc(inc(PANA-msgnum)),EAP-Request,inc(msgnum
),aka-method,rand, Add(sqn, f5(k(I,R), rand)), f1(k(I,R), sqn, rand));
recv_6(R,I,PANA-Response,inc(inc(PANA-msgnum)),EAP-Response,inc(msgn
um),aka-method,f2(k(I,R), rand));
send_7(I,R,PANA-Request,inc(inc(inc(PANA-msgnum))),PANA-Success,EAP-
Success,inc(inc(msgnum)),h(prf(prf2(h(R,sqn,rand,h(R,f4(k(I,R),
rand),f3(k(I,R),rand)))),PANA-msgnum,PaC-nonce,PAA-nonce),PANA-Reque
st,inc(inc(inc(PANA-msgnum))),PANA-Success,EAP-Success,inc(inc(msgnu
```

```
m))));
recv_8(R,I,PANA-Response,inc(inc(inc(PANA-msgnum))),h(prf(prf2(h(R,s
qn,rand,h(R,f4(k(I,R),rand),f3(k(I,R),rand)))),PANA-msgnum,PaC-nonce
,PAA-nonce),PANA-Response,inc(inc(inc(PANA-msgnum)))));
}

 role R {
var rand, sqn: Nonce;
var msgnum: Nonce;
var PANA-msgnum, PAA-nonce: Nonce;
fresh PaC-nonce: Nonce;

recv_1(I,R,PANA-Request,PANA-msgnum);
send_2(R,I,PANA-Response,PANA-msgnum);
recv_3(I,R,PANA-Request,inc(PANA-msgnum),PAA-nonce,EAP-Request,msgnu
m,Identity,I);
send_4(R,I,PANA-Response,inc(PANA-msgnum),PaC-nonce,EAP-Response,msg
num,Identity,R);
recv_5(I,R,PANA-Request,inc(inc(PANA-msgnum)),EAP-Request,inc(msgnum
),aka-method,rand, Add(sqn, f5(k(I,R), rand)), f1(k(I,R), sqn, rand));
claim(R,Running,I,rand, sqn);
send_6(R,I,PANA-Response,inc(inc(PANA-msgnum)),EAP-Response,inc(msgn
um),aka-method,f2(k(I,R), rand));
recv_7(I,R,PANA-Request,inc(inc(inc(PANA-msgnum))),PANA-Success,EAP-
Success,inc(inc(msgnum)),h(prf(prf2(h(R,sqn,rand,h(R,f4(k(I,R),
rand),f3(k(I,R),
rand)))),PANA-msgnum,PaC-nonce,PAA-nonce),PANA-Request,inc(inc(inc(P
ANA-msgnum))),PANA-Success,EAP-Success,inc(inc(msgnum))));
send_8(R,I,PANA-Response,inc(inc(inc(PANA-msgnum))), h(prf(prf2(h(R,
sqn, rand, h(R, f4(k(I,R), rand), f3(k(I,R), rand)))), PANA-msgnum,
PaC-nonce, PAA-nonce), PANA-Response, inc(inc(inc(PANA-msgnum)))));
}
```

```
}
```

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 R（サーバ）のセキュリティ要件
        claim(R, Secret, prf2(h(R, sqn, rand ,h(R, f4(k(I,R), rand),
f3(k(I,R), rand)))));
    claim(R,Commit,I,rand, sqn);
    claim(R,Secret, k(I,R));
    claim(R,Alive);
    claim(R,Weakagree);
    claim(R,Niagree);
//ロール I（ピア）のセキュリティ要件
        claim(I,Secret,prf2(h(R,sqn,rand,h(R,f4(k(I,R),
rand),f3(k(I,R), rand)))));
    claim(I,Commit,R,rand, sqn);
    claim(I,Secret, k(I,R));
    claim(I,Alive);
    claim(I,Weakagree);
    claim(I,Niagree);
```

なお、両ロールともに、Commit に対応する Running クレイムをシーケンス中に埋め込んだ。Running クレイムは埋め込み箇所によって意味が変わるため、別に抜き出すことはしていない。

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）ではあるが、攻撃は発見されなかった。

```
claim    pana-auth-aka,I Secret_I2
prf2(h(R,sqn,rand,h(R,f4(k(I,R),rand),f3(k(I,R),rand))))          Ok
[no attack within bounds]
claim    pana-auth-aka,I Commit_I3          (R,rand,sqn)      Ok        [no
```

```
attack within bounds]
claim    pana-auth-aka,I Secret_I4        k(I,R)  Ok       [no attack
within bounds]
claim    pana-auth-aka,I Alive_I5         -       Ok       [no attack
within bounds]
claim    pana-auth-aka,I Weakagree_I6     -       Ok       [no attack
within bounds]
claim    pana-auth-aka,I Niagree_I7       -       Ok       [no attack
within bounds]
claim                                pana-auth-aka,R          Secret_R2
prf2(h(R,sqn,rand,h(R,f4(k(I,R),rand),f3(k(I,R),rand))))          Ok
[no attack within bounds]
claim    pana-auth-aka,R Commit_R3        (I,rand,sqn)    Ok       [no
attack within bounds]
claim    pana-auth-aka,R Secret_R4        k(I,R)  Ok       [no attack
within bounds]
claim    pana-auth-aka,R Alive_R5         -       Ok       [no attack
within bounds]
claim    pana-auth-aka,R Weakagree_R6     -       Ok       [no attack
within bounds]
claim    pana-auth-aka,R Niagree_R7       -       Ok       [no attack
within bounds]
```

3.2. 攻撃の解説

前述のとおり、bounded ではあるが、攻撃は発見されなかった。

3.3. 備考

PANA は相互認証を目的としており、認証方式自体は EAP に依存している。しかし、EAP の中には、片側認証を目的とした暗号プロトコル（たとえば EAP-MD5-Challenge）がある。誤って PANA の中でこれらの片側認証プロトコルを用いた場合、システムが脆弱となる可能性がある。

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形式化から削除した。

4.2. 妥当性

抽象化はしているが、情報が大きく損なわれるような形式化は行っていない。Scyther の開発者らは、DH 鍵交換を上記の記述のように形式化することを提案しているが、この形式化がどの程度正確なのかについては分かっていない。ただし、既存の結果では、上述の形式化で得られた結果と他のツールでの評価結果が食い違っているケースはないと思われる。

4.3. 検証ツールとの相性

プロトコル仕様、攻撃者モデル、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 28 分 48 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。