Yahalom 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

Yahalom

✧ 機能

共通鍵暗号の鍵サーバを用いた相互認証・鍵交換プロトコル。

✧ 関連する文書

M.Burrows, M.Abadi, and R.Needham, “A Logic of Authentication,” Research Report 39, Digital Equipment Corp. Systems Research Center, Feb. 1989.

2. Scyther の文法による記述

2.1. プロトコル仕様

```
protocol Yahalom(A, B, S)
{
        role A
        {
                fresh Na: Nonce;
                var Kab:Nonce;
                var Nb: Nonce;
                send_1(A, B, (A, Na));
                recv_3(S, A, ({B, Kab, Na, Nb}k(A, S), {A, Kab}k(B,
S)));
                send_4(A, B, ({A, Kab}k(B, S), {Nb}Kab));
        }
        role B
        {
                fresh Nb: Nonce;
                var Na: Nonce;
```

```
                var Kab: Nonce;
                recv_1(A, B, (A, Na));
                send_2(B, S, (B, {A, Na, Nb}k(B, S)));
                recv_4(A, B, ({A, Kab}k(B, S), {Nb}Kab));
        }
        role S
        {
                freash Kab: Nonce;
                var Nb: Nonce;
                var Na: Nonce;
                recv_2(B, S, (B, {A, Na, Nb}k(B, S)));
                send_3(S, A, ({B, Kab, Na, Nb}k(A, S), {A, Kab}k(B,
S)));
        }
}
```

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 A のセキュリティ要件
claim_a1(A, Secret, Kab);
claim_a2(A, Alive);
claim_a3(A, Weakagree);
claim_a4(A, Niagree);
claim_a5(A, Nisynch);
// ロール B のセキュリティ要件
claim_b1(B, Secret, Kab);
claim_b2(B, Alive);
claim_b3(B, Weakagree);
claim_b4(B, Niagree);
claim_b5(B, Nisynch);
// ロール S のセキュリティ要件
claim_s1(S, Secret, Kab);
```

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、攻撃は発見されなかった。すなわち、Yahalom では、鍵 Kab の共有は成功しており、認証については、ロール A、ロール B ともに non-injective synchronization を満たす。

```
claim id [Yahalom,a1], Secret(Kab)   : No attacks within bounds.
claim id [Yahalom,a2], Alive                 : No attacks.
claim id [Yahalom,a3], Weakagree     : No attacks.
claim id [Yahalom,a4], Niagree               : No attacks.
claim id [Yahalom,a5], Nisynch               : No attacks.
claim id [Yahalom,b1], Secret(Kab)   : No attacks within bounds.
claim id [Yahalom,b2], Alive                 :  No  attacks  within
bounds.
claim id [Yahalom,b3], Weakagree     : No attacks within bounds.
claim id [Yahalom,b4], Niagree               :  No  attacks  within
bounds.
claim id [Yahalom,b5], Nisynch               :  No  attacks  within
bounds.
claim id [Yahalom,s1], Secret(Kab)   : No attacks.
```

3.2. 攻撃の解説

前述のとおり、攻撃は発見されなかった。

4. 形式化

4.1. 方針

Yahalom は学術的な暗号プロトコルであり、元論文に記載された当初からモデル化されていた。このため、形式化で特筆すべき事項はない。

4.2. 妥当性

同上。

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、BAN ロジックで記述された性質とまったく同じ性質を調べることは難しい。おおよそで言えば、Scyther で評価できるセキ

ュリティプロパティでは以下が該当すると思われる。

- ロール A, ロール B にとって鍵 Kab が secrecy を満たすこと。
- ロール A, ロール B にとって、non-injective agreement を満たすこと。

その一方で、Yahalom は学術的な暗号プロトコルであるため、本評価では鍵 Kab の secrecy 及びロール A, ロール B の認証に関して、Scyther で評価可能なセキュリティプロパティをすべて評価している。

4.4. 検証ツール適用時の性能

検証時間は 0.8 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。