Yahalom-Paulson's modified 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

Paulson's strengthened version of 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

Lawrence C.Paulson, "Relations between secrets: Two formal analyses of the yahalom protocol," J. Computer Security, 2001.

2. Scyther の文法による記述

2.1. プロトコル仕様

```
protocol Yahalom-Paulson's-strengthed(A, B, S)
{
    role A
    {
        fresh Na: Nonce;
        var Kab: Nonce;
        var Nb: Nonce;

        send_1(A, B, (A, Na));
        recv_3(S, A, (Nb, {B, Kab, Na}k(A, S), {A, B, Kab, Nb}k(B, S)));
        send_4(A, B, ({A, B, Kab, Nb}k(B, S), {Nb}Kab));
    }

    role B
    {
        fresh Nb: Nonce;
```

```
        var Kab: Nonce;
          var Na: Nonce;

          recv_1(A, B, (A, Na));
          send_2(B, S, (B, Nb, {A, Na}k(B, S)));
          recv_4(A, B, ({A, B, Kab, Nb}k(B, S), {Nb}Kab));
    }

    role S
    {
          fresh Kab: Nonce;
          var Na: Nonce;
          var Nb: Nonce;

          recv_2(B, S, (B, Nb, {A, Na}k(B, S)));
          send_3(S, A, (Nb, {B, Kab, Na}k(A, S), {A, B, Kab, Nb}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, Nisynch);
claim_a5(A, Niagree);
// 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, Nisynch);
```

```
claim_b5(B, Niagree);
// 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、ロール A, ロール B について weak agreement を満たすが、non-injective agreement を満たさない。すなわち、ロール A, ロール B は互いの通信相手がそれぞれロール B, ロール A であることの確証を得ているが、暗号プロトコルの実行において、共有できていないデータが存在する。

```
claim id [Yahalom-Paulson's-strengthed,a1], Secret(Kab)          :
No attacks within bounds.
claim id [Yahalom-Paulson's-strengthed,a2], Alive       : No attacks.
claim id [Yahalom-Paulson's-strengthed,a3], Weakagree   : No attacks.
claim id [Yahalom-Paulson's-strengthed,a4], Nisynch     : At least 1
attack.
claim id [Yahalom-Paulson's-strengthed,a5], Niagree     : At least 1
attack.
claim id [Yahalom-Paulson's-strengthed,b1], Secret(Kab)          :
No attacks within bounds.
claim id [Yahalom-Paulson's-strengthed,b2], Alive       : No attacks
within bounds.
claim id [Yahalom-Paulson's-strengthed,b3], Weakagree   : No attacks
within bounds.
claim id [Yahalom-Paulson's-strengthed,b4], Nisynch     : At least 1
attack.
claim id [Yahalom-Paulson's-strengthed,b5], Niagree     : At least 1
attack.
claim id [Yahalom-Paulson's-strengthed,s1], Secret(Kab)          :
No attacks within bounds.
```

3.2. 攻撃の解説

以下の図で、Yahalom-Paulson's modified がロール A について non-injective agreement を満たさない例を示している。攻撃者はロール B（Bob）から送信されたデータの一部を改ざんしているが、ロール A（Charlie）はそれと気づかないまま暗号プロトコルを終了している。すなわち、ロール A とロール B はロール B が生成するナンスを共有できていない。

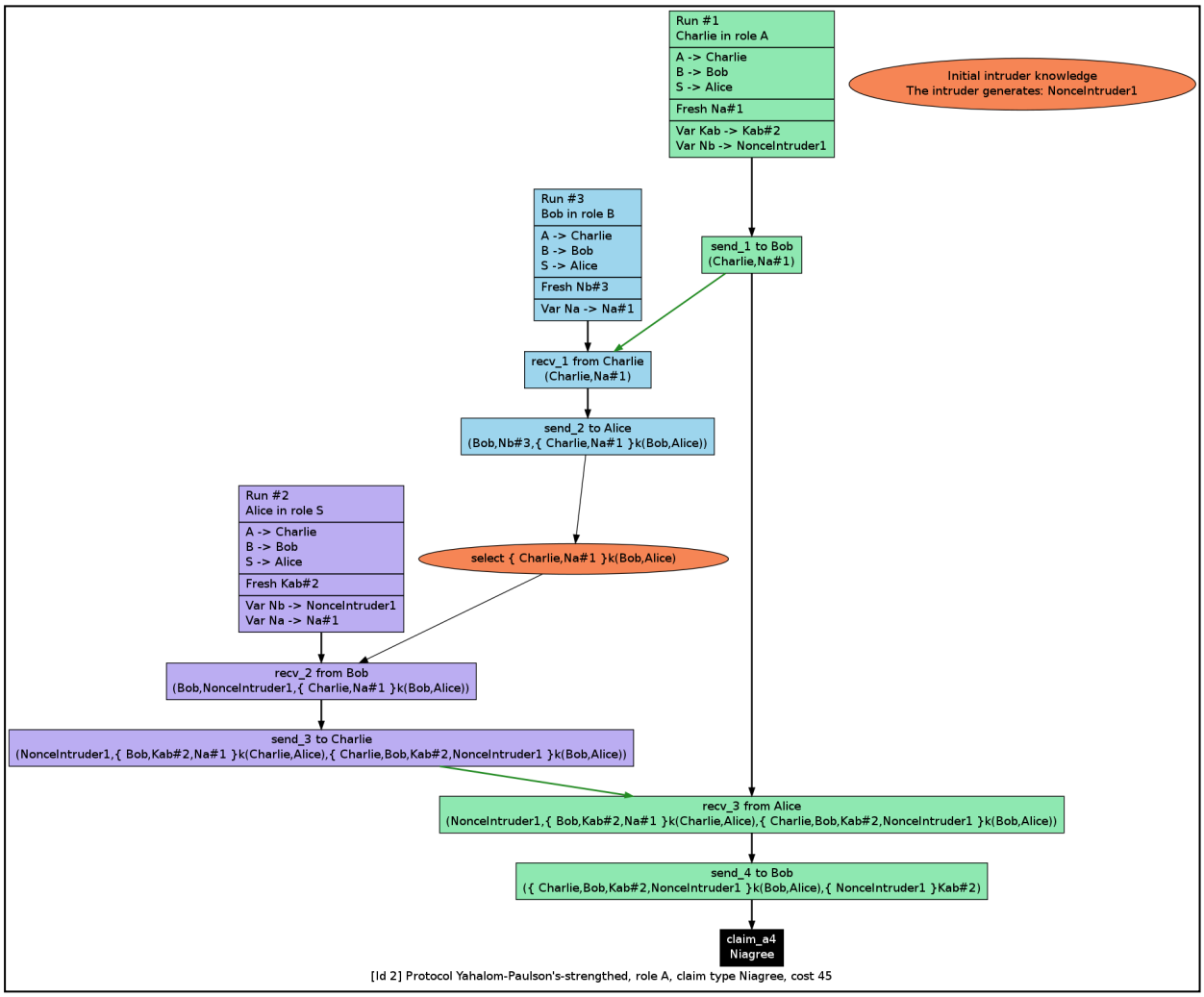

図 1. 攻撃に関する解説

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。