暗号プロトコル評価結果

独立行政法人　情報通信研究機構

1. プロトコル名：Kerberos with ticket caching

2. 関連する標準

IETF：RFC4120 (The Kerberos Network Authentication Service (V5)), July 2005.
http://www.rfc-editor.org/rfc/pdfrfc/rfc4120.txt.pdf

3. 使用したツール：Scyther

4. 評価の概要:Scyther による評価では、セッション鍵の漏洩の可能性が指摘されているが、通常 Kerberos ではサーバ間の階層構造は固定されており、また、サーバが不正な行動をとることは想定されていない。そのため、サーバの不正が無い限りにおいては問題にならない。

5. Scyther による評価

5.1. シーケンス記述

```
usertype Time;
usertype String;
usertype SessionKey;
const hash: Function;
protocol Kerberos-ticket-caching(C, A, G, S)
{
        role C
        {
                fresh N2: Nonce;
                fresh N1: Nonce;
                fresh U: String;
                fresh T2: Time;
                fresh T1: Time;
                var Tstart2: Time;
                var Tstart1: Time;
                var Texpire2: Time;
                var Texpire1: Time;
```

```
                var Tcg: Ticket;
                var Tcs: Ticket;
                var Kcg, Kcs: SessionKey;
                send_1(C, A, (U, G, N1));
                recv_2(A, C, (C, Tcg, {G, Kcg, Tstart1, Texpire1,
N1}k(C, A)));
                send_3(C, G, (S, N2, Tcg, {C, T1}Kcg));
                recv_4(G, C, (U, Tcs, {S, Kcs, Tstart2, Texpire2,
N2}Kcg));
                send_5(C, S, (Tcs, {C, T2}Kcs));
                recv_6(S, C, {T2}Kcs);
        }
        role A
        {
                fresh Tstart1: Time;
                fresh Texpire1: Time;
                fresh Kcg: SessionKey;
                var N1: Nonce;
                var U: String;

                recv_1(C, A, (U, G, N1));
                send_2(A, C,
                        (C,
                        {U, C, G, Kcg, Tstart1, Texpire1}k(A, G),
                        {G, Kcg, Tstart1, Texpire1, N1}k(C, A)));
        }
        role G
        {
                fresh Tstart2: Time;
                fresh Texpire2: Time;
                fresh Kcs: SessionKey;
                var N2: Nonce;
```

```
                var U: String;
                var Tstart1: Time;
                var Texpire1: Time;
                var T1: Time;
                var Kcg: SessionKey;

                recv_3(C, G,
                        (S, N2,
                        {U, C, G, Kcg, Tstart1, Texpire1}k(A, G),
                        {C, T1}Kcg));
                send_4(G, C,
                        (U,
                        {U, C, S, Kcs, Tstart2, Texpire2}k(G, S),
                        {S, Kcs, Tstart2, Texpire2, N2}Kcg));
        }
        role S
        {
                var U: String;
                var Tstart2: Time;
                var Texpire2: Time;
                var T2: Time;
                var Kcs: SessionKey;

                recv_5(C, S,
                        ({U, C, S, Kcs, Tstart2, Texpire2}k(G, S),
{C, T2}Kcs));
                send_6(S, C, {T2}Kcs);
        }
}
```

5.2.攻撃者モデル

仕様には特段の記述が見つからなかったので、今回のモデル化では、Dolev-Yao モデルを想

定した。

5.3.セキュリティプロパティの記述

```
// ロールCのセキュリティプロパティ
claim_C1(C, Secret, Kcg);
claim_C2(C, Secret, Kcs);
claim_C3(C, Alive);
claim_C4(C, Weakagree);
// ロールAのセキュリティプロパティ
claim_A2(A, Alive);
claim_A3(A, Weakagree);
// ロールGのセキュリティプロパティ
claim_G1(G, Secret, Kcg);
claim_G2(G, Alive);
claim_G3(G, Weakagree);
// ロールSのセキュリティプロパティ
claim_S1(S, Secret, Kcs);
claim_S2(S, Alive);
claim_S3(S, Weakagree);
```

5.4.検証結果

○評価ツールの出力

```
claim id [Kerberos-ticket-caching,C1], Secret(Kcg)     : No attacks within bounds.
claim id [Kerberos-ticket-caching,C2], Secret(Kcs)     : No attacks within bounds.
claim id [Kerberos-ticket-caching,C3], Alive : No attacks within bounds.
claim id [Kerberos-ticket-caching,C4], Weakagree       : At least 1 attack.
claim id [Kerberos-ticket-caching,A2], Alive : Exactly 1 attack.
claim id [Kerberos-ticket-caching,A3], Weakagree       : Exactly 1
```

```
attack.
claim id [Kerberos-ticket-caching,G1], Secret(Kcg)     : At least 12
attacks.
claim id [Kerberos-ticket-caching,G2], Alive  : At least 12 attacks.
claim id [Kerberos-ticket-caching,G3], Weakagree       : At least 12
attacks.
claim id [Kerberos-ticket-caching,S1], Secret(Kcs)     : At least 12
attacks.
claim id [Kerberos-ticket-caching,S2], Alive  : At least 12 attacks.
claim id [Kerberos-ticket-caching,S3], Weakagree       : At least 12
attacks.
```

○攻撃に関する解説

次ページの図 1.では、Kerberos with ticket caching プロトコルにおいて、ロール C とロール S で共有されるセッション鍵 Kcs がロール S について Secret を満たさない例を示している。Kerberos プロトコルでは、サーバ間の階層構造は固定されており、また、サーバが不正な行動をとることは想定されていないと思われる。図では、攻撃者がロール A を演じている。ロール A はセッション鍵 Kcs を暗号化している鍵 Kcg を自分で生成できるので、ロール G が生成するセッション鍵 Kcs を入手できる。

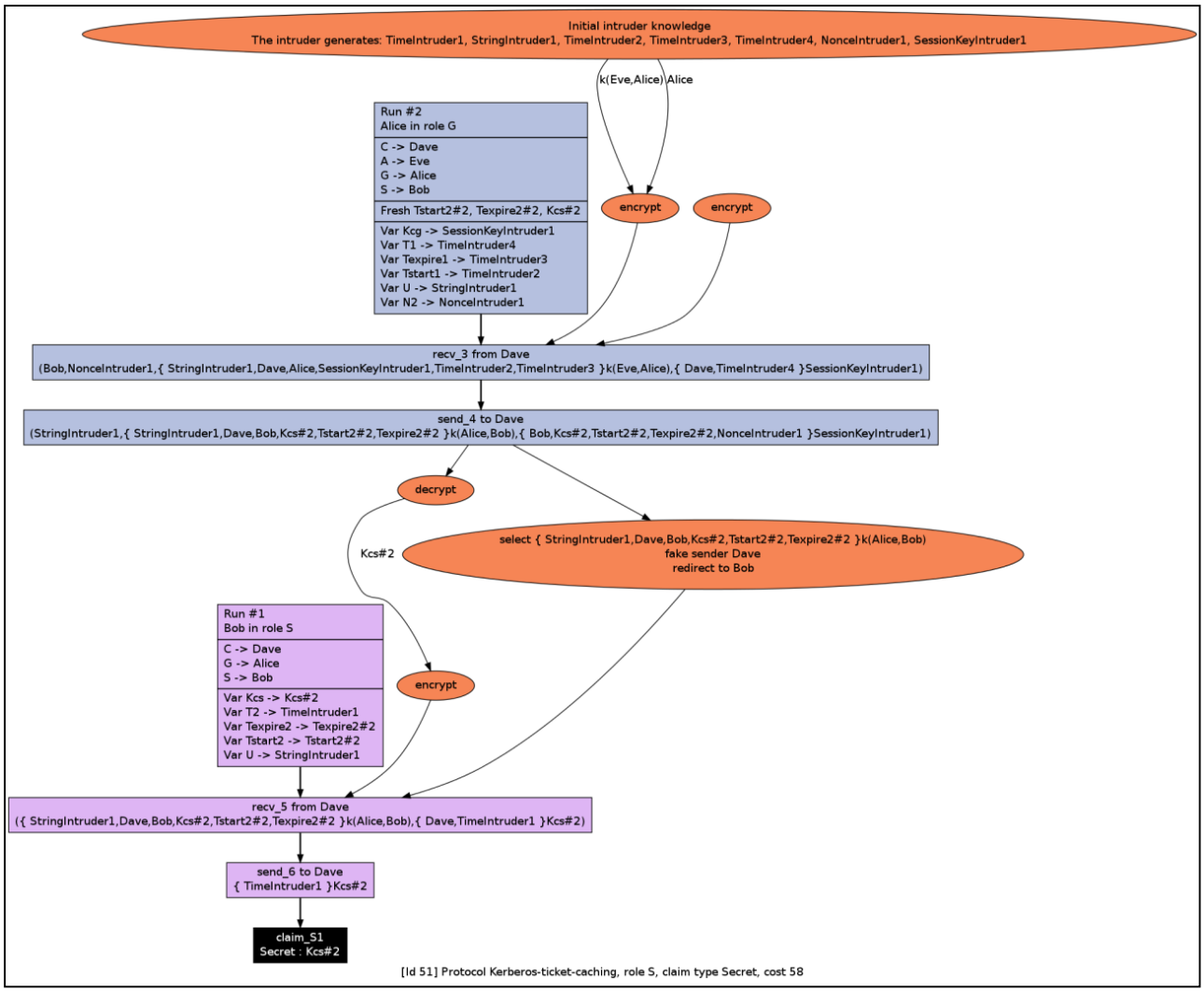

図 1. 攻撃に関する解説

5.5. モデル化

〇モデル化プロセス

Kerberos プロトコルでは、チケットの有効期限を時刻情報で記述している。モデル化では、Time という型を宣言し、時刻情報に関する変数を Time 型として記述した。

5.6. モデル化の妥当性

Kerberos プロトコルでは、サーバの不正行為を想定していないと思われる。Scyther では、攻撃者の能力を制限することができないため、仕様が目的としている安全性を評価できていない可能性がある。

5.7.評価ツールとの相性

〇暗号プロトコルの記述可能性

モデルでは時刻情報についてTimeという型を宣言しているが、時刻の整合性をチェックしているわけではなく、基本的にナンスとして取り扱われる。

〇セキュリティプロパティの記述可能性

暗号プロトコルの目的は、サーバによるクライアントの片側認証及びセッション鍵（チケット）の交換である。したがって、Kcs に関する秘匿性及びクライアント - サーバ間で non-injective agreement が成立していれば良いと思われる。また、RFC4120 中では、時刻情報を盛り込むことで、リプレイ攻撃を防げるとしている。したがって、クライアント - サーバ間で injective agreement が満たされていることが望ましい。

Scyther では、injectivity を評価することができない。しかし、評価の結果、Kerberos with ticket caching プロトコルは Weakagreement を満たさないため、これが問題となることはなかった。

Kerberos プロトコルでは、サーバの不正行為を想定していないと思われる。Scyther では、攻撃者の能力を制限することができないため、仕様が目的としている安全性を評価できていない可能性がある。

5.8.評価ツールの性能

・評価環境

- CPU：AMD Phenom X4 9750B (2.4GHz)
- メモリ：1.7GB

・性能

- 検証時間：約 6 分

5.9.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。