

束縛のタイミングを考慮した認証プロトコルについて

萩谷 昌己†, 齋藤 孝道‡, 溝口 文雄‡

† 東京大学, ‡ 東京理科大学

hagiya@is.s.u-tokyo.ac.jp, saito@is.noda.sut.ac.jp

あらまし

インターネットに対する脅威として DoS (Denial of Service) 攻撃がある。DoS 攻撃を防ぐ観点からすると、サーバ/クライアント間で認証を行う場合、クライアントであるイニシエータのリクエストに対するサーバであるレスポンドのリプライになるべく計算機資源を消費しないものが望ましいことは明らかである。つまり、レスポンドの最初のリプライを簡略化したい。本論文では、このような観点から、プロトコルにおける通報を分割する方法を考察する。この中で特に、束縛のタイミングを考慮して、late binding という概念を導入し、ある種の攻撃に耐性を持つプロトコルを提案する。また、本論文では、この late binding を用いたプロトコルの安全性を示し、この概念を考慮した認証プロトコルの例をいくつか示す。

キーワード 認証, 認証プロトコル, 形式的検証, DoS, 安全性

Authentication Protocol based on Timing of Binding

Masami HAGIYA†, Takamichi SAITO‡, Fumio MIZOGUCHI‡

†University of Tokyo, ‡Science University of Tokyo

hagiya@is.s.u-tokyo.ac.jp, saito@is.noda.sut.ac.jp

Abstract

The DoS (Denial of Service) attack is notorious and threatening in the normal use or management of the internet. For preventing the attack, when authenticating between a server and a client, we need extra stratagems in the protocol: the server's response obviously shouldn't consume its CPU resource as much as possible. Namely, the first response in an authentication protocol should be simplified to process a large amount of initiators' request. In this paper, we consider and propose the method of dividing message in the protocol from such a viewpoint. Especially, we introduce the concept of *late binding*, which is based on timing of binding in a protocol, and design a new protocol tolerant of DoS. Furthermore, we show the safety of the protocols based on late binding, and give the example of protocols.

key words authentication, authentication protocol, formal method, DoS, security

1 はじめに

インターネットに対する脅威として、通報の傍受、改竄、捏造、再送や成り済まし攻撃などがある。認証プロトコルを効果的に用いることにより、これらの攻撃の多くを防ぐことができる。しかしながら、認証プロトコル自体も弱点を有している。すなわち、レスポンドがある種の DoS (Denial of Service) 攻撃に晒される危険性がある。例として、Needham-Schroeder-Lowe 公開鍵プロトコル [5, 4] をあげる：

(通報 0) $A \rightarrow B : \{A, N_A\}_{P_B}$

(通報 1) $B \rightarrow A : \{B, N_A, N_B\}_{P_A}$

(通報 2) $A \rightarrow B : \{N_B\}_{P_B}$

ただし、 A をイニシエータ、 B をレスポンド、 N_A を A のノンス、 N_B を B のノンスとする。 P_A と P_B を、それぞれ A と B の公開鍵とする。

ここで、イニシエータの最初の通報に対するレスポンドの返信のための計算の負荷は、相当に重いものである。悪意を持った主体がイニシエータとして、認証を行うつもりもなく、このプロトコルの最初の通報を一定時間内に大量に送りつけた場合、レスポンド側ではシステムダウンの可能性がある。

このような観点からすると、認証プロトコルを用いる場合、イニシエータのリクエストに対するレスポンドのリプライのための計算コストが低い方が望ましいことは明らかである。つまり、レスポンドの最初のリプライを簡略化したい。本論文では、この要件を満たすために、*late binding* (後述) と呼ぶ概念を導入する。この概念を用いて、DoS 攻撃に耐性のある認証プロトコルを以下の通り提案する：

(通報 0) $A \rightarrow B : \{A, N_A\}_{P_B}$

(通報 1) $B \rightarrow A : \{Ack\}_{N_A}$

(通報 2) $A \rightarrow B : Ack$

(通報 3) $B \rightarrow A : \{N_B\}_{P_A}$

(通報 4) $A \rightarrow B : \{B\}_{f(N_A, N_B)}$

ただし、 Ack は受信確認の通報を示す。また、 $f(N_A, N_B)$ は N_A と N_B から得られるハッシュ値を表す。

ここで、レスポンドが、発信 IP アドレスを偽った悪意のあるイニシエータにより最初の通報を大量に送りつける攻撃に晒される状況を考える。この状況において、上で提案したプロトコル (Δ) と Needham-Schroeder-Lowe 公開鍵プロトコル (+) において、それぞれのレスポンドが、複数の同時アクセスを受け、それを処理して最初のレスポンスを返すまでの時間を比較したのが Figure 1 である。また、メモリの占有率を比較したのが Figure 2 である。実験環境は UltraSPARC-III (270 MHz), 128M バイト, SunOS5.6 である。ここで、最初の通報 $\{A, N_A\}_{P_B}$ を受け取って復号化する時間を T_{RSADec} 、セッション鍵 N_B (DES, 56 ビット) を生成する時間を T_{creat} とする。また、通報 B, N_A, N_B を公開鍵 P_A (RSA, 1024 ビット) で暗号化する時間を T_{RSAEnc} 、受信確認 Ack (8 ビット) を共通鍵 N_A (DES, 56 ビット) で暗号化する時間を T_{DESEnc} とする。このとき、提案したプロトコル (Δ)

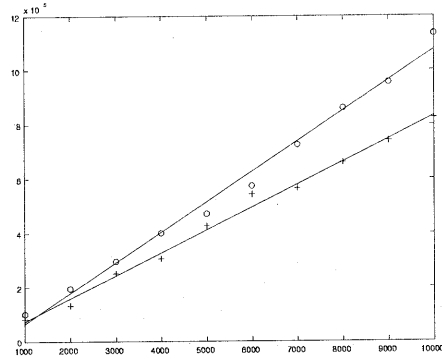


Figure 1: 縦軸は計算時間 (msec), 横軸はアクセス数 (回)

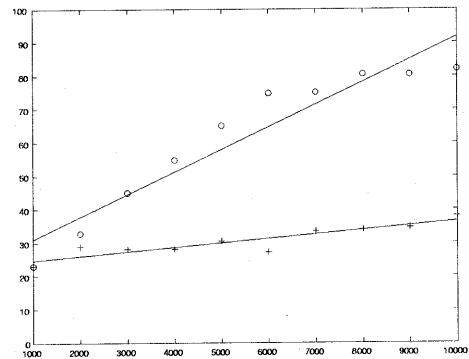


Figure 2: 縦軸はメモリ占有率 (%), 横軸はアクセス数 (回)

, Needham-Schroeder-Lowe 公開鍵プロトコル (+) が通報 0 を受けてから、通報 1 を返すまでの時間をそれぞれ T_{HSM} , T_{NSL} とする。このとき、Figure 1 の縦軸の値はそれぞれ以下の通りになっている：

$$\begin{aligned} T_{HSM} &= T_{RSADec} + T_{DESEnc} \\ T_{NSL} &= T_{RSADec} + T_{creat} + T_{RSAEnc} \end{aligned}$$

明らかに、提案するプロトコルの方がこの攻撃に対して有効であることが分かる。

以上に述べたように、認証プロトコルを設計する際には、認証プロトコルの論理的な安全性だけでは捉えることのできない様々な要因を考慮しなければならない。特に、DoS 攻撃を防ぐためには、通報を適切に分割して送受信することが有効である。本論文では、認証プロトコルの安全性を保証しつつ、どのように通報を分割して送受信すればよいかに関し、認証プロトコルの設計のための指針を示す。この指針に従うことにより、上述のような付加的な性質を満たしつつ安全な認証プロトコルを設計することができる。

本論文は以下の 7 章からなる：2 章では、本論文で用いられる記法について簡潔にまとめ、3 章では、認証プロトコルを設計するための構成要素技術、特に「束縛」の概念を説明し、4 章では、early binding と late binding

の定義を示し、5章では、束縛のタイミングを考慮して通報の分割を行った事例について述べる。6章では、ノンス解析とストランドの概念を用いて4、5章の議論を形式化する。最後に、7章では、まとめと今後の課題について述べる。また、付章において、3章で示した方法を用いて構成される認証プロトコルの例を示す。

2 準備

ここでは、本論文を通して共通に用いる記法を説明する。また、認証プロトコルに対する前提を簡潔にまとめておく。

2.1 記法

記号 A によって、認証を開始する主体、つまりイニシエータ、記号 B によって、イニシエータのアクセスを受ける主体、つまりレスポндаを表す。記号 P_A, P_A^{-1} によって、それぞれ A の公開鍵、対応する A の秘密鍵を表す。また、乱数などのノンスを表すための記号を用意する。例えば、 A が生成したノンスを N_A で表す。

暗号化された通報を次のように表す。鍵 P_A によって暗号化された通報 X を $\{X\}_{P_A}$ によって、 A によって電子署名された通報 X を $\{X\}_{P_A^{-1}}$ によって表す。Ack は受信確認の通報を表す。Msg は任意の通報を意味する。

$f(N_A)$ は N_A のハッシュ値、もしくは N_A そのものを表す。同様に、 $f(N_A, N_B)$ は N_A, N_B を連結した通報から得られるハッシュ値、もしくは連結した通報そのものを表す。ノンスやノンスから得られたハッシュ値を共通鍵とする場合もある。

また、3章以降では、表記の簡略化のため、プロトコルの最初の通報を m_0 、2番目、3番目の通報をそれぞれ m_1, m_2 と記す。更に、必要に応じて、レスポндаからイニシエータへの通報 m_3 を加える。したがって、Needham-Schroeder-Lowe 公開鍵プロトコルは以下のように記述される：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{B, N_A, N_B\}_{P_A} \\ m_2 &= \{N_B\}_{P_B} \end{aligned}$$

さらに、次のような表記法を用いる。例えば $m_0 = A | N_A$ と記述した時には、 m_0 において A または N_A を送信することを意味する。これに対して、 $m_0 = \{A, N_A\}$ と記述した時には、 A と N_A を連結した通報を送信することを意味する。

2.2 前提

TCP/IP 通信などを利用して公開鍵による認証を行う際には、以下のような前提を設けることが一般的である。本論文でもこれらの前提を仮定する。

1. 各主体は正しい公開鍵を取得することができる。
2. 公開鍵を用いて暗号化通報を生成できる。
3. 公開鍵を用いた暗号化通報はそれに対応する秘密鍵を持つ主体以外、復号化することができない。

4. 秘密鍵を用いた電子署名はその秘密鍵を持つ主体以外には行えない。
5. 秘密鍵を用いた電子署名の検証は対応する公開鍵を用いて行える。
6. 認証が行われる以前には存在しない乱数をノンスとして生成できる。
7. ノンスは、それを生成した主体以外、どの主体がいつ生成したかを示せない。
8. 当の主体以外が不正に他の主体に成り済まして通報を送受信している可能性がある。
9. 偽のイニシエータ、偽のレスポндаは、情報を漏洩・偽称する。
10. 本物のイニシエータ、本物のレスポндаは、情報を漏洩・偽称しない。

3 認証プロトコル

3.1 認証プロトコルの要件

イニシエータ A とレスポнда B の間の認証が行われ、両者がセッション鍵 N_B を共有するためには、以下の要件が満たされなければならない：

1. B が A との間のセッションのために生成したセッション鍵 N_B は、 A と B 以外には漏れない。
2. A は B を認証する。特に、 A はセッション鍵 N_B が新規である（詳細は後述）と確信できる。
3. B は A を認証する。特に、「 A は『 N_B が A と B の間のセッション鍵である』と確信している」と B が確信できる（詳細は後述）。

以上の要件を満たすために、以下で示す**認証**、**束縛**、**新規性**の概念を用いて認証プロトコルの手続きを構成する。

3.2 認証

認証とは、通報をやり取りしている相手が、「ある主体と正しく対応している公開鍵」に対する秘密鍵を持っているかを確認することである。 A と B のそれぞれに対して、ノンスによって認証を行う（チャレンジ・レスポンス）か、署名によって認証を行うかの2つの方法が考えられる：

3.2.1 [B ノンス認証]

A は N_A を用いて B を認証する。 N_A は A と B 以外には知られてはならない。したがって、 m_0 は、以下のような形がベースになる：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ &\quad | \{A, \{N_A\}_{P_B}\} \end{aligned}$$

ここで、 m_0 は A からのチャレンジである。

次に、 B によるレスポンスを A が確認するために、 m_1 は以下のような形になる：

$$m_1 = \begin{array}{l} \{f(N_A), \text{Msg}\}_{P_A} \\ | \\ \{\text{Msg}\}_{f(N_A)} \end{array}$$

3.2.2 [A 署名認証]

B は P_A^{-1} を用いて A を認証する。基本的には署名した通報を送信するだけでよく、 m_0 もしくは m_2 に対して以下のような形がベースになる：

$$\begin{array}{l} m_0 = \{\text{Msg}\}_{P_A^{-1}} \\ m_2 = \{\text{Msg}\}_{P_A^{-1}} \end{array}$$

ただし、再送 (replay) による攻撃を防ぐため、 Msg の中にはセッション毎にユニークな通報を入れておく。

3.2.3 [A ノンス認証]

B は N_B を用いて A を認証する。 N_B は A と B 以外には知られてはならない。したがって、 m_1 に対して以下のような形がベースになる：

$$m_1 = \{N_B, \text{Msg}_1\}_{P_A}$$

m_2 (つまり A のレスポンス) は、early binding (後述) ならば、以下のようなものになる：

$$m_2 = \{f(N_B), \text{Msg}_2\}_{P_B}$$

この場合、 A において B と N_B の束縛 (後述) が存在するので、 N_B は B 以外には漏れない。early binding でも late binding でも、 m_2 を以下の形とすることができる：

$$m_2 = \{\text{Msg}_2\}_{f(N_B)}$$

3.2.4 [B 署名認証]

A は P_B^{-1} を用いて B を認証する。基本的には署名した通報を送信するだけでよく、 m_1 に対して以下のような形がベースになる：

$$m_1 = \{\text{Msg}\}_{P_B^{-1}}$$

ただし、再送による攻撃を防ぐため、 Msg の中にはセッション毎にユニークな通報を入れておく。

3.3 束縛

束縛 (binding) [1] とは、2 つ以上の通報の組を電子署名もしくは暗号化して送信することにより、受信した主体の状態の中でそれらの通報が関連付けられることをいう。2 つの通報 Msg_1 、 Msg_2 の組を電子署名したものと暗号化したものは以下の通りある：

$$\begin{array}{l} (1) \ \{\text{Msg}_1, \text{Msg}_2\}_{P_A^{-1}} \\ (2) \ \{\text{Msg}_1, \text{Msg}_2\}_{P_B} \end{array}$$

(1)、(2) 共に、 Msg_1 と Msg_2 を関連付けている。

(1) において Msg_1 を主体の識別子 I とし、 Msg_2 をその公開鍵 P_I とし、 A を Certificate Authority CA と

する。このとき、 $\{I, P_I\}_{P_{CA}^{-1}}$ は、 CA によって発行された『 I の公開鍵が P_I であること』を保証する証明書となる。 CA がその秘密鍵 P_{CA}^{-1} を漏らさず誤った証明書を発行しない限りにおいて、この証明書を受けとった主体の状態の中で I と P_I の対応が保証される。

一方、(2) においては、公開鍵 P_B は誰でも入手できるので、それを用いた暗号化は誰でも行える。したがって、誰がその組み合わせによる通報を生成、もしくは、保証しているかは不明である。しかしながら、その組み合わせを暗号化してから、対応する秘密鍵 P_B^{-1} を用いて復号化するまで、第三者はその組み合わせを変更することは出来ない。つまり、生成者の意図を反映した組み合わせを意図した主体に渡すことができる。

今、主体 A が主体 B にセッション鍵を要求して、 B がセッション鍵 K を返信をする状況を考える：

$$\begin{array}{l} m_0 = A \\ m_1 = \{A, B, K\}_{P_A} \end{array}$$

このとき、 m_1 の通報 $\{A, B, K\}_{P_A}$ により、 A は A 、 B 、 K を束縛することが出来る、すなわち、 K を A と B のセッション鍵と結論づけることができる。しかしながら、この K がだれによって生成されたかが不明であるので、安全なプロトコルとしては、認証も併せて行う必要がある。

3.4 新規性

ノンスと他の通報を束縛することによって、その通報が自分の要求以後に作られたものであるかどうかを判定することができる。つまり、それ以前に存在しないノンス N_A を用いることにより、 A は N_A と束縛された通報 Msg が、ノンスが生成された後に、そのノンスに対応して作られたものであることを確認できる：

$$\begin{array}{l} m_0 = N_A \\ m_1 = \{N_A, \text{Msg}\}_{P_A} \end{array}$$

このとき、 A はこの Msg が新規であるとみなすことができる。

4 early binding と late binding

認証プロトコルの各通報は、 A 、 B 、 N_A 、 N_B の間の束縛を行っていると考えられる。認証プロトコルは、この束縛をいつ行うかにより、early binding と late binding のどちらかに分類することができる。

次節では、 A において B 、 N_A 、 N_B の束縛をいつ行うかにより、認証プロトコルを early binding と late binding の2つに分類する。つまり、 m_1 を A が受信した時点で A において B 、 N_A 、 N_B の束縛が生じることを **early binding** という。このとき、『 A が『 N_B を A と B の間のセッション鍵である』と確信している』と B が確信することができる。

以上に対して、 m_1 を A が受信した時点で A における B 、 N_A 、 N_B の束縛が完全ではなく、その束縛関係を B に問い合わせる通報を m_2 として送信し、それを受信した B が束縛関係を確認することを **late binding** という。

late binding は更に以下の2種に分かれる。詳しくは 5.2 で説明する：

- (1) m_1 において B と N_B の束縛がない。
 (2) m_1 において B と N_B の束縛がある。

5 束縛のタイミングを考慮した設計の事例

本節では、[A ノンス認証] と [B ノンス認証] を用いて、 A と B が相互に認証を行うプロトコルの事例を示す。特に、(1-2) と (2) のプロトコルにおいて、認証のための通報が分割されていることに注意する。

5.1 early binding

Needham-Schroeder-Lowe 公開鍵プロトコルは early binding である：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{B, f(N_A), N_B\}_{P_A} \\ m_2 &= \{f(N_B)\}_{P_B} \end{aligned}$$

m_0 と m_1 により A は B を認証し、 m_1 と m_2 により B は A を認証していることがわかる。特に、 m_1 を受信した時点で A は『 N_B が A と B の間のセッション鍵である』と確信できる。つまり、 A は N_B 、 A 、 B を束縛する。また、 m_0 と m_1 により A は N_B が新規であると確信できる。

5.2 late binding

late binding は、 m_1 において B と N_B との束縛がないかあるかにより、次の二つの場合 (1) と (2) に分かれる。

- (1) m_1 において B と N_B の束縛がない： A は N_B が誰との間のセッション鍵であるかを確定できないので、 m_2 において B の公開鍵を用いて暗号化すると Lowe の man-in-the-middle 攻撃 [4] を防ぐことができない。したがって、 m_2 において N_B の生成者と共有する秘密である N_B そのものを用いて暗号化を行う。
- (2) m_1 において B と N_B の束縛がある： N_B はその生成者である B とのセッション鍵であることが明言されているので、 m_2 において N_B だけでなく B の公開鍵を用いて暗号化してもよい。

以下、(1) と (2) についてより詳しく説明する。

(1) m_1 において B と N_B の束縛がない

さらに、次の 2 つの場合 (1-1) と (1-2) に分かれる。

(1-1) m_1 において N_A と N_B の束縛がある

この場合、 N_A と B の束縛はないと仮定する。 N_A と B の束縛があると N_A と N_B と B の束縛があることになり、early binding になってしまうからである。以下のプロトコルが考えられる：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{f(N_A), N_B\}_{P_A} \\ m_2 &= \{B\}_{f(N_B)} \end{aligned}$$

m_1 を受信した時点で、 A は N_B を新規であると確信できるが、上述の通り、 N_B が誰との間のセッション鍵であるかを確定できない。したがって、 m_2 によって A は N_B を B とのセッション鍵とみなしていることを B に伝えている。

(1-2) m_1 において N_A と N_B の束縛がない

この場合、 N_A と B の束縛は本質的ではないので、 N_A と B の束縛もない場合のみ考える。以下のようなプロトコルが考えられる：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{\{Ack\}_{f(N_A)}, \{N_B\}_{P_A}\} \\ &\quad | \{\{f(N_A)\}_{P_A}, \{N_B\}_{P_A}\} \\ m_2 &= \{B\}_{f(N_A, N_B)} \\ &\quad | \{B, f(N_A)\}_{f(N_B)} \\ &\quad | \{B, f(N_B)\}_{f(N_A)} \\ m_3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

m_1 を受信した時点で、 N_A と N_B の束縛がないので A は N_B が新規であることを確信できず、 N_B が誰との間のセッション鍵であるかも確定できない。したがって、(1-1) と同様に m_2 を送信することによって A は N_A と N_B と B の束縛を B に確認している。 B は、 N_B が N_A に対応する A との間のセッション鍵であると確認できたとき、 m_3 を返信する。

特に、以下は 1 章での提案したプロトコルである：

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{\{Ack\}_{N_A}, \{N_B\}_{P_A}\} \\ m_2 &= \{B\}_{f(N_A, N_B)} \\ m_3 &= \{Ack\}_{N_B} \end{aligned}$$

m_1 の前半部分が (通報 1) に、後半部分が (通報 3) に対応している。 m_2 は (通報 4) に対応する。なお、1 章のプロトコルにおいて m_3 に相当する部分がないのは、 N_B を鍵とする B から A への最初の通報をこれに代えることができるからである。

さらに $\{Ack\}_{f(N_A)}$ を省略して、 m_1 は以下のように簡略化することも可能である。

$$m_1 = \{\{N_B\}_{P_A}\}$$

(2) m_1 において B と N_B の束縛がある

この場合、 m_2 において B の公開鍵を用いて暗号化してもよいので、以下のようなプロトコルが考えられる。

$$\begin{aligned} m_0 &= \{A, N_A\}_{P_B} \\ m_1 &= \{\{Ack\}_{f(N_A)}, \{B, N_B\}_{P_A}\} \\ &\quad | \{\{f(N_A)\}_{P_A}, \{B, N_B\}_{P_A}\} \\ m_2 &= \{Ack\}_{f(N_A, N_B)} \\ &\quad | \{f(N_A)\}_{f(N_B)} \\ &\quad | \{f(N_B)\}_{f(N_A)} \\ &\quad | \{f(N_A), f(N_B)\}_{P_B} \\ m_3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

m_1 において B と N_B の束縛があるため、 m_2 において P_B を暗号化のための鍵として使うことができる。(1-2) と同様に、 A は N_A と N_B の束縛、すなわち N_B の新規性を m_2 と m_3 によって確認できる。

6 ノンス解析に基づく議論

A が m_0 を送信した直後の A の内部状態を $A : s_0(A, B, N_A)$, m_1 を受信し m_2 を送信した直後の内部状態を $A : s_2(A, B, N_A, N_B)$ と書く. m_3 のあるプロトコルの場合は, m_3 を受信した直後の内部状態を $A : s_4(A, B, N_A, N_B)$ と書く.

B が m_0 を受信し m_1 を送信した直後の B の内部状態を $B : s_1(A, B, N_A, N_B)$, m_2 を受信し (m_3 を送信し) た直後の内部状態を $B : s_3(A, B, N_A, N_B)$ と書く.

ノンス解析とは, プロトコルのある時点で生成されたノンスが, それ以後のどのような通報や内部状態に現れるかを解析することである [3, 7].

以下では, 主として B による A の認証について議論する. まず, A と B は正規の (攻撃者ではない) 主体であると仮定する. また, B はプロトコルの仕様に従って通報を送受信し内部状態を遷移させ, 最終的に m_2 を受信して内部状態 $B : s_3(A, B, N_A, N_B)$ に至ったと仮定する.

遡って, B がノンス N_B を作成し通報 m_1 を送信した時点に着目する. このとき, B は $B : s_1(A, B, N_A, N_B)$ という内部状態にいる.

6.1 early binding

early binding である Needham-Schroeder-Lowe 公開鍵プロトコルの場合, 通報 $m_1 = \{B, f(N_A), N_B\}_{P_A}$ を受信できるのは, $A : s_0(A, B, N_A)$ という内部状態にいる A に限られる. A は m_1 を受信して $A : s_2(A, B, N_A, N_B)$ という内部状態に遷移する. ここで, A の内部状態のすべての要素が確定してしまう. これが early binding の意味である.

6.2 late binding

(1-1) m_1 において N_A と N_B の束縛がある

まず, (1-1) のプロトコルに着目する. N_B が作成され $m_1 = \{f(N_A), N_B\}_{P_A}$ が送信された直後の時点において, N_B が現れる通報は m_1 に限られる. これを受信することのできる A の内部状態は $A : s_0(A, _, N_A)$ という形でなければならない. 記号 $_$ は通報や内部状態の確定しない部分を表している.

m_1 を受信した後, A は $m_2 = \{ _ \}_{f(N_B)}$ を送信して, 内部状態 $A : s_2(A, _, N_A, N_B)$ に遷移する. m_2 は N_B を用いて暗号化されているので, N_B そのものが A と B 以外に漏れることはない.

このようにして, N_B が出現する通報と内部状態は以下のように網羅される.

- $m_1 = \{f(N_A), N_B\}_{P_A}$
- $m_2 = \{ _ \}_{f(N_B)}$
- $B : s_1(A, B, N_A, N_B)$
- $A : s_2(A, _, N_A, N_B)$
- $B : s_3(A, B, N_A, N_B)$

さて, ここでは最終的に B は m_2 を受信して内部状態 $B : s_3(A, B, N_A, N_B)$ に至ったと仮定している

が, 通報 $m_2 = \{ _ \}_{f(N_B)}$ を受信できたのは, 内部状態 $B : s_1(A, B, N_A, N_B)$ にいた B だけである. したがって, $m_2 = \{B\}_{f(N_B)}$ でなければならない. すると, $A : s_2(A, _, N_A, N_B)$ は $A : s_2(A, B, N_A, N_B)$ でなければならない.

このようにして, A と B の内部状態が確定する. より厳密には, 上述のノンス解析の結果を用いて, ストランド空間モデル [8, 9, 10, 6] に基づく解析を行うことにより, B に対する A の agreement を証明することができる [7]. すなわち, B のストランド (プロトコルの実行) に対して, A の対応するストランドが存在することを示すことができる. また, ストランド空間モデルに基づいて定式化された authentication test の考え方をすることもできる [2].

このプロトコルの場合, m_1 において N_A と N_B の束縛があるので, A は N_B の新規性を m_1 を受信した時点で確認することができる. すなわち, A による B の認証は, A が m_1 を受信した時点で完了する. より正確には, ノンス解析とストランドの議論により, A に対する B の agreement を証明することができる.

(1-2) m_1 において N_A と N_B の束縛がない

次に, (1-2) のプロトコルに着目する. N_B が作成され m_1 が送信された直後の時点において, N_B が現れる通報は $\{N_B\}_{P_A}$ に限られる. すると, これを受信することのできる A の内部状態は, $A : s_0(A, _, _)$ という形ならば何でもよい.

このように, (1-2) のプロトコルの場合, m_1 において B と N_B の束縛がなく, N_A と N_B の束縛もないので, A の内部状態は全く確定しない.

m_1 を受信した後, A は内部状態 $A : s_2(A, _, _, N_B)$ に遷移して, 通報 $m_2 = \{ _ \}_{f(N_B)}$ を送信する. なお, A の内部状態は N_B 以外は確定しないが, m_2 が N_B を用いて暗号化されているため, N_B は A と B 以外には漏れないことに注意しておく.

A の内部状態は確定しないが, $m_2 = \{ _ \}_{f(N_B)}$ という形の通報を受信できるのは, $B : s_1(A, B, N_A, N_B)$ という内部状態にいる B に限られる. したがって, (1) の場合と同様に, B が m_2 を正しく受信したということを用いて, A の内部状態を確定することができる.

N_A と N_B の束縛がないので, A は m_1 を受信した時点で N_B の新規性を確認できない. したがって, A による B の認証は, A が m_1 を受信した時点では完了しない. このため, A は B に m_2 を送信し, B がこれを正しく受信したことを確認することによって B の認証を行う. 最後の通報 m_3 は, B が m_2 を正しく受信したことを A に知らせる通報である.

なお, m_1 と m_2 において N_A は鍵の一部としてのみ用いられているので, A と B 以外に漏れることはない.

(2) m_1 において B と N_B の束縛がある

(2) のプロトコルの場合は, m_1 において B と N_B の束縛があるので, m_1 を受信した時点で A の内部状態の中の B の部分が確定する. 従って, m_2 を暗号化する際に, B の公開鍵 P_B を用いることができる.

7 さいごに

認証プロトコルの論理的な安全性だけでは捉えることのできない様々な要因を考慮しなければならない。特に、DoS 攻撃を防ぐためには、通報を適切に分割して送受信することが有効である。本論文では、認証プロトコルの安全性を保証しつつ、どのように通報を分割して送受信すればよいに関し、認証プロトコルの設計のための指針を示した。この指針に従うことにより、上述のような付加的な性質を満たしつつ安全な認証プロトコルを設計することができる。

謝辞

東京理科大学大学院理工学研究科情報科学専攻の梅澤健太郎君、鬼頭利之君には実験に関して支援を頂きました。記して感謝します。

References

- [1] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols, *IEEE Transactions on Software Engineering*, Vol.22, No.1, pp.6-15, 1996.
- [2] Joshua D. Guttman and F. Javier Thayer Fábrega. Authentication Tests, *Proceedings, 2000 IEEE Symposium on Security and Privacy*, 2000.
- [3] Masami Hagiya, Yozo Toda and Yoshiaki Fukuba. *Implementation and Verification of Authentication Protocols Using Proof Procedures in HOL, 2nd SSR Enterprise Security Workshop*, Information Media Center, Science University of Tokyo, Nov 1999, <http://nicosia.is.s.u-tokyo.ac.jp/pub/staff/hagiya/ssr99/protveri.ps>
- [4] Gavin Lowe. Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and analysis of Systems. Second International Workshop, TACAS '96*, LNCS 1055, pp.147-166, 1996.
- [5] Roger Needham and Michael Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12), pp.9930-999, 1978.
- [6] Dawn Xiaodong Song. Athena: a New Efficient Automatic Checker for Security Protocol Analysis, *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, 1999, pp.192-202.
- [7] 高橋孝一, 戸田洋三, 萩谷昌己, ノンス解析とストランド空間モデル, 日本ソフトウェア科学会第 17 回大会論文集, 2000.
<http://nicosia.is.s.u-tokyo.ac.jp/pub/staff/hagiya/jssst2000/jssst2000j.ps>

- [8] F. Javier Thayer Fábrega, Jonathan C. Herzog and Joshua D. Guttman. Strand spaces: Why is a Security Protocol Correct? *Proceedings of 1998 IEEE Symposium on Security and Privacy*, 1998, pp.160-171.
- [9] F. Javier Thayer Fábrega, Jonathan C. Herzog and Joshua D. Guttman. Honest Ideas on Strand Spaces. *Proceedings of the 11th IEEE Computer Security Foundations Workshop*, 1998, pp.66-77.
- [10] F. Javier Thayer Fábrega, Jonathan C. Herzog and Joshua D. Guttman. Mixed Strand Spaces. *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, 1999, pp.72-82.

A 認証プロトコルの例

A.1 [B 署名認証][A ノンス認証]

[B 署名認証] と [A ノンス認証] を用いて認証を行う。

A.1.1 early binding

最初の 2 つの m_1 において, B は P_A によって暗号化されていることに注意。

$$\begin{aligned} m_0 &= \{A, N_A\} \\ m_1 &= \{\{B, f(N_A), N_B\}_{P_A}\}_{P_B^{-1}} \\ &\quad | \{f(N_A), \{B, N_B\}_{P_A}\}_{P_B^{-1}} \\ &\quad | \{\{A, f(N_A), N_B\}_{P_B^{-1}}\}_{P_A} \\ &\quad | \{f(N_A), \{A, N_B\}_{P_B^{-1}}\}_{P_A} \\ &\quad | \{\{A, f(N_A)\}_{P_B^{-1}}, N_B\}_{P_A} \\ m_2 &= \{f(N_B)\}_{P_B} \\ &\quad | \{Ack\}_{f(N_B)} \end{aligned}$$

A.1.2 late binding

(1) m_1 において B と N_B の束縛がない

このとき, m_1 において B の署名を用いるため, N_A と N_B の束縛がないパターンのみになる。

$$\begin{aligned} m_0 &= \{A, N_A\} \\ m_1 &= \{\{N_B\}_{P_A}, \{\{A, f(N_A)\}_{P_B^{-1}}\}_{P_A}\} \\ &\quad | \{\{N_B\}_{P_A}, \{\{B, f(N_A)\}_{P_A}\}_{P_B^{-1}}\} \\ m_2 &= \{B, f(N_A)\}_{f(N_B)} \\ &\quad | \{B\}_{f(N_A, N_B)} \\ m_3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

(2) m_1 において B と N_B の束縛がある

$$\begin{aligned} m_0 &= \{A, N_A\} \\ m_1 &= \{\{B, N_B\}_{P_A}, \{\{A, f(N_A)\}_{P_B^{-1}}\}_{P_A}\} \\ &\quad | \{\{B, N_B\}_{P_A}, \{\{B, f(N_A)\}_{P_A}\}_{P_B^{-1}}\} \\ &\quad | \{\{B, N_B\}_{P_A}\}_{P_B^{-1}}, f(N_A)\} \\ &\quad | \{\{A, N_B\}_{P_B^{-1}}\}_{P_A}, f(N_A)\} \\ m_2 &= \{f(N_A), f(N_B)\}_{P_B} \\ &\quad | \{f(N_A)\}_{f(N_B)} \\ &\quad | \{Ack\}_{f(N_A, N_B)} \\ m_3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

A.2 [B ノンス認証][A 署名認証]

[B ノンス認証]と[A 署名認証]を用いて認証を行う。このとき、Aの署名を用いた通報を $m0$ で送信する時(ア)と、 $m2$ で送信する時(イ)の2つに分かれる。

A.2.1 early binding

(ア) $m0$ で送信する時

$$\begin{aligned} m0 &= \{ \{B, N_A\}_{P_A^{-1}} \}_{P_B} \\ &\quad | \{ \{A, N_A\}_{P_B} \}_{P_A^{-1}} \\ m1 &= \{B, f(N_A), N_B\}_{P_A} \\ &\quad | \{B, \{N_B\}_{f(N_A)}\}_{P_A} \\ &\quad | \{B, N_B\}_{f(N_A)} \\ m2 &= \{Ack\} \end{aligned}$$

(イ) $m2$ で送信する時

$$\begin{aligned} m0 &= \{A, N_A\}_{P_B} \\ m1 &= \{B, f(N_A), N_B\}_{P_A} \\ &\quad | \{B, \{N_B\}_{f(N_A)}\}_{P_A} \\ m2 &= \{f(N_A)\}_{P_A^{-1}} \\ &\quad | \{ \{Ack\}_{P_A^{-1}} \}_{f(N_A)} \end{aligned}$$

A.2.2 late binding

(ア-1) $m1$ において B と N_B の束縛がない

(ア-1-1) $m1$ において N_A と N_B の束縛がある

$$\begin{aligned} m0 &= \{ \{B, N_A\}_{P_A^{-1}} \}_{P_B} \\ &\quad | \{ \{A, N_A\}_{P_B} \}_{P_A^{-1}} \\ m1 &= \{f(N_A), N_B\}_{P_A} \\ &\quad | \{ \{N_B\}_{f(N_A)} \}_{P_A} \\ &\quad | \{N_B\}_{f(N_A)} \\ m2 &= \{B\}_{f(N_A)} \\ m3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

(ア-1-2) $m1$ において N_A と N_B の束縛がない

このとき、 $m1$ と $m2$ でチャレンジレスポンスになるので、このパターンはない。以下のように、 N_A と N_B を束縛するために、 N_B をレスポンスしているため：

$$\begin{aligned} m0 &= \{ \{B, N_A\}_{P_A^{-1}} \}_{P_B} \\ &\quad | \{ \{A, N_A\}_{P_B} \}_{P_A^{-1}} \\ m1 &= \{ \{f(N_A)\}_{P_A}, \{N_B\}_{P_A} \} \\ &\quad | \{ \{f(N_A), \{N_B\}_{P_A}\} \} \\ m2 &= \{B, f(N_A)\}_{f(N_B)} \\ &\quad | \{B, f(N_B)\}_{f(N_A)} \\ &\quad | \{B\}_{f(N_A, N_B)} \\ m3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

(ア-2) $m1$ において B と N_B の束縛がある

(ア-1-2)と同様に、このとき、 $m1$ と $m2$ でチャレンジレスポンスになるので、このパターンはない。

(イ-1) $m1$ において B と N_B の束縛がない

(イ-1-1) $m1$ において N_A と N_B の束縛がある

$$\begin{aligned} m0 &= \{A, N_A\}_{P_B} \\ m1 &= \{f(N_A), N_B\}_{P_A} \\ &\quad | \{ \{N_B\}_{f(N_A)} \}_{P_A} \\ m2 &= \{ \{B\}_{P_A^{-1}} \}_{f(N_A)} \\ &\quad | \{B, f(N_A)\}_{P_A^{-1}} \\ m3 &= \{Ack\}_{f(N_B)} \end{aligned}$$

(イ-1-2) $m1$ において N_A と N_B の束縛がない

(ア-1-2)と同様に、このとき、 $m1$ と $m2$ でチャレンジレスポンスになるので、このパターンはない。

(イ-2) $m1$ において B と N_B の束縛がある

(ア-1-2)と同様に、このとき、 $m1$ と $m2$ でチャレンジレスポンスになるので、このパターンはない。

A.3 [B 署名認証][A 署名認証]

[B 署名認証]と[A 署名認証]を用いて認証を行う。このとき、Aの署名を用いた通報を $m0$ で送信する時(ア)と、 $m2$ で送信する時(イ)の2つに分かれる。

A.3.1 early binding

最初の2つの $m1$ において、 B は P_A によって暗号化されていることに注意。

(ア) $m0$ で送信する時

$$\begin{aligned} m0 &= \{N_A\}_{P_A^{-1}} \\ m1 &= \{ \{B, f(N_A), N_B\}_{P_A} \}_{P_B^{-1}} \\ &\quad | \{f(N_A), \{B, N_B\}_{P_A}\}_{P_B^{-1}} \\ &\quad | \{ \{A, f(N_A), N_B\}_{P_B^{-1}} \}_{P_A} \\ &\quad | \{f(N_A), \{A, N_B\}_{P_B^{-1}}\}_{P_A} \\ &\quad | \{ \{A, f(N_A)\}_{P_B^{-1}}, N_B \}_{P_A} \\ m2 &= \{Ack\} \end{aligned}$$

(イ) $m2$ で送信する時

$$\begin{aligned} m0 &= \{A, N_A\} \\ m1 &= \{ \{B, f(N_A), N_B\}_{P_A} \}_{P_B^{-1}} \\ &\quad | \{f(N_A), \{B, N_B\}_{P_A}\}_{P_B^{-1}} \\ &\quad | \{ \{A, f(N_A), N_B\}_{P_B^{-1}} \}_{P_A} \\ &\quad | \{f(N_A), \{A, N_B\}_{P_B^{-1}}\}_{P_A} \\ &\quad | \{ \{A, f(N_A)\}_{P_B^{-1}}, N_B \}_{P_A} \\ m2 &= \{N_A\}_{P_A^{-1}} \end{aligned}$$

A.3.2 late binding

このときも、 $m1$ と $m2$ でチャレンジレスポンスになるので、このパターンはない。