公開鍵を用いた認証プロトコルについて

齋藤孝道[†] 萩谷昌己[††] 溝口文雄[†]

本論文では，公開鍵を用いた認証プロトコルの安全性について考える．特定の相手との内容を第三者に秘匿する通信をする際，確定した相手との秘密の通信のための鍵，つまり，セッション鍵の交換が行われる．それを実現するための公開鍵を用いた認証プロトコルがいくつか提案されているが，その中には悪意ある第三者にセッション鍵が漏洩するなどの問題を持つものもある．そこで，本論文ではセッション鍵の交換を含めた認証プロトコルはどのような要件を満たすべきかを考察し，安全な認証プロトコルに関する枠組みを定める．また，その枠組みを用いて，認証プロトコルの設計例を示し，その要件を満たすかどうかという観点で，既存のいくつかの認証プロトコルの安全性を確かめる．

On Authentication Protocols Using Public-key Cryptography

Takamichi Saito,[†] Masami Hagiya[††] and Fumio Mizoguchi[†]

In this paper, we discuss authentication protocols using public-key cryptography. When initiator and responder communicate securely, they authenticate each other and exchange a session key. There are lots of authentication protocols for this purpose. However, some protocols may have critical flaw. Therefore, in this paper, we provide requirements and framework for secure authentication protocol, and design some protocols derived from the framework. Finally, we check some popular protocols by the requirements.

1. はじめに

近年，インターネットなどのコンピュータネットワークの普及にともない，電子メールや WWW（World Wide Web）など様々な応用が実現されている．また，コンピュータネットワークが社会基盤の 1 つであるという認識も定着しつつある．しかしながら，それによる様々な利点とともにコンピュータネットワークの持つ脅威も社会生活において現実問題となる．つまり，情報セキュリティ問題が我々の日常に現実的なものとなるわけである．

この情報セキュリティは一般に様々な様相を持つが，本論文ではネットワーク・セキュリティ，特に，最も基本的な技術である認証プロトコルの安全性について取り扱う．ここでの認証プロトコルは，通報をやり取りしている相手を確認する手続きと，その認証プロトコルの手続きが終了した時点で，確認した相手と秘密の通信をするための鍵を交換する手続きという 2 つの面を持つ．この 2 つの目的を達成する手続きは，数ステップからなり，共通鍵，もしくは公開鍵暗号用いたものが主である．

認証プロトコルは，上述のとおりの 2 つの目的を実現するという点で，複雑な手続きではない．しかしながら，プロトコルの設計者の意図に反するあらゆる問題を，その設計時に列挙することは困難なことのようである．また，認証プロトコルという手続きによって，どういったことが保証されるのかが示されていないことが多い．

このような背景を鑑みて，本論文では，ネットワーク・セキュリティの要である**認証プロトコルの安全性と，その安全性を満たす認証プロトコルの要件を提案する．**

まず，認証プロトコルにおける前提を列挙し，何を守るべきかを最初に考える．そのうえで，安全性とは状況によって解釈が違うので，本論文における公開鍵を用いた認証プロトコルの安全性を定める．そのうえで，安全な認証プロトコルの満たすべき要件を定める．さらに，認証プロトコルの設計者が，新たな認証プロトコルを設計する際，もしくは，既存の認証プロトコルを改造する際に，どのような通報のやり取りをすればいいのかの枠組みを与える．この枠組みに従うことにより，本論文の意味での安全性を満たす認証プロト

† 東京理科大学
Science University of Tokyo
†† 東京大学
The University of Tokyo

コルを設計できる．例として，その枠組みに基づいた安全な認証プロトコルを示す．また，他の多くの検証は認証プロトコルの欠点を見つけるのに対して，本論文では，ターゲットの認証プロトコルが安全な認証プロトコルの要件を満たしているかどうかを確かめる方法をとる．例として，SSL（Secure Socket Layer）の認証プロトコル，Needham-Schroeder 公開鍵プロトコルなどの基本的な認証プロトコルを確かめる．

2. 準　　備

ここでは，本論文を通して共通に用いる記法と用語を簡潔にまとめておくことにする．

2.1 記　　法

記号 A，B によって認証を行おうとする主体，認証のためのアクセスを受ける主体などを表すことにする．記号 $K_{A,B}$，$P(A)$，$P^{-1}(A)$ によって A と B の共通鍵，A の公開鍵，および，対応する A の秘密鍵をそれぞれ表す．また，乱数などのナンスを表すための記号を用意する．たとえば，A が生成したナンスを N_A で表すなどである．また，ナンスを共通鍵とするときもある．さらに暗号化された通報を次のように表すことにする．鍵 $K_{A,B}$ で暗号化された通報 X を $\{X\}_{K_{A,B}}$ によって，また，A によって電子署名されている通報 X を $\{X\}_{P^{-1}(A)}$ によってそれぞれ表す．また，Ack は受信確認 Ack（文字列）を表す．

2.2 用　　語

以下のとおりに，用語を定める：

やり取り：ある主体がもう一方に通報を送信し，他方がそれを受信することを 1 回のやり取りとする．

手続き：1 回のやり取りに付随する暗号化や復号化，電子署名の検証などを含めたものを 1 回の手続きとする．

（相互通信）プロトコル：手続きを特定の 2 者が 1 回以上行うことを（相互通信）プロトコルとする．

セッション鍵：プロトコルが終了した後，確定した相手と，秘密のやり取りのための共通鍵．

認証プロトコル：相手を認証し，セッション鍵を交換するプロトコルを認証プロトコルとする．

レスポンダ（responder）：他の主体からの要求に応じてセッション鍵を定める通報を送信する主体．

イニシエータ（initiator）：セッション鍵を定める通報の送信を，レスポンダに要求する主体．

3. 認証プロトコルの安全性

3.1 前　　提

一般の TCP/IP 通信などを利用して公開鍵を用いた認証を行ううえで，いくつかの基本的な前提を以下のとおりに定める：

(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．
(3) 公開鍵を用いて暗号化通報を生成できる．
(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ナンスとして生成できる．
(7) ナンスは，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なりす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，偽のレスポンダは，情報を漏洩・偽称する．
(10) 本物のイニシエータ，本物のレスポンダは，情報を漏洩・偽称しない．
(11) 暗号化された通報から，その暗号化のための鍵を特定できない．たとえば，$\{X\}_{N_B}$ から，鍵 N_B を特定できない．
(12) 秘密鍵は対応する主体のみが所有する．たとえば，主体 A の秘密鍵 $P^{-1}(A)$ は A のみが所有する．
(13) 認証を行う以前には，イニシエータ，レスポンダの間で共通鍵を共有していない．

3.2 安全な認証プロトコル

ここでは 3.1 節の前提のもと，公開鍵を用いた認証プロトコルの安全性について考える．

ネットワークセキュリティの確保のためには，一般に，機密性，完全性，可用性を持つ通信であることが必要とされている[3]．この中でも，機密性と完全性が特に重要である．この 2 つを満たす安全な通信を以下のとおりとする：各セッションごとに生成される，つまり，新規であるセッション鍵 N_B を，A と B の間で第三者に漏洩せず共有し，そうであると両者が合意したとき，N_B を用いて**安全な通信**を行うことができる．

イニシエータ A とレスポンダ B の間での安全な通信を行うための認証プロトコル，つまり，**安全な認証プロトコルの要件**を以下に示す：

(1) N_B の生成者である B から A への N_B を含む通報は，A の公開鍵 $P(A)$ で暗号化した通

報である．一方，A から B への N_B を含む通報は，以下の 2 種類のいずれかである：N_B を暗号化の鍵としている通報である．もしくは，A, B, N_B の 3 つ組みを束縛している A により公開鍵 $P(B)$ で暗号化した通報である．

(2) A は B を認証する．特に，A はセッション鍵 N_B が新規であると確信できる．

(3) B は A を認証する．特に，「A が，A，B，N_B の 3 つ組みを束縛する」ための通報を，B が A に送信し，「A が，A，B，N_B の 3 つ組みを束縛した」と B が確認できる．

ここでの，「認証」，「束縛」，「新規性」は，次の章で説明をする．

4. 基本的な議論

ここでは，3.1 節で定めた前提のもと，TCP/IP 通信などを利用して認証を行ううえで基本的な手続きについて考える．

4.1 セッション鍵の生成と共有

ここでは，ナンスを用いて生成する**セッション鍵の共有**について述べる．

まず，主体 A が主体 B にセッション鍵を要求し，B がナンス N_B をセッション鍵として送信し，共有する状況を考える．

以下のとおり，N_B の生成者であるレスポンダ B がイニシエータ A に送信すれば，A，B ともに N_B を共有できる：

(手続き 1) $A \rightarrow B \ : \ A$

(手続き 2) $B \rightarrow A \ : \ N_B$

この共有により，両者は一度はナンス N_B を手にして，それを共通鍵 $K_{A,B}$ と見なしている．以後，文脈に応じて，共通鍵は N_B，もしくは，$K_{A,B}$ で表す．

しかしながら，この手続きは，明らかに，要件 (1) を満たさない．したがって，たとえば，以下のように，N_B を暗号化することにより，レスポンダの手続きは要件 (1) を満たす：

(手続き 1′) $A \rightarrow B \ : \ A$

(手続き 2′) $B \rightarrow A \ : \ \{A, B, N_B\}_{P(A)}$

次に，複数のナンスを用いて，セッション鍵を生成する状況を考える．今，A が B にセッション鍵を要求し，B が返信するとする．ただし，セッション鍵は，A，B がそれぞれ生成したナンス N_A，N_B による $F(N_A, N_B)$（関数 F は，たとえばアペンドなど）とする：

(手続き 1″) $A \rightarrow B \ : \ A, N_A$

(手続き 2″) $B \rightarrow A \ : \ \{A, B, N_B\}_{P(A)}$

このとき，A はセッション鍵 $F(N_A, N_B)$ の N_A を知っているので，B 側からの返信に含まれる N_B に依存することが分かる．これが 3 つ以上のナンスを用いて定める際にもあてはまる．すでに，両者がそれまでのナンスを共有しているとき，最後のナンスを一方が生成し，暗号化して送信することにより，そのナンスによってセッション鍵が定まる．つまり，最後のナンスが文字どおり鍵となる．

4.2 認　　証

ここでは，**認証**について述べる．

3.1 節の前提 (8) にあるように，通信資源からの情報だけでは，発信元アドレスなどが偽造されている可能性もある．また，意図しない主体に通報を解読される可能性もある．したがって，そのような前提のもとでは，やり取りしている主体が公開鍵に対応する秘密鍵を持つかどうかが，「やり取りしている相手が誰であるか」を判定する唯一の手段となりうる．以下で述べる 2 つの認証方法を適宜用いて，認証プロトコルの要件を満たすために認証を行う．

4.2.1 電子署名を用いた場合の認証

電子署名を施してある通報は，明らかに，対応する主体が生成した通報であることが分かる．しかしながら，その通報そのものを盗み取り，送信することは可能であるので，やり取りしている相手がその対応する公開鍵を持つかどうかは不明である．たとえば，今，主体 A が主体 B に，通報 $\{X\}_{P^{-1}(A)}$ を送信する状況を考える：

(手続き 1) $A \rightarrow B, C \ : \ \{X\}_{P^{-1}(A)}$

$\vdots$

(手続き 1′) $C \rightarrow B \ : \ \{X\}_{P^{-1}(A)}$

ここで，$A \rightarrow B, C$ は，「A は B に通報を送信するが，C もその通報を獲得する」状況を示す．攻撃者 C は，この手続き 1′ において，発信元を A と偽って，以前取得した A の正当な通報 $\{X\}_{P^{-1}(A)}$ を自由に再送できる．つまり，通報 $\{X\}_{P^{-1}(A)}$ は誰が生成したかは分かるが，誰が送信したかは不明であるため，正規の主体 A，B が不利益を被る可能性がある．たとえば，ある会社において，A が会社の社長，B がその会社の経理担当者，C を悪徳社員とする．この会社は業務命令のいっさいを電子メールでやり取りするとする．また，X は「C にボーナス 100 万円を与えなさい」という社長命令とする．このとき，$\{X\}_{P^{-1}(A)}$ は社長の電子署名がついているので，社長の正当な手続き 1 の通報によって経理担当者 B はその命令に従うことになる．悪徳社員 C は，この手続き 1 の通報

$\{X\}_{P^{-1}(A)}$ を取得しておいて，適宜，経理担当者 B に送信すれば，お金持ちになれるわけである．このような攻撃を防ぐために，後述の新規性を満たすように，セッションごとにユニークな通報を含む必要がある．

4.2.2 暗号化を用いた場合（チャレンジ・レスポンス）の認証

今，主体 A は主体 B に『自分は主体 A であり，その公開鍵 $P(A)$ を用いてナンス N_A と通報 Y を暗号化した通報 $\{N_A, Y\}_{P(A)}$ を返信してください』と知らせるつもりで手続き 1 を行う．それに応じて，B が返信をする状況を考える：

(手続き 1) $A \to B$: $\{A, N_A\}_{P(B)}$
(手続き 2) $B \to A$: $\{N_A, Y\}_{P(A)}$

ここで通報 Y は B によって生成された任意の通報．

この手続き 1 の後，正しく対応する秘密鍵 $P^{-1}(B)$ を持つ主体 B だけが $\{A, N_A\}_{P(B)}$ を復号化できる．また，N_A はナンスなので，第三者は，この $\{A, N_A\}_{P(B)}$ を生成できず，B は本物のレスポンダであるとき漏洩もない．

しかし，この通報 $\{N_A, Y\}_{P(A)}$ を返信したのが，B とは限らないことに注意する．たとえば，主体 A は攻撃者 C に『自分は主体 A であり，その公開鍵 $P(A)$ を用いて N_A と Y を暗号化した通報 $\{N_A, Y\}_{P(A)}$ を返信してください』と知らせるつもりで手続き $1'$ を行うと以下の攻撃が考えられる：

(手続き $1'$) $A \to C$: $\{A, N_A\}_{P(C)}$
(手続き $2'$) $C \to B$: $\{A, N_A\}_{P(B)}$
(手続き $3'$) $B \to C$: $\{N_A, Y\}_{P(A)}$
(手続き $4'$) $C \to A$: $\{N_A, Y\}_{P(A)}$

このように，手続き $4'$ において，A は通報 $\{N_A, Y\}_{P(A)}$ の生成者を特定できないことが分かる．したがって，攻撃者 C は，あたかも，その通報 $\{N_A, Y\}_{P(A)}$ を自分で作ったことにして，確認などのための返信（たとえば，$\{A, Y\}_{P(C)}$）を A にさせることにより Y を獲得することができる．

これは，$\{N_A, Y\}_{P(A)}$ の生成者 B が，その意図，すわなち，「$\{N_A, Y\}_{P(A)}$ は A と B との通信に対応した通報であること」を明言していないことが問題なのである．本物の B であれば嘘をつかないので，**自分を指し示す識別子** Bを入れれば，その意図を反映した形，つまり，A，B，N_A，Y の組合せを A は受け取ることができる：

(手続き $1''$) $A \to B$: $\{A, N_A\}_{P(B)}$
(手続き $2''$) $B \to A$: $\{B, N_A, Y\}_{P(A)}$

ここで，手続き $2''$ の通報が $P(A)$ を用いて暗号化されていることと，そこに含まれる通報 B により，A と B との間の通信に対応した通報であると，A は解釈できる．

4.3 電子署名と暗号

ここでは，Abadi ら[7]に基づいて通報における暗号と電子署名の関係を示す．今，X を通報とするとき，主体 A から主体 B への通報 X に対しての暗号化と電子署名の順序を入れ替えたものは以下の 2 通りある：

(1) $\{\{X\}_{P(B)}\}_{P^{-1}(A)}$
(2) $\{\{X\}_{P^{-1}(A)}\}_{P(B)}$

(1) では電子署名された通報 $\{X\}_{P(B)}$ を第三者が獲得できることに注意する．したがって，第三者 C が $\{X\}_{P(B)}$ を用いて，$\{\{X\}_{P(B)}\}_{P^{-1}(C)}$ というように，あたかも自分が X を生成したように見せかけることができる．今，主体 A が主体 B に通報 $\{X\}_{P(B)}$ を送信する状況を考える：

(手続き 1) $A \to C(B)$: $\{\{X\}_{P(B)}\}_{P^{-1}(A)}$
(手続き 2) $C \to B$: $\{\{X\}_{P(B)}\}_{P^{-1}(C)}$

このとき，上述の $A \to C(B)$ は，「A は B に送っているつもりが，C によってその通報を奪われ，悪用される」状況を示す．このように，手続き 1 の後，攻撃者 C が通報 $\{X\}_{P(B)}$ を奪い取り，自分の電子署名を施すことにより，あたかも，自分がその通報を生成したかのように B に思わせることができる．つまり，A の意図は反映されずに受信される可能性がある．しかしながら，$\{X\}_{P(B)}$ は生成者の意図どおり，つまり，B に X を伝えることができるので，手続き 1 の通報を $\{\{A, X\}_{P(B)}\}_{P^{-1}(A)}$ とする．この変更により，A の意図を違えずに B に通報を渡すことができる．実際に，受信者 B は識別子 A を解読して，電子署名を検証すればよい．

一方，(2) においては，暗号化の行為自体をどの主体が行ったかを示すことはできないが，$\{X\}_{P^{-1}(A)}$ は A によって生成された通報であることを示すことができる．したがって，(2) を $\{\{B, X\}_{P^{-1}(A)}\}_{P(B)}$ とすることにより，生成者 A の意図を反映することができる．

4.4 束　　縛

ここでは，安全な認証プロトコルの要件で最も重要な概念，束縛について述べる．**束縛**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送信者の意図どおり関連付けることをいう．

4.4.1 電子署名と暗号化による組み

2 つの通報 X，Y の組を電子署名したものと暗号

化したものは以下の2通りある：
(1) $\{X,Y\}_{P^{-1}(A)}$
(2) $\{X,Y\}_{P(B)}$

(1), (2) ともに，X と Y を関連付けている．たとえば，(1) において，X を主体の識別子 I とし，Y をその公開鍵 $P(I)$ とし，$P^{-1}(A)$ の A を，たとえば，Certificate Authority CA とする．このとき，$\{I,P(I)\}_{P^{-1}(CA)}$ 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 $P^{-1}(CA)$ を漏らさず，誤った証明書を発行しない限りにおいて，この証明書を受け取った主体の状態の中で I と P_I の対応が保証される．しかし，この通報 I と P_I は第三者にも獲得できる．

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だれでもできる．したがって，誰がその組合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，その組合せを暗号化してから，対応する秘密鍵 $P^{-1}(B)$ を用いて復号化するまで，第三者はその組合せを変更することはできない．つまり，生成者の意図を反映した組合せで，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 $K_{A,B}$ を返信をする状況を考える．B は自分の意図（$K_{A,B}$ は A と B とのセッション鍵であること）を伝える：

(手続き 1) $A \rightarrow B \quad : \quad A$
(手続き 2) $B \rightarrow A \quad : \quad \{A,B,K_{A,B}\}_{P(A)}$

このとき，A は，手続き 2 の通報 $\{A,B,K_{A,B}\}_{P(A)}$ により，共通鍵 $K_{A,B}$ を A と B のセッション鍵と結論づけることができる．ただし，手続き 2 の通報において用いられている公開鍵 $P(A)$ によって，一方の当事者である A を特定することができるので，手続き 2 の通報を以下のとおり変更しても，A にとっては等しい：

(手続き 2′) $B \rightarrow A \quad : \quad \{B,K_{A,B}\}_{P(A)}$

しかしながら，前述のとおり,「誰が,（B と $K_{A,B}$ の） 組合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」ことより，次の項で示すようなりすまし攻撃に考慮しなければならない．

4.4.2 偽りのセッション鍵

前項と同じように，主体 A が主体 B に，セッション鍵を要求して，B が返信をする状況を考える．このとき，攻撃者 C による以下のような「$K_{A,C}$ を A と B との間のセッション鍵と A に思い込ませる」攻撃に注意しなければならない：

(手続き 1) $A \rightarrow B \quad : \quad A$
(手続き 1.5) $B \rightarrow C(A) \quad : \quad \{B,K_{A,B}\}_{P(A)}$
(手続き 2) $C \rightarrow A \quad : \quad \{B,K_{A,C}\}_{P(A)}$

これは主体 A が主体 B にセッション鍵を要求するための手続きをした後，B から A への返信を攻撃者 C が奪い取り，あたかも，B からの送信のように，C は $\{B,K_{A,C}\}_{P(A)}$ に置き換えている（手続き 2）．プロトコルでこの返信に含まれる共通鍵をセッション鍵とすると定められているとき，A は B とのセッション鍵として $K_{A,C}$ を採用することになる．ここで，主体 A は，プロトコルに従うので，$K_{A,C}$ を生成したのは攻撃者 C か，主体 B かは判別できないことに注意する．また，攻撃者 C にとって，主体 B になりすますことが目的であるから，以下のような手続き 2^* の行為は意味がないことは明らかである：

(手続き 2^*) $C \rightarrow A \quad : \quad \{C,K_{A,C}\}_{P(A)}$

さらに，手続き 2 の時点で，共通鍵 $K_{A,B}$ を攻撃者 C が手に入れることは，ナンスの前提によりありえない．

4.4.3 束縛のパターン

では，上述の攻撃に対する防御法を考える．

ここで，手続き 2 の時点で，主体 A は一方的に与えれた共通鍵 $K_{A,B}$，もしくは $K_{A,C}$ を，盲目的にセッション鍵と見なさなければならない状況にある．したがって，A は，4.3 節の送信者の意図を反映する通報を受け取るか，または，4.2.2 項で最後に示したプロトコルを利用して，その生成者が誰であるかを特定すればよい．

送信者の意図を反映する形で，共通鍵 $K_{A,B}$ を受け取るには，4.3 節で示したとおり 2 種類ある．ここで，それぞれを束縛のパターン 1，2 とする：

署名して暗号化する：

[束縛のパターン 1]
(手続き 1′) $A \rightarrow B \quad : \quad A$
(手続き 2′) $B \rightarrow A \quad : \quad \{\{A,K_{A,B}\}_{P^{-1}(B)}\}_{P(A)}$

暗号化して署名する：

[束縛のパターン 2]
(手続き 1″) $A \rightarrow B \quad : \quad A$
(手続き 2″) $B \rightarrow A \quad : \quad \{\{B,K_{A,B}\}_{P(A)}\}_{P^{-1}(B)}$

以上の 2 つのプロトコルにおいて，B の意図を反映した形，つまり，$B,K_{A,B}$（と A）の組合せを，A は受け取ることができる，つまり,「A はこの組合せを束縛する」.

また，4.2.2 項の最後で示したプロトコルの Y をセッション鍵 $K_{A,B}$ とする．同様に,「B，N_A，$K_{A,B}$

(と A) の組合せを A は束縛する」. これを束縛のパターン 3 とする:

[束縛のパターン 3]

(手続き 1) $A \to B$: $\{A, N_A\}_{P(B)}$

(手続き 2) $B \to A$: $\{B, N_A, K_{A,B}\}_{P(A)}$

4.5 新規性

ここでは, 通報の新規性について考える. ここでの方法を用いて, 認証プロトコルの要件の新規性を確かめることができる.

ナンスと他の通報を束縛することによって, 自分の要求以後にその通報が相手により作られたものである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はずのナンス N_A を用いることにより, 生成される前と生成された後の因果関係により, 主体 A はそのナンスを含む通報 X (と N_A の組) はナンスが生成された後に作られたものであることが分かる:

(手続き 1) $A \to B$: N_A

(手続き 2) $B \to A$: $\{N_A, X\}_{P(A)}$

このとき, A は「この通報 X は新規である」と解釈できる.

4.6 安全な通信と安全な認証プロトコル

ここまでで,「認証」,「束縛」,「新規性」の説明をした. これらを用いて, 3.1 節の前提のもとに, 3.2 節で提案した要件を満たす認証プロトコルによって, N_B による「安全な通信」を行うことができることを示す. ここで, A はイニシエータ, B はレスポンダとする.

要件 (1) に関して:

B は N_B を A の公開鍵 $P(A)$ で暗号化して送信しているとき, 前提の (2) にあるとおり, 公開鍵の性質から, A 以外の主体が解読して取得することはできない. 一方, A から B への N_B を含む通報の送信においては, 以下である: N_B を暗号化の鍵としている通報, たとえば, $\{X\}_{N_B}$ であるとき, 前提 (11) より, 第三者は N_B を入手することができない. また, A において, A, B, N_B の 3 つ組みを束縛できたとき, 4.4.3 項より N_B は B によって生成されているといえる. したがって, 公開鍵 $P(B)$ で暗号化した通報, たとえば, $\{N_B\}_{P(B)}$ は, 生成者 B のみが解読する. 以上より, プロトコルにおいて, N_B を含む通報を網羅しているので, 要件 (1) を満たすプロトコルは, N_B を第三者に漏洩しない.

逆に, N_B を第三者に漏洩していないとき, B からの送信において, 前提 (13) より, 漏洩せずに渡す方法は公開鍵を用いる以外にありえないので, B は N_B を $P(A)$ で暗号化して送信している. また, A からの送信において, 前提 (13) より, 漏洩せずに渡す方法は公開鍵を用いるか, N_B を鍵としている以外にありえないので, A は, N_B の生成者 B の鍵 $P(B)$ を用いて N_B を暗号化しているか, N_B を鍵としている.

要件 (2) に関して:

4.2 節の方法を用いて, A が B を認証したとき, やり取りをしている相手が対応する秘密鍵を持つ. つまり, 秘密鍵 $P^{-1}(B)$ を持っている. この方法を用いて確認できたとき, 前提 (12) の秘密鍵の性質により, A がやり取りをしている相手は B である.

逆に, 安全な通信において, A がやり取りをしている相手は B であるとき, 前提 (12) の秘密鍵の性質により, 認証された主体である.

また, N_B が新規であることは, N_B の生成者 B にとっては明らかである. したがって, A が N_B を新規であることを確認する. A は, 4.5 節の方法を用いて, N_B を新規であるかどうか判定できる. したがって, この方法を用いて確認できたとき, N_B は当該セッションにおいて生成されたものである.

要件 (3) に関して:

4.2 節の方法を用いて B が A を認証したとき, やり取りをしている相手が対応する秘密鍵を持つ. つまり, 秘密鍵 $P^{-1}(A)$ を持っている. この方法を用いて確認できたとき, 前提 (12) の秘密鍵の性質により, B がやり取りをしている相手は A である.

逆に, 安全な通信において, B がやり取りをしている相手は A であるとき, 前提 (12) の秘密鍵の性質により, 認証された主体である.

また, プロトコルにおいて,「A が, A, B, N_B の 3 つ組みを束縛する」ための通報を, A が受信して,「A が, A, B, N_B の 3 つ組みを束縛した」と B が確認できれば, 4.4.3 項より, A は「N_B が A と B の間のセッション鍵である」と解釈する. よって, N_B の生成者 B は「N_B が A と B の間のセッション鍵である」としているので, 両者が「N_B を A と B との間でのセッション鍵である」という合意をする.

逆に, この合意があるとき, 組合せを違えずに, A, B, N_B の 3 つ組みを共有している.

5. 安全な認証プロトコルの設計

ここでは, 安全な認証プロトコルの要件のために, イニシエータとレスポンダがどのように「認証」,「束縛」,「新規性」を用いるべきかの枠組みを述べる.

5.1 束縛のパターンを基にしたプロトコル設計

以後では, 4.4.3 項における束縛のパターンを基に議論を進める.

3つの束縛のパターンは，やり取りしている通報より，要件(1)を満たしていることは明らかである．

よって，認証プロトコルの要件の(2)におけるセッション鍵の $K_{A,B}$ の新規性について，束縛のパターンを再考する．束縛のパターン3は，B からの返信に N_A を含むので，A は返信を受け取った時点で，$K_{A,B}$ が新規であると解釈できる．一方，束縛のパターン1，2は，N_A を含まないので，たとえば，以下のように変更する：

[束縛のパターン 1']

(手続き 1′)　$A \rightarrow B : A, N_A$

(手続き 2′)　$B \rightarrow A : \{\{A, N_A, K_{A,B}\}_{P^{-1}(B)}\}_{P(A)}$

[束縛のパターン 2']

(手続き 1″)　$A \rightarrow B : A, N_A$

(手続き 2″)　$B \rightarrow A : \{\{B, N_A, K_{A,B}\}_{P(A)}\}_{P^{-1}(B)}$

このプロトコルにおいて，B からの返信を受け取った時点で，A は B，N_A，$K_{A,B}$（と A）の組合せを束縛し，$K_{A,B}$ を新規であると解釈できる．

また，このパターン1'，2'は署名を用いて，パターン3は暗号化を用いて，A による B の認証している．したがって，この1'，2'，3ともに安全な認証プロトコルの要件(2)を満たしている．

さらに，安全な認証プロトコルの要件(3)の B による A の認証に関しては，たとえば，$K_{A,B}$ は B が生成したナンス N_B であるので，4.2.2 項で示した暗号化を用いた認証を用いて行うことができる．すなわち，これらのパターンの3番目に以下の手続き3を加える：

(手続き 3)　$A \rightarrow B : \{K_{A,B}\}_{P(B)}$

また，手続き3を以下としても暗号化を用いた認証を行うことはできる：

(手続き 3′)　$A \rightarrow B : \{Ack\}_{K_{A,B}}$

以上の「B による A の認証」と，束縛のパターンによる「A における B，N_A，$K_{A,B}$（と A）の組合せの束縛」により安全な認証プロトコルの要件(3)が満たされる．

5.2 NS プロトコル再考

ここで，欠陥のある Needham-Schroeder 公開鍵プロトコルを再考してみる．今，このプロトコルの手続き2までをもって束縛のパターン4とする：

[束縛のパターン 4]

(手続き 1)　$A \rightarrow B$　：　$\{A, N_A\}_{P(B)}$

(手続き 2)　$B \rightarrow A$　：　$\{N_A, K_{A,B}\}_{P(A)}$

このプロトコルにおいて，B からの返信を受け取った時点で，A は B，N_A，$K_{A,B}$（と A）の組合せを束縛しておらず，N_A，$K_{A,B}$（と A）の組合せのみを束縛している．これは，4.2.2 項の議論にあるとおり，B が十分な情報を渡していないためである．したがって，B は A がこの B，N_A，$K_{A,B}$ を束縛しているかどうかを確認する必要がある．すなわち，N_A，$K_{A,B}$（と A）の組合せを A は束縛しているので，B，$K_{A,B}$（と A, N_A）の組合せを A が束縛している旨の通報を A は B に返信する：

(手続き 3)　$A \rightarrow B : \{B\}_{K_{A,B}}$

この A からの返信において，$K_{A,B}$ を鍵として使っているので漏洩せず，$K_{A,B}$ の生成者 B に A の意図を違えずに渡すことができる．よって，B は「A が B，N_A，$K_{A,B}$（と A）を束縛していること」を確認できる．

5.3 プロトコルの例

ここで示す認証プロトコルは，これまでの議論から束縛のパターンをもとに設計されたものである．ここでのセッション鍵は N_B とする．

束縛のパターン 1' をベースにしたプロトコル：

(手続き 1)　$A \rightarrow B$　：　A, N_A

(手続き 2)　$B \rightarrow A$　：　$\{\{A, N_A, N_B\}_{P^{-1}(B)}\}_{P(A)}$

もしくは　$\{N_A, \{A, N_B\}_{P^{-1}(B)}\}_{P(A)}$

(手続き 3)　$A \rightarrow B$　：　$\{N_B\}_{P(B)}$

もしくは　$\{Ack\}_{N_B}$

束縛のパターン 2' をベースにしたプロトコル：

(手続き 1)　$A \rightarrow B$　：　A, N_A

(手続き 2)　$B \rightarrow A$　：　$\{\{B, N_A, N_B\}_{P(A)}\}_{P^{-1}(B)}$

もしくは　$\{N_A, \{B, N_B\}_{P(A)}\}_{P^{-1}(B)}$

(手続き 3)　$A \rightarrow B$　：　$\{N_B\}_{P(B)}$

もしくは　$\{Ack\}_{N_B}$

束縛のパターン 3 をベースにしたプロトコル：

(手続き 1)　$A \rightarrow B$　：　$\{A, N_A\}_{P(B)}$

(手続き 2)　$B \rightarrow A$　：　$\{B, N_A, N_B\}_{P(A)}$

(手続き 3)　$A \rightarrow B$　：　$\{N_B\}_{P(B)}$

もしくは　$\{Ack\}_{N_B}$

束縛のパターン 4 をベースにしたプロトコル：

(手続き 1)　$A \rightarrow B$　：　$\{A, N_A\}_{P(B)}$

(手続き 2)　$B \rightarrow A$　：　$\{N_A, N_B\}_{P(A)}$

(手続き 3)　$A \rightarrow B$　：　$\{B\}_{N_B}$

6. 既存の認証プロトコルの安全性

ここでは，公開鍵を用いた既存の認証プロトコルについて，束縛の概念などを用いて安全性を確かめる．6.2，6.3 節は議論が重複するが，ここでは，与えられたプロトコルの安全性を確かめるという視点で扱う．

6.1 SSL プロトコル

ここでは，SSL ハンドシェイクプロトコル[8]について述べる．このプロトコルで広く利用されているのはサーバ認証，つまり，クライアントはサーバを認証するが，サーバはクライアントを認証しないというモードである．ここでは，このモードについて，認証プロトコルの要件を確かめる．

このプロトコルにおいては，匿名であるクライアント C とクライアントから認証されたサーバ S の間で暗号化通信のための秘密鍵生成のための種ナンス N'_C が生成される．本論文では，議論を簡潔にするため，このプロトコルは以下の 4 つの手続きからなるとする：

(手続き 1) $C \to S$: N_C, T_C
(手続き 2) $S \to C$: N_S, T_S
(手続き 3) $S \to C$: $\{S, P(S)\}_{P^{-1}(CA)}$
(手続き 4) $C \to S$: $\{N'_C\}_{P(S)}$

ここで，関数 F を適当に定めたとき，セッション鍵 $K_{C,S}$ は

$$K_{C,S} = F(N_C, N_S, N'_C),$$

である．また，T_C，T_S はタイムスタンプを表すが，ここでは無視する．CA は，TTP（Trusted Third Party）としての Certificate Authority である．このプロトコルにおいて，クライアント C がレスポンダ，サーバ S がイニシエータとなっていることに注意する．

一見，正しい公開鍵を取得できるという本論文の前提のもとでは，手続き 3 は無意味に見えるが，実はそうではない．本論文の前提のもとでは手続き 3 は以下と等しい：

(手続き 3′) $S \to C$: S

このようにして見ると，レスポンダ C は，手続き 4 で S の公開鍵で暗号化しているので，安全な認証プロトコルの要件 (1) を満たすことが分かる．

一方，イニシエータ側 S の視点で見ると，セッション鍵 $K_{C,S}$ は，S が生成したナンス N_S に依存することより，新規性を持つが，「$K_{C,S}$ と S」が「C」と束縛されていないことが分かる．したがって，このプロトコルにおいて，クライアント側からの通報を秘密に S に届けることができるが，だれから送られているかは S は特定できない．であるから，蛇足であるが，このモードのとき，S は，クレジットカードの番号などの情報を送ってもらうことにより，秘密鍵を用いた方法以外で C を特定するのである．

6.2 Needham-Schroeder 公開鍵プロトコル

このプロトコルは次の 3 つの手続きに従って実行される：

(手続き 1) $A \to B$: $\{A, N_A\}_{P(B)}$
(手続き 2) $B \to A$: $\{N_A, N_B\}_{P(A)}$
(手続き 3) $A \to B$: $\{N_B\}_{P(B)}$

このプロトコルは，これまでの議論で分かるように，安全でないプロトコルである．実際，手続き 2 のレスポンダ B の通報は不十分で，A が B，N_A，$K_{A,B}$（と A）を束縛できない．したがって，手続き 3 で A は相手を確証できないまま，セッション鍵 N_B を送信している．

6.3 Needham-Schroeder-Lowe 公開鍵プロトコル

このプロトコルは，Lowe によって提案された Needham-Schroeder 公開鍵プロトコルの訂正版である[2]．プロトコルの仕様として 2 番目の手続きにおける通報に自分の識別子を入れるように変更されている．このプロトコルは次の 3 つの手続きに従って実行される：

(手続き 1) $A \to B$: $\{A, N_A\}_{P(B)}$
(手続き 2) $B \to A$: $\{B, N_A, N_B\}_{P(A)}$
(手続き 3) $A \to B$: $\{N_B\}_{P(B)}$

このプロトコルは，5.3 節の束縛のパターン 3 で示したプロトコルである．したがって，セッション鍵 N_B を共有することができる安全な認証プロトコルである．

7. おわりに

本論文では，ネットワーク・セキュリティの要である認証プロトコルの安全性と，その安全性を満たす認証プロトコルの要件を提案した．

認証プロトコルの設計者が，新たな認証プロトコルを設計する際，もしくは，既存の認証プロトコルを改造する際に，この枠組みに従うことにより，本論文の意味での安全性を満たす認証プロトコルを設計することができる．

また，要件を満たすかどうかで，既存の認証プロトコルの安全性を検証することができる．既存の SSL や Needham-Schroeder 公開鍵プロトコルの安全性をこの枠組みを用いて確かめた．

今後の課題として，本論文での議論を形式的な記述をし，自動検証器の実現という方向性がある[5),6)]．

謝辞 有益なご意見をいただきました梅澤健太郎氏ならびに査読者に感謝いたします．

この研究は，平成 13 年度文部科学省特定領域研究（B）「社会基盤としてのセキュア・コンピューティングの実現方式の研究」（課題番号：12133209，12133101）の成果の一部である．

参 考 文 献

1) Schneier, B.: *Applied Cryptography*, 2nd ed., John Wiley & Sons, Inc. (1996).
2) Lowe, G.: Breaking and fixing the Needham-Schroeder public-key protocol using CSP and FDR, *Tools and Algorithms for the Construction and Analysis of Systems, Second International Workshop, TACAS'96*, Margaria, T. and Steffen, B. (Eds.), LNCS, Vol.1055, pp.147–166 (1996).
3) 佐々木良一：インターネットセキュリティ入門，岩波新書 (1999).
4) Stallings, W.: *Cryptography and Network Security — Principls and Practice*, 2nd ed., Prentice Hall (1999).
5) Hagiya, M., Toda, Y. and Fukuba. Y.: Implementation and Verification of Authentication Protocols Using Proof Procedures in HOL, *2nd SSR Enterprise Secuirty Workshop*, Information Media Center, Science University of Tokyo (Nov. 1999).
`http://nicosia.is.s.u-tokyo.ac.jp/pub/staff/hagiya/ssr99/protveri.ps`
6) 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`
7) Abadi, M. and Needham, R.: Prudent engineering practice for cryptographic protocols, *IEEE Trans. Softw. Eng.*, Vol.22, No.1, pp.6–15 (1996).
8) Freier, A., Kocher, P. and Kaltorn, P.: SSL V3.0 Specification, Technical Report, HTM, IETF Task Force (Mar. 1996).
`http://home.netscape.com/eng/ssl3/s-SPEC`

(平成 12 年 12 月 1 日受付)
(平成 13 年 6 月 19 日採録)

齋藤 孝道

1996 年東京理科大学理工学部情報科学科卒業．1998 年同大学院理工学研究科情報科学専攻修士課程修了．同年，同学科助手．システムセキュリティに興味を持つ．

萩谷 昌己（正会員）

1982 年東京大学大学院理学系研究科情報科学専攻修士課程修了．京都大学数理解析研究所を経て，現在，東京大学大学院情報理工学系研究科教授．基本的に，演繹的推論を計算上に実装することに興味を持っている．また，最近では，生命情報関連の研究（特に，分子計算）も行っている．

溝口 文雄（正会員）

1941 年生．1968 年東京理科大学大学院修士課程修了．同年同大学理工学部経営工学科助手．1987 年教授となり，現在大学院研究科委員および情報メディアセンター長．工学博士．1994 年 Stanford 大学の Center for the Study of Language and Information（CSLI）の上級研究員となる．人工知能，認知科学およびソフトウェア工学に興味を持ち，Java を用いた情報家電や自律型ロボットの研究を中心に進めている．現在 Artificial Intelligence Journal の論文編集委員．日本ソフトウェア科学会，日本認知科学会，米国人工知能学会各会員．