論文 / 著書情報
Article / Book Information

題目(和文)	並行セッションを利用した攻撃を考慮するセキュリティプロトコルの安全性検証に関する研究
Title(English)	
著者(和文)	根岸和義
Author(English)	
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第4583号, 授与年月日:2000年9月30日, 学位の種別:課程博士, 審査員:米崎 直樹
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第4583号, Conferred date:2000/9/30, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

博士論文

並行セッションを利用した攻撃を考慮する
セキュリティプロトコルの安全性検証に関する研究

指導教官 米崎 直樹 教授

東京工業大学大学院

情報理工学研究科計算工学専攻

根岸 和義

2000 年 9 月

目 次

第1章　まえがき

インターネットを利用した電子商取引の普及にともない、金銭に関するデータや取引の重要情報のデータなどがインターネットを通じて伝達されるようになって来た。このため、通信のセキュリティを守ることによる機密の漏洩防止や不正防止の必要性が高まっている。通信のセキュリティを守るためには、個別のメッセージのセキュリティを守るための暗号化の技術は必須であるが、それだけでは十分ではない。メッセージのすり替え等の攻撃によりプロトコルの実行中にセキュリティが破られることがある。このような攻撃に耐性のあるプロトコル (セキュリティプロトコル) の技術が必要とされる。本論文では、悪意をもつ攻撃者（イントルーダ）の存在を前提として、セキュリティプロトコルの検証に関して検討する。

セキュリティプロトコルに関する研究は古くから行われてきた [4, 11]。従来のセキュリティプロトコルの検証法の研究としては、代数的モデルに対する状態生成と解析 (モデルチェック) によるもの [8]、論理によるプロトコル解析を行う BAN 論理 [2] 、これを拡張して型によるメッセージのすり替えの可能性検出を行う SG 論理 [5] 等の論理によるものがあった。また、BAN 論理による自動解析ツールの研究も行われている。[14] さらに、BAN 論理をセキュリティではなく一般のネットワークプロトコルの完全性の検証に用いる研究も行われている [10]。BAN 論理に対しては、並行するセッションを利用した攻撃に対処出来ないとの批判があり [1, 12, 13]、メッセージの型チェックによるすり替えの可能性検出を提案している SG 論理もこの問題を解決出来ていない。これら、論理による考察ではプロトコルの定義をもとに問題となりそうなメッセージを選択し、これらメッセージに関して個別に検証を行なうのが通例であった。

我々の想定するプロトコルの実行環境のモデルを図 1.1 に示す。プロトコルは確定した参加者を伴ったプロセスにより実行される。プロトコルの１回の実行をセッションと呼ぶ。各プロセスは一度に一つのセッションを実行する。同一参加者の異なるプロセスはセッションを並行して実行することができる。

本論文では、セキュリティプロトコルの目的を、電子商取引への適用を前提として、下記の 2 項目とし、イントルーダの攻撃により 、これらの目的が達成できない可能性を検証する。

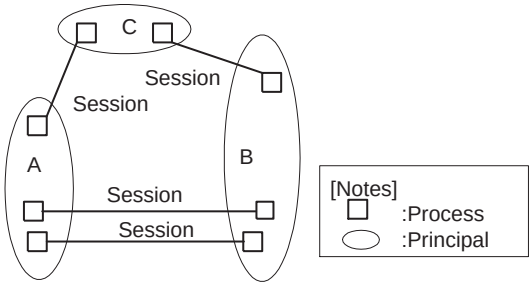

図 1.1: 参加者、プロセスとセッション

プロトコルの一貫した実行 プロトコルを実行し、正常終了したプロセスがセッションの最初から最後まで同一の定められた相手のプロセスからのメッセージのみを受信し、他のセッションからメッセージとすり替えられたりしない。

正常終了の一致 あるセッションでプロトコルを実行している複数のプロセスのうち一つがセッションを正常に終了したと判断するならば、残りのプロセスも同一の判断を下す。

なお、セキュリティプロトコルの目的のひとつである秘密の保持に関しては、本論文の検討範囲外とし、別の手段により保持されているものとする。秘密情報、特に参加者の秘密キー、共用キーがイントルーダに知られることはないと仮定する。

本論文の第 2 章では、プロトコル検証に関する主な研究と本研究の位置づけを述べる。第 3 章では、本研究に共通のプロトコルのモデルとその初期条件を示し、第 4 章では送信者の状態推定によるメッセージの認証の検証につき述べる。第 5 章ではメッセージすり替えの可能性のないことの検証を述べ、第 6 章では別セッションの同一部分間のすり替えの可能性のないことの検証につき述べ、第 7 章では適用例を示す。

第2章　プロトコル検証に関する主な研究と本論文の位置付け

本章では、従来の論理によるプロトコルセキュリティの研究の代表的なものとして、BAN 論理および SG 論理に関して述べる。また、これらの研究に対する本論文の研究の位置付けに関しても述べる。

2.1 BAN 論理の概要

BAN 論理によるセキュリティプロトコル検証の概要を、主に文献 [2] に基づいて説明する。

2.1.1 用語

BAN 論理では、参加者、暗号化キー、ステートメント（メッセージ）をオブジェクトとする。その他、本節で使用する用語は以下の通りである。

特定の参加者 A, B, S

特定の共用キー K_{ab}, K_{as}, K_{bs}

特定の公開キー K_a, K_b, K_s

対応する秘密キー $K_a^{-1}, K_b^{-1}, K_s^{-1}$

特定のステートメント N_a, N_b, N_s

参加者に対する記号 P, Q, R

ステートメントに対する記号 X, Y

暗号化キーに対する記号 K

2.1.2 論理式

BAN 論理で用いる論理式は以下である。

$P \mid\equiv X$ P は X を信じる。P は X を真として振る舞う。

$P \lhd X$ P は X を見る。誰かが P に X を送った。

$P \mid\sim X$ P はかって X と言った。

$P \mid\Rightarrow X$ P は X の管轄権がある。

$\sharp(X)$ 式 X は新鮮である。X は以前のプロトコル実行では送られたことはない。

$P \overset{K}{\leftrightarrow} Q$ P と Q は共用キー K を通信に用いる。

$\overset{K}{\mapsto} P$ P は K を公開キーとする。

$P \overset{X}{\rightleftharpoons} Q$ 式 X は P と Q のみが知っている秘密である。

$\{X\}_K$ X はキー K により暗号化されていることを表す。

$\langle X \rangle_Y$ X は Y と結合されている。Y は X の発言者を示す。

2.1.3 推論規則

BAN 論理で用いる推論規則を以下に示す。考え方として重要なポイントは現在実行中のセッションとのかかわりにより、過去と現在を分けること。過去とは、現在のセッションの開始する以前である。

- 「かって何々といった」を推論する規則

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, \quad P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X} \tag{2.1}$$

$$\frac{P \mid\equiv \overset{K}{\mapsto} Q, \quad P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X} \tag{2.2}$$

$$\frac{P \mid\equiv Q \overset{Y}{\rightleftharpoons} P, \quad P \lhd \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X} \tag{2.3}$$

- fresh(新鮮) なものはすり替えられていないとする規則

$$\frac{P \mid\equiv \sharp(X), \quad P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X} \tag{2.4}$$

- 管理権規則

$$\frac{P \mid\equiv Q \mid\Rightarrow X, \quad P \mid\equiv Q \mid\equiv X}{P \mid\equiv X} \tag{2.5}$$

- 「信じる」の必要とする性質

$$\frac{P \mid\equiv X, \quad P \mid\equiv Y}{P \mid\equiv (X, Y)} \tag{2.6}$$

$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv X} \tag{2.7}$$

$$\frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X} \tag{2.8}$$

- 「かって何々といった」の結合は分解出来る。

$$\frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X} \tag{2.9}$$

ここで、暗号化しているメッセージは分解できないことに注意。

- 式を受信したなら、その要素も受信した。

$$\frac{P \lhd (X, Y)}{P \lhd X} \tag{2.10}$$

$$\frac{P \lhd \langle X \rangle_Y}{P \lhd X} \tag{2.11}$$

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, \quad P \lhd \{X\}_K}{P \lhd X} \tag{2.12}$$

$$\frac{P \mid\equiv \overset{K}{\mapsto} P, \quad P \lhd \{X\}_K}{P \lhd X} \tag{2.13}$$

$$\frac{P \mid\equiv \overset{K}{\mapsto} Q, \quad P \lhd \{X\}_{K^{-1}}}{P \lhd X} \tag{2.14}$$

- 式の一部が新鮮なら式全体も新鮮

$$\frac{P \mid\equiv \sharp(X)}{P \mid\equiv \sharp(X, Y)} \tag{2.15}$$

  ここで、この規則は、並行セッションがある場合には正しくないことに注意。

- 同一のキーを参加者の間で双方向に使用する。

$$\frac{P \mid\equiv R \overset{K}{\leftrightarrow} R'}{P \mid\equiv R' \overset{K}{\leftrightarrow} R} \tag{2.16}$$

$$\frac{P \mid\equiv Q \mid\equiv R \overset{K}{\leftrightarrow} R'}{P \mid\equiv Q \mid\equiv R' \overset{K}{\leftrightarrow} R} \tag{2.17}$$

- 同様に秘密も参加者の間で双方向に使用する。

$$\frac{P \mid\equiv R \overset{X}{\rightleftharpoons} R'}{P \mid\equiv R' \overset{X}{\rightleftharpoons} R} \tag{2.18}$$

$$\frac{P \mid\equiv Q \mid\equiv R \overset{X}{\rightleftharpoons} R'}{P \mid\equiv Q \mid\equiv R' \overset{X}{\rightleftharpoons} R} \tag{2.19}$$

BAN 論理 [2] には、いくつかの例がとりあげられているが、ここでは、そのうち 2 個を取り上げる。

2.1.4 The Otway-Rees Protocol への適用例

2 人の参加者と認証サーバによる認証プロトコルへの適用例である。

用語

この例に固有の用語を説明する。

A, B 参加者

K_{as}, K_{bs} サーバとの共用キー

S 認証サーバ

N_a, N_b A, B の生成する nonce

K_{ab} S の生成する A、B のセッションキー

プロトコル

以下は、元のプロトコルである。

$$
\begin{array}{lll}
\text{Message1} & A \to B: & M, A, B, \{N_a, M, A, B\}_{K_{as}} \\
\text{Message2} & B \to S: & M, A, B, \{N_a, M, A, B\}_{K_{as}}, \\
 & & \{N_b, M, A, B\}_{K_{bs}} \\
\text{Message3} & S \to B: & M, \{N_a, K_{ab}\}_{K_{as}}, \{N_b, K_{ab}\}_{K_{bs}} \\
\text{Message4} & B \to A: & M, \{N_a, K_{ab}\}_{K_{ab}}
\end{array}
$$

理想化したプロトコル

元のプロトコルを BAN 論理で処理できる型式に抽象化 (idearlization と呼ぶ) したのが下記である。BAN 論理の特徴は、メッセージとして論理式を書くことが出来る点であり、これにより、メッセージの用途、例えば、共用キーであるといったことをその場で伝えることが出来る。

$$
\begin{array}{lll}
\text{Message1} & A \to B: & \{N_a, N_c\}_{K_{as}} \\
\text{Message2} & B \to S: & \{N_a, N_c\}_{K_{as}}, \{N_b, N_c\}_{K_{bs}} \\
\text{Message3} & S \to B: & \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \mid\sim N_c)\}_{K_{as}}, \\
 & & \{N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \mid\sim N_c)\}_{K_{bs}} \\
\text{Message4} & B \to A: & \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \mid\sim N_c)\}_{K_{as}}
\end{array}
$$

初期条件

BAN 論理では、なにが初期条件になるか、定式的に定義はされていない。この例では、以下のような論理式が仮定されている。

- 共用キーに関するもの。

$$A \mid\equiv A \overset{K_{as}}{\leftrightarrow} S \tag{2.20}$$
$$B \mid\equiv B \overset{K_{bs}}{\leftrightarrow} S \tag{2.21}$$
$$S \mid\equiv A \overset{K_{as}}{\leftrightarrow} S \tag{2.22}$$
$$S \mid\equiv B \overset{K_{bs}}{\leftrightarrow} S \tag{2.23}$$

- サーバははじめに A, B の共用キーになるものを知っている。

$$S \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B \tag{2.24}$$

- サーバは A、B の共用キーを管轄する。また、相手の話を正しく伝える。

$$A \mid\equiv (S \mid\Rightarrow A \overset{K}{\leftrightarrow} B) \tag{2.25}$$
$$B \mid\equiv (S \mid\Rightarrow A \overset{K}{\leftrightarrow} B) \tag{2.26}$$
$$A \mid\equiv (S \mid\Rightarrow (B \mid\backsim X)) \tag{2.27}$$
$$B \mid\equiv (S \mid\Rightarrow (A \mid\backsim X)) \tag{2.28}$$

- 生成する nonce は新鮮である。

$$A \mid\equiv \sharp(N_a) \tag{2.29}$$
$$B \mid\equiv \sharp(N_b) \tag{2.30}$$
$$A \mid\equiv \sharp(N_c) \tag{2.31}$$

プロトコル解析

以上の、条件から、プロトコルの解析を行う。

A から B へのメッセージ送信

$$B \lhd \{N_a, N_c\}_{K_{as}}$$

B はメッセージを見るが暗号を解読出来ないので、内容は理解できない。

B から S へのメッセージ送信

$$S \lhd (\{N_a, N_c\}_{K_{as}}, \{N_b, N_c\}_{K_{bs}})$$

S は結合の各要素を見ることができるから、下記となる。

$$S \lhd \{N_a, N_c\}_{K_{as}}, S \lhd \{N_b, N_c\}_{K_{bs}}$$

さらに、S は、共用キーを知っていることから、公理 (2.1) を適用して下記となる。

$$S \mid\equiv A \mid\sim (N_a, N_c)$$
$$S \mid\equiv B \mid\sim (N_b, N_c)$$

S から B へのメッセージ送信 B は下記のメッセージを見るが、キーを持たないので内容を理解できない。

$$B \lhd \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \mid\sim N_c)\}_{K_{as}}$$

一方、下記に対しては、公理 (2.1) を適用し、

$$B \lhd \{N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \mid\sim N_c)\}_{K_{bs}}$$

下記を得る。

$$B \mid\equiv S \mid\sim (N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \mid\sim N_c))$$

一方、初期条件 (2.30) と、公理 (2.15) より、下記が言える。

$$B \mid\equiv \sharp(N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \mid\sim N_c))$$

これらと公理 (2.4) より、下記が言える。

$$B \mid\equiv S \mid\equiv (N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \mid\sim N_c))$$

公理 (2.8) により、これを分解できる。

$$B \mid\equiv S \mid\equiv N_b \tag{2.32}$$
$$B \mid\equiv S \mid\equiv (A \overset{K_{ab}}{\leftrightarrow} B) \tag{2.33}$$
$$B \mid\equiv S \mid\equiv (A \mid\sim N_c) \tag{2.34}$$

式 (2.33) と、初期条件 (2.26) に公理 (2.5) を適用して下記を得る。

$$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

また、式 (2.34) と、初期条件 (2.28) に公理 (2.5) を適用して下記を得る。

$$B \mid\equiv A \mid\sim N_c$$

B から A へのメッセージ送信 下記のメッセージに対して、B の場合と同じ手順により、

$$B \lhd \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \mid\sim N_c)\}_{K_{as}}$$

以下を得る。

$$A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B \tag{2.35}$$

$$A \mid\equiv B \mid\sim N_c \tag{2.36}$$

さらに、式 (2.36) と、初期条件 (2.31) に公理 (2.4) を適用して、下記を得る。

$$A \mid\equiv B \mid\equiv N_c$$

このプロトコルを解析した結果、問題点は以下である。

- K_{ab} を暗号化キーとして使用しないので、相手がキーを知っているか確認できない。
- A は B が新鮮なメッセージを送信したことを知っているが B は A が新鮮なメッセージを送信したかどうかわからない。
- いくつかの冗長な項目がある。

2.1.5 The CCITT X.509 Protocol への適用例

CCITT X.509 のドラフトによれば、２人の参加者の間の、秘密を保つ通信を行う。互いに相手の公開キーを知っていることを前提とする。ここでは、３メッセージ版を対象とする。

プロトコル

$$\begin{aligned}
&\text{Message1} \quad A \to B: \quad A, \{T_a, N_a, B, X_a, \{Y_a\}_{K_b}\}_{K_a^{-1}} \\
&\text{Message2} \quad B \to A: \quad B, \{T_b, N_b, A, N_a, X_b, \{Y_b\}_{K_a}\}_{K_a^{-1}} \\
&\text{Message3} \quad A \to B: \quad A, \{n_b\}_{K_a^{-1}}
\end{aligned}$$

ここで、T_a, T_b はタイムスタンプ、N_a, N_b は nonce、X_a, Y_a, X_b, Y_b はユーザデータである。

理想化したプロトコル

$$
\begin{aligned}
&\text{Message1} \quad A \to B: \quad \{T_a, N_a, X_a, \{Y_a\}_{K_b}\}_{K_a^{-1}} \\
&\text{Message2} \quad B \to A: \quad \{T_b, N_b, N_a, X_b, \{Y_b\}_{K_a}\}_{K_a^{-1}} \\
&\text{Message3} \quad A \to B: \quad \{N_b\}_{K_a^{-1}}
\end{aligned}
$$

タイムスタンプ T_a, T_b を単なる nonce と考える。

初期条件

この例固有の初期条件は以下の通り。

- 各参加者は自分の秘密キーと相手の公開キーを知っている。

$$A \mid\equiv \overset{K_a}{\mapsto} A \tag{2.37}$$
$$B \mid\equiv \overset{K_b}{\mapsto} B \tag{2.38}$$
$$A \mid\equiv \overset{K_b}{\mapsto} B \tag{2.39}$$
$$B \mid\equiv \overset{K_a}{\mapsto} A \tag{2.40}$$

- 自分の nonce と相手のタイムスタンプが新鮮であることを信じる。

$$A \mid\equiv \sharp(N_a) \tag{2.41}$$
$$B \mid\equiv \sharp(N_b) \tag{2.42}$$
$$A \mid\equiv \sharp(T_b) \tag{2.43}$$
$$B \mid\equiv \sharp(T_a) \tag{2.44}$$

プロトコル解析

以下の手順により、プロトコル解析を行う。

A から B への送信

$$B \lhd \{T_a, N_a, X_a, \{Y_a\}_{K_b}\}_{K_a^{-1}}$$

公理 (2.2) より、

$$B \mid\equiv A \mid\sim \{T_a, N_a, X_a, \{Y_a\}_{K_b}\}_{K_a^{-1}}$$

となる。公理 (2.9) により、

$$B \mid\equiv A \mid\sim X_a$$
$$B \mid\equiv A \mid\sim \{Y_a\}_{K_b}$$

である。初期条件 (2.44) より、公理 (2.4) を用いて、

$$B \mid\equiv A \mid\equiv X_a$$

である。

B から A への送信 同様にして、

$$A \mid\equiv B \mid\sim X_b$$

および、

$$A \mid\equiv B \mid\equiv X_b$$

を得る。

A から B への送信 同様にして、

$$B \mid\equiv A \mid\equiv N_b$$

を得る。

問題点

このプロトコルでは、

$$B \mid\equiv A \mid\equiv Y_a$$
$$A \mid\equiv B \mid\equiv Y_b$$

を示せないことである。Y_a, Y_b は、受信者の公開キーにより暗号化されたメッセージの一部である。一般に、送り手が送った暗号化メッセージの各部分を知っているとは限らない。

2.1.6 BAN 論理の問題点

BAN 論理は、最小限の推論規則と論理式により、イントルーダの攻撃の可能性を検証することがてきることを、初めて示した歴史的に価値のある検証方法である。しかし、以下のような問題がある。

1. fresh という論理式により同一セッションのメッセージかどうかを判別しているが、この論理式は現在のセッション以前のセッションで送信されたメッセージかどうかを判別しているのみなので、他のセッションであっても、並行する同時に実行されているセッションからのすり替えを検証出来ない。また、同様の理由により、同一セッション内のメッセージ同士のすり替えを想定していない。
2. idearization という抽象化を人手により行う必要があり、この抽象化の手順は定式化されていないため、誤りや個人の主観が入り込みやすい。
3. 前提条件として、なにを用意すべきか、検証すべき対象となる論理式はなにかが、その時その時に応じてきめられており、原理実験としては問題ないが、実際のプロトコルの検証を機械的に多数行う場合には、個人の主観による違いが生じやすい。

2.2 SG 論理の概要

SG 論理によるセキュリティプロトコル検証の概要を、主に文献 [5] に基づいて説明する。

2.2.1 用語

SG 論理で用いる用語を以下に定義する。

$\mathcal{P}$: 参加者および参加者の識別子の集合を表す。

$[P, \ldots, Q]$: $\mathcal{P}$ の部分集合。

$\{K; m\}$: メッセージ m がキー K により暗号化または電子署名されたものをあらわす。

$h(m)$: メッセージ m のハッシュ値を表す。

K^{-1}; K の逆キーを表す。

K_{AB}: 参加者 A, B により共用される共用キーをあらわす。ただし、だれが実際にこのキーを保持しているかに関する論理的結論をだすことはできない。

(r, s): あたえられたプロトコルの実行 r のステップ s を表す。$s \leq s'$ の必要十分条件は $s = s'$ または、ステップ s をステップ s' より以前に実行である。$r \leq r'$ の必要十分条件は $r = r'$ または、r の第一ステップが r' より先に実行である。プロトコル r' の実行はプロトコル r の実行より先に終わることがある。

$\mathcal{T}(P, (r, s))$: 参加者 P が実行 r のステップ s で受理する全ての時間に依存するパラメータ（nonce, タイムスタンプ等）の集合を表す。

$\mathcal{R}(P, (r, s))$: 参加者 P が実行 r のステップ s で認識した全てのメッセージの集合。

$\mathcal{G}(P, (r, s))$: 参加者 P が実行 r のステップ s を実行する前に生成した可能性のあるすべてのメッセージの集合。

$\mathcal{G}(\mathcal{P}, (r, s))$: 全ての参加者 P が実行 r のステップ s を実行する前に生成した可能性のあるすべてのメッセージの集合。

nonce N_P をステップ k で生成し、あとのステップ k' で受け取るならば、$N_P \in \mathcal{T}(P, (r, s))$、ただし、$(k \leq s \leq k')$ である。

タイムスタンプ T_Q をステップ k からステップ k' の間受理するならば、$T_Q \in \mathcal{T}(P,(r,s))$、ただし $(k \leq s \leq k')$ である。

参加者 P がメッセージ m, m' をそれぞれステップ s, s' で受取り $(s \leq s')$、プロトコルの記述により参加者が $m = m'$ または $m \neq m'$ をチェックすることを、要求される時、$m \in \mathcal{R}(P,(r,s'))$ である。

参加者 P がステップ s でメッセージ m を受取り、プロトコル記述が参加者に、$m \in \mathcal{P}$ または $m \in \mathcal{T}(P,(r,s))$ のチェックを要求するなら、$m \in \mathcal{R}(P,(r,s))$ である。

プロトコル記述には、送信すべきメッセージのみでなく、メッセージの受信時になすべきチェックも記述もある。

定義１

$P \in \mathcal{P}$ を参加者、m をアトミックメッセージとする。参加者 P に関する実行 r、ステップ s におけるメッセージ m のタイプ τ は以下のように定義される。

$$\tau(P,m,(r,s)) :=$$

$$\begin{cases} n\overline{n_1}\ldots\overline{n_r} & \text{もし } n, n_1, \ldots, n_r \\ & \in \mathcal{R}(P,(r,s)) \\ & \text{かつ} \\ & P \text{ は以下をチェックした} \\ & m = n \text{ または} \\ & \text{すべての } j \in \{1,\ldots,r\} \\ & \text{に対して } m \neq n_j \\ \Box & \text{その他} \end{cases}$$

また、結合、ハッシュ、暗号化に対しては、下記である。

$$\tau(P,(m_1,\ldots,m_n),(r,s)) \\ := \tau(P,m_1,(r,s)),\ldots,\tau(P,m_n,(r,s))$$

$$\tau(P,h(m_1,\ldots,m_n),(r,s)) \\ := h(\tau(P,m_1,(r,s)),\ldots,\tau(P,m_n,(r,s)))$$

$$\tau(P, \{K; m_1, \ldots, m_n\}, (r, s)) :=$$
$$\begin{cases} (K, \tau(P, m_1, (r, s), \ldots, & \text{もし } P \text{ が } K^{-1} \text{ を持ち、} \\ \tau(P, m_n, (r, s))) & \{K; m_1, \ldots, m_n\} \\ & \text{に適用するなら} \\ \Box & \text{その他} \end{cases}$$

二つのメッセージは、P がそれらを区別できなければ、P に関して同じタイプを持つ。

$\mathcal{G}(P, (r, 1))$ は P がそれ以前のプロトコル実行で生成したすべてのメッセージを含む。$\mathcal{G}(P, (r, s))$ は、P がステップ 1 から $s-1$ で生成したすべてのメッセージを含む。また、P が並行するほかのセッションで生成したメッセージも含む。

2.2.2 論理式

SG 論理で使用する論理式は以下の通り。

$\epsilon : \mathbf{K} \to \mathbb{P}(\mathbf{P})$; 関数 ϵ はキー K から K を保有しているか、またはプロトコルの実行後に保有し、K^{-1} により生成されたテキストの電子署名をチェックできる全ての参加者の集合へのマップ。

$\delta : \mathbf{K} \to \mathbb{P}(\mathbf{P})$: 関数 δ はキー K から K を保有しているか、またはプロトコルの実行後に保有し、K^{-1} により生成されたテキストの暗号化を解読できる全ての参加者の集合へのマップ。

$in(P, V)$: P は V に含まれる。V として $\epsilon(K), \delta(K)$ を用いて、P がその目的に K を使用できるかを示す述語として用いる。

$number(V, W)$: $V, W \in \mathbb{P}(\mathbf{P})$ であり、V と W が同じ要素数。

$leg(x, y)$: x と y はプロトコルのステップであり、$x = y$ またはステップ x の後にステップ y を実行する。

$enc(K)$: K は暗号化、または K^{-1} により生成された電子署名のチェックに用いられる。

$sig(K)$ K は電子署名の生成、および K^{-1} により生成された暗号化メッセージの解読に用いる。

$sees(P, m, r, s)$: P は実行 r のステップ s でメッセージ m を受け取る。

$has(P, m)$: P は以前にメッセージ m を受取り、現在も保持している。

$said(P, K, m, r, s)$: P はメッセージ $\{K; m\}$ を送信したが受信していない。

$not\_said(P, K, m, r, s)$: P が (r, s) までに生成したメッセージのなかには、P と (r, s) に関して $\{K; m\}$ と同じタイプを持つメッセージ m' はない。すなわち、

$$\forall m' \in \mathcal{G}(P, (r, s))[\tau(P, m', (r, s)) \neq \tau(P, \{K; m\}, (r, s))]$$

である。

$recog(P, m, r, s)$: P は (r, s) でメッセージ m を認識する。すなわち、$m \in \mathcal{R}(P, (r, s))$ である。

$in\_time(P, m, r, s)$: P が (r, s) でメッセージ m をタイムリーな特性をもち受け取った。すなわち、$m \in \mathcal{T}(P, (r, s))$ である。

$determ(P, m)$: メッセージ m が P により決定された。これは、P がそれ以前のステップで m を受信していないことを表す。

$says(P, K, m)$: P は受理可能な期間内にキー K を用いて m と言った。すなわち現在のプロトコル実行中に言った。

$controls(P, \varphi)$: ステートメント φ に関して P は信頼できる。

$believes(P, \varphi)$: P は φ を信じ、そのように行動する。

$believes$ と $controls$ は BAN 論理に準拠して構成、$sees, said$ は BAN 論理の構成、$says$ は Abadi and Tuttle に準拠し、必要に応じて (r, s) に関して変更した。

意味的に ϵ, δ と等価な関数 E, D により、キーは $\mathbb{P}(\mathbf{P}) \times \mathbb{P}(\mathbf{P})$ の要素として表される。対称キー K_{PQ} では、

$$K_{PQ} = K_{PQ}^{-1}$$
$$E(K_{PQ}) = D(K_{PQ}) = E(K_{PQ}^{-1}) = D(K_{PQ}^{-1}) = [P, Q]$$

である。また公開キーアルゴリズム RSA によれば、私用キーで生成された暗号化テキストは、対応する公開キーで復号できる。($\{K_P; \{K_P^{-1}; m\}\} = m$) 従って、下記である。

$$E(K_P) = D(K_P) = \mathcal{P}$$
$$E(K_P^{-1}) = D(K_P^{-1}) = [P]$$

2.2.3 推論規則

SG 論理で使用する推論規則を以下に示す。以下の規則では、universal quantification は使用していない。従って、$P, Q, \ldots$ は、$\forall P, \forall Q, \ldots$ と読むべきである。各々のステートメントは、ステートメント $(\varphi_1, \varphi_2, \ldots)$、実行 $(r, r', \ldots)$、ステップ $(s, s', \ldots)$ に対して有効である。

plaintext recover rule(PR) いつ、参加者 P が暗号化テキストを復号できるかを述べる。このためには、暗号化テキストを受取り (see)、復号に必要なキーを持っている必要がある。復号化のためには、キーの品質に関する信頼はいらないことが示されている。

$$\frac{sees(P, \{K; m\}, r, s) \wedge in(P, \delta(K^{-1}))}{sees(P, m, r, s)}$$

message meaning rules(MM1,MM2) 暗号化テキストの受信からその生成者に関する結論を得る。P は暗号文を受信し、かつ対応する平文を持っていなければならない。共用キーではこれは明らかだが、非対称キーでは電子署名から平文を復元出来ず、この仮定を明確にチェックする必要がある。

さらに、メッセージの一部が P により認識されなければならない。もちろん、P は認証に必要なキーを持っており $(in(P, \epsilon(K)))$、だれがキーを持っているかを知っている $(believes(P, \delta(K^{-1}) = [P_1, \ldots, P_n]))$。すると P は後者の各々が暗号化テキストをつくる可能性があると結論する。

$$\frac{\begin{array}{r} sees(P, \{K^{-1}; m_1, \ldots, m_n\}, r, s) \wedge has(P, m_1, \ldots, m_n)) \\ \wedge(recog(P, m_1, r, w) \vee \ldots \vee recog(P, m_n, r, s)) \\ \wedge in(P, \epsilon(K)) \wedge believes(P, \delta(K^{-1}) = [P_1, \ldots, P_n]) \end{array}}{\begin{array}{r} believes(P, said(P_1, K^{-1}, (m_1, \ldots, m_n), r, s) \vee \ldots \\ \ldots \vee said(P_n, K^{-1}, (m_1, \ldots, m_n), r, s)) \end{array}}$$

非対称キーでは、秘密キーの保持者が電子署名を作ったと結論するに (MM1) は十分である。しかし、対象キーでは、P, Q のどちらかが暗号テキストを作ったと結論できるだけである。この場合、P が暗号化テキストを作れない場合に結論をだすもうひとつの推論規則 (MM2) が必要である。P が作った可能性のあるすべてのメッセージの集合中の暗号化テキストが、今問題にしている暗号化テキストとタイプが異なるなら真である。

$$\frac{not\_said(P, K, m, r, s)}{believes(P, \neg(said(P, K, m, r, s)))}$$

nonce verification rule(NV) もし P がメッセージ m が Q によりある種のタイムリーな性質を持って生成されたと信ずるなら、P はそのメッセージが受理可能な時間範囲で生成されたと信ずる。この範囲とは、現在のプロトコル実行、あるいは証明 (certificate) の場合は今と証明が生成された時の間隔である。いったん、メッセージが一定の時間内に生成されたと受け入れられたなら、生成の実行やステップは興味を持たれない。

$$\frac{\begin{array}{r} believes(P, said(Q, K, (m_1, \ldots, m_n), r, s)) \\ \wedge(in\_time(P, m_1, r, s) \vee \ldots \vee in\_time(P, m_n, r, s) \\ \vee in\_time(P, K, r, s) \end{array}}{believes(P, says(Q, K, (m_1, \ldots, m_n)))}$$

jurisdiction rule(J1,J2) J1 は BAN 論理とその拡張に基本的に同じであり、キーサーバの特別な役割に対応する。もし P が Q(キーサーバ) が、ステートメント ϕ に関して、信頼に足ると信じており、ステートメントが受理可能な期間に生成されたと信じるなら、P はステートメント φ を信じる。

$$\frac{believes(P, says(Q, K, \varphi)) \wedge believes(P, controls(Q, \varphi))}{believes(P, \varphi)}$$

J2 は参加者 P がキーの生成と使用の両方を行う (すなわち P はキーサーバ) 場合をカバーする。

$$\frac{\begin{array}{r} believes(P, says(Q, K', K)) \\ \wedge believes(P, controls(Q, in(Q, \delta(K^{-1})))) \wedge sig(K) \end{array}}{believes(P, in(Q, \delta(K^{-1})))}$$

has の規則 (H1,H2,H3,H4) 持つに関する規則である。

$$\frac{believes(P, says(Q, K, m))}{believes(P, has(Q, K))} \quad \frac{has(P,m)}{has(P,h(m))}$$

$$\frac{believes(P, says(Q, K, m))}{believes(P, has(Q, m))} \quad \frac{sees(P,m,r,s)}{has(P,m)}$$

key rules (K1,K2,K3,K4) k1,k2 は参加者が持っている任意のキーを使うことが出来ることに対応する。もちろんキーは目的に従ってし

か使用出来ない。

$$\frac{has(P,K) \wedge enc(K)}{in(P,\epsilon(K))}$$

$$\frac{has(P,K) \wedge sig(K)}{in(P,\delta(K)) \wedge believes(P,in(P,\delta(K)))}$$

$$\frac{\begin{array}{r}(believes(P,says(Q,K',(R,K))) \\ \vee believes(P,says(Q,K',(K,R)))) \wedge sig(K^{-1})\end{array}}{believes(P,says(Q,K',in(R,\delta(K^{-1}))))}$$

$$\frac{\begin{array}{r}believes(P,in(P_1,\delta(K)) \wedge \ldots \wedge in(P_n,\delta(K))) \\ \wedge number([P_1,\ldots,P_n],\delta(K))\end{array}}{believes(P,\delta(K) = [P_1,\ldots,P_n])}$$

K3 は暗号化テキストの一部として、参加者の名前 R をキー K と関連してどのように解釈するかを述べている。このメッセージの作成者は R が K に対応する K^{-1} を持っているという事実を表現しようとした。K4 は K を使って電子署名や証明を作れる参加者の集合に関する推論を許す。

Recognizability rules(R1,R2,R3,R4 ある種の前提のもと、メッセージの認識性は伝えられる。

$$\frac{recog(P,m,r,s)}{recog(P,h(m),r,s)}$$

$$\frac{recog(P,m,r,s) \wedge in(P,\epsilon(K))}{recog(P,\{K;m\},r,s) \wedge recog(P,\{K^{-1};m\},r,s)}$$

$$\frac{recog(P,m,r,s) \wedge in(P,\delta(K))}{recog(P,\{K;m\},r,s)}$$

$$\frac{recog(P,m_1,r,s) \wedge recog(P,m_2,r,s)}{recog(P,(m_1,m_2),r,s)}$$

Time rules(T1,T2,T3) ある種の前提のもと、メッセージの時間に間に合う性質は伝えられる。

$$\frac{in\_time(P,m,r,s)}{in\_time(P,h(n_1,\ldots,m,\ldots,n_k),r,s)}$$

$$\frac{\begin{array}{r}in\_time(P,m_i,r,s) \wedge in(P,\epsilon(K^{-1})) \\ \wedge \neg(m_i = \{K^{-1};m_1,\ldots,m_k\})\end{array}}{in\_time(P,\{K;,m_1,\ldots,m_k\},r,s)}$$

$$\frac{in\_time(P,K,r,s)}{in\_time(P,\{K;m\},r,s)}$$

composition,decomposition rules (C,D1,D2,D3,D4) 技術的な理由から我々は以下の合成分解規則を必要とする。

$$\frac{has(P,m_1) \wedge has(P,m_2)}{has(P,(m_1,m_2))}$$

$$\frac{sees(P,(m_1,m_2),r,s)}{sees(P,m_1,r,s) \wedge sees(P,m_2,r,s)}$$

$$\frac{has(P,(m_1,m_2))}{has(P,m_1) \wedge has(P,m_2)}$$

$$\frac{believes(P,said(Q,K,(m_1,m_2),r,s))}{believes(P,said(Q,K,m_1,r,s)) \wedge believes(P,said(Q,K,m_2,r,s))}$$

$$\frac{believes(P,says(Q,K,(m_1,m_2)))}{believes(P,says(Q,K,m_1)) \wedge believes(P,says(Q,K,m_2))}$$

ステートメント $\varphi_1,\ldots,\varphi_n$ に関する規則 (S) 我々は読者が first order 論理や様相論理の通常の規則を良く知っていると想定して、ステートメント $\varphi_1,\ldots,\varphi_n$ の規則に触れるだけにする。

$$\frac{believes(P,\varphi_1 \vee \ldots \vee \varphi_n) \wedge believes(P,\neg\varphi_1 \wedge \ldots \wedge \neg\varphi_{i-1} \wedge \neg\varphi_{i+1} \wedge \ldots \wedge \neg\varphi_n)}{believes(P,\varphi_i)}$$

2.2.4 Neuman-stubblebine プロトコル

Neuman-Stubblebine プロトコルのステップ 4 で B か受信するメッセージが S が生成する必要のないことを示す。

1. $\mathcal{T}(B,(r,s)),\mathcal{R}(B,(r,s)),\mathcal{G}(B,(r,s))$ と実行 r に先立つ前提条件
   B は 4 でメッセージ 4 を受信するから、

$$sees(B,(\{K_{BS};A,K_{AB},T_B\},\{K_{AB};N_B\}),r,4) \tag{2.45}$$

   である。また、B は 2 で $(B,\{K_{BS};A,N_A,T_B\},N_B)$ を送信するから、

$$\{K_{BS};A,N_A,T_B\} \in \mathcal{G}(B,(r,4)) \tag{2.46}$$

   である。プロトコルで用いるキーは対称キーであるから、

$$K_{BS} = K_{BS}^{-1} \tag{2.47}$$

$$K_{AB} = K_{AB}^{-1} \tag{2.48}$$

$$sig(K_{AB}) \tag{2.49}$$

$$number([P,Q],\delta(K_{AB})) \tag{2.50}$$

となる。K_{BS} は B, S に共用される対称キーであるから、

$$in(B, \delta(K_{BS}^{-1})) \tag{2.51}$$
$$in(B, \epsilon(K_{BS}^{-1})) \tag{2.52}$$
$$believes(B, \delta(K_{BS}^{-1}) = [B, S]) \tag{2.53}$$

である。キーサーバはセッションキーに関するステートメントにつき信頼されているから、

$$believes(B, controls(S, in(P, \delta(K)))) \tag{2.54}$$

となる。B はタイムスタンプ T_B と nonce N_B をステップ 2 で送信し、4 で受け取っているから、$\mathcal{T}(B, (r, 4)) = \{T_B, N_B\}$ である。特に、

$$in\_time(P, T_B, r, 4) \tag{2.55}$$

である。(NeSt93) のプロトコル記述は、B にステップ 4 で、A, T_B, N_B のチェックを要求しているが、K_{AB} のチェックは要求しない。特に、$N_A \neq K_{AB}$ はチェックしない。従って、$N_A \notin \mathcal{R}(B, (r, 4))$ かつ $\mathcal{R}(B, (r, 4)) = \{A, T_B, N_B\}$ である。この事から、

$$recog(B, A, r, 4) \tag{2.56}$$

となる。

2. 解析

プロトコルの目的の一つは A, B がセッションキー K_{AB} を共用すること。ゆえに、$believes(B, \delta(K_{AB}) = [A, B])$ を証明する必要がある。

$$(2.45) \stackrel{D1}{\rightarrow} \quad sees(B, \{K_{AB}; A, K_{AB}, T_B\}, r, 4) \tag{2.57}$$
$$(2.57), (2.51) \stackrel{PR}{\rightarrow} \quad sees(B, (A, K_{AB}, T_B), r, 4) \tag{2.58}$$
$$(2.58) \stackrel{H4}{\rightarrow} \quad has(B, (A, K_{AB}, T_B)) \tag{2.59}$$
$$\begin{aligned} (2.47), (2.52) \stackrel{MM1}{\rightarrow} \quad & believes(B, \\ (2.53), (2.56) \quad & \quad said(B, K_{AB}^{-1}, (A, K_{AB}, T_B), r, 4) \\ (2.57), (2.58) \quad & \vee said(S, K_{BS}^{-1}, (A, K_{AB}, T_B), r, 4)) \end{aligned} \tag{2.60}$$

$(r, 4)$ で (A, K_{AB}, T_B) と言ったのが S であることを結論するには、$not\_said(B, K_{BS}^{-1}, (A, K_{AB}, T_B), r, 4)$ が成立つか、すなわち、$\tau(B, m, (r, 4)) = \tau(B, \{K_{BS}; A, K_{AB}, T_B\}, r, 4)$ を満たす、メッセージ $m \in \mathcal{G}(B, (r, 4))$ が存在するかである。

(2) より $\{K_{BS}; A, N_A, T_B\} \in \mathcal{G}(B, (r, 4))$ であり、さらに、

$\tau(B, \{K_{BS}; A, N_A, T_B\}, (r, 4)) = (K_{BS}, A, \Box, T_B) =$
$\tau(B, \{K_{BS}; A, K_{AB}, T_B\}, r, 4)$ であるから、$not\_said(B, K_{BS}^{-1}, (A, K_{AB}, T_B), r, 4)$ は成立しない。
規則 (MM2) を適用できないので、B にとりセキュアであることは証明出来ない。実際、$\{K_{BS}; A, N_A, T_B\} \in \mathcal{G}(B, (r, 4))$ が、B が 4 で受け取る暗号化テキストと同じタイプであり、B はこれらを区別出来ない。従って、イントルーダは、このすり替えを B に気づかれずに行える。

3. 考えられる対策
   二つのテキストが B に関して区別できれば良い。従って、
   - B が 4 で $N_A \neq K_{AB}$ をチェックする。
   - 第二のメッセージで、N_A, T_B の順序を入れ換える (Syv93)。

   が考えられる。
   後者のやり方では、$not\_said(B, K_{BS}^{-1}, (A, K_{AB}, T_B), r, 4)$ が示せる。解析を続けると、さらに、$believes(B, \delta(K_{AB}) = [A, B])$ が言える。以下目的の達成は明らか。

このやり方は弱点は示せるが、プロトコルのセキュリティは示せない。

2.2.5 SG 論理の問題点

SG 論理は、BAN 論理の欠点を解決すべく提案された、セキュリティプロトコル検証法である。SG 論理では、メッセージの型を考えることにより、型によるすり替えの防止に関して検証がされている。BAN 論理で必要とされた idealization が不要になるなど、改善されている部分もあるが、下記の問題を持っている。

1. 型をチェックするための論理式において、その参加者が過去に受信した可能性のあるメッセージとの比較により、メッセージの再使用 (Replay) を検出するため、過去に受信した可能性のあるメッセージをリストアップする必要がある。これは、過去のプロトコルの実行の履歴を考える必要があり、しかもどこまで、過去にさかのぼれば良いのか名にも基準が示されていない。従って、どこまでの履歴を考慮すべきかは、検証担当者により決められる。
2. メッセージの新鮮さを評価するために in_time という論理式を使用しているが、これは並行セッションを十分に考慮しているとは言えず、BAN 論理における fresh と同様に、これにより、並行セッションからのすり替えの可能性を検証することは出来ない。
3. BAN 論理と同様に、前提条件として、なにを用意すべきか、検証す

べき対象となる論理式はなにかが、その時その時に応じてきめられており、原理実験としては問題ないが、実際のプロトコルの検証を機械的に多数行う場合には、個人の主観による違いが生じやすい。

Classification of attack			Checking Method	Previous Reseach	Class
Forging			*	BAN logic, SG logic	1
Replacement with the meaasge	of the different principals		**	BAN logic, SG logic	2
	of the same principals	in the different part of session	Type check	SG logic	3
		in the same part of session	***	not found	4

[Note]
* :Checking whether a message coded by keys which is not known by a intruder.
** :Checking of sender and reciever of a messge with the contents of message.
***:Checking of the coupling relations between submessages in each transmission .

図 2.1: メッセージすり替えのクラス分け

2.3 本研究の位置付け

本研究を従来の研究と比較して、以下のようにその位置付けを議論する。

2.3.1 対象とするイントルーダの攻撃手段による位置付け

プロトコル検証の主なテーマの一つは、各プロトコルの各データ転送の段階において、受信したメッセージが真に正当な参加者により作成されたものであるかどうかを決められるようになっているかを判断することである。我々は受信したメッセージをイントルーダが偽造できない部分とそれ以外に分けて、図 2.1 に示すカテゴリにより分類する。

偽造が可能であるかどうかは、イントルーダが復号キー保持していない暗号化キーにより暗号化されているがどうかにより決定される。偽造の可能なメッセージは、クラス 1 に分類される。次に、偽造ができないメッセージに対しては、イントルーダが過去に盗聴などにより入手した類似のメッセージとのすり替えが考えられる。このすり替えにも、同一の参加者ペアによるメッセージとのすり替えと、そうでないものがある。異なる参加者ペアとのすり替えの可能性のないこと検証は、メッセージ中に参加者を表す情報が含まれているかどうかにより行う。この攻撃をクラス 2 に分類する。最後に、同一参加者ペア間のすり替えを、プロトコルの同一部分同士のすり替えであるかどうかによりさらに分類する。同一部分同士のすり替え以外は、二つのメッセージの構造 (結合や暗号化の組み合わせかた) によりすり替えの可能性のないことの検証が可能である。しかしながら、同一の構造のメッセージ同士のすり替えの可能性のないことをさらに詳細に検証する方策を本論文で提案する。この攻撃をクラス 3 に分類する。同一部分同志のすり替えによる攻撃は、クラス 4 に分類する。BAN 論理は

クラス 1,2 に対処するものであり、SG 論理はクラス 1,2,3 に対処する方策である。この論文では、主に図 2.1 のクラス 4 に相当する攻撃に関する検証を行う。

2.3.2 検証手段による位置付け

本研究は、イントルーダによるセキュリティプロトコルへの攻撃の可能性のないことを検証するために、論理を使用している。我々の使用している検証方法では、通常 1 個のセッションの状態のみを考える。セッション間のメッセージのすり替えを検証する場合でも、たかだか、2 組のセッションの状態の組み合わせを、考えるのみであり、他のモデルチェックによる検証のように、想定される状態の組み合わせ、すなわち各参加者のプロセスが保有するメッセージの集合の組み合わせを全て生成する必要がない。

論理による推論規則をこのような範囲に限定しているとともに、その前提条件となる各種の集合を準備する際にも、セッションの状態の組み合わせを考える必要がないことが特徴である。これに対して、SG 論理ではメッセージの参加者 P から見たタイプを定義するに際して、メッセージの要素であるアトミックメッセージ毎に、過去に参加者 P が受信したメッセージと異なるメッセージであることをチェックする項目を設けている。このチェックすべき比較対象のデータを用意するためには、どのようなメッセージを受信する可能性があるかを検証者が考えて、メッセージの流れと状態を追って行く必要がある。

第3章　モデル

3.1　前提条件と用語

本節では本論文に共通する前提条件と用語に関して述べる。

3.1.1　前提条件

前提条件は以下の通りである。

プロトコル **Pro1** 扱うプロトコルは1種類のみとする。

Pro2 プロトコルは2人の参加者の間の通信を定義する。また、これら2人の参加者は役割を逆にすることはない。

Pro3 プロトコルは2人の参加者の交互の通信手順であり、プロトコルの流れは途中で分岐しない。

Pro4 プロトコルはイントルーダの攻撃のない場合、正しく動作する。例えば、保持していないメッセージを送信したりせず、キーは適切な参加者に配布されている。

Pro5 プロトコルで送信されるメッセージは、セッションの参加者以外の参加者の名前やキーを含まない。受信者の名前やキーが含まれているならそれを宛先を表す情報と解釈する。

Pro6 プロトコルでは、暗号化メッセージを受信した後、そのキーを受信して復号化することはないものとする。

セッション **S1** セッションはプロトコルの一連の実行である。

S2 セッションは並行して複数同時実行可能である。

S3 セッション間でメッセージの受け渡しはない。

S4 セッション中で使用されるすべてのメッセージはセッションの開始時に与えられる。

参加者 **Pri1** 正規の参加者はプロトコルに従い、不正をしない。また、プロトコルの送受信で想定されている内容、他の参加者の初期状態を知っている。

Pri2 メッセージ固有の特性 (共用キーなど) は、初期状態で保持しているものはその時点、受信したものなら受信した時点で参加者にわかる。

Pri3 セキュリティを守るために可能な限り受け取ったメッセージが想定されている内容か否かのチェックを行なう。

暗号 **E1** メッセージは暗号化キーにより暗号化したメッセージに変換され、復号化キーによりこの逆の変換がなされる。二種類のキーが同一の場合 (共用キー) と、異なる場合 (公開キーと秘密キー) がある。秘密キーで暗号化したメッセージを電子署名されたメッセージと呼ぶ。

E2 暗号化メッセージは完全で、キーなしでは解読できない。

E3 暗号化メッセージと同じ物を偶然に作ることはできない。

E4 暗号化メッセージの集合からキーを推定することはできない。

イントルーダとその攻撃 **I1** イントルーダは自分の秘密キーと、参加者の公開キー以外のキーは持っていない。

I2 送受信されたメッセージは全て傍受出来るとする。

I3 正規の参加者の送信するメッセージを傍受、改変、または横取りすることができる (但し、暗号化されたメッセージはキーがなければ復号や改変はできない)。

I4 正規の参加者になりすまして、偽造、あるいは過去に傍受したメッセージを送信することができる。(知らないキーで暗号化されたメッセージは偽造できない。)

3.1.2 用語

以下の説明と論理式中で用いられる用語を定義する。

参加者、プロセスおよびその状態

1. 参加者
   参加者は以下のいずれかである。

   A, B, C**:** 特定の参加者を表す定数。

   p, q 参加者を表す変数。

   以後、P, Q を参加者を表すメタ変数として用いる。

2. プロセス
   P が参加者である時、P^k で参加者 P が実行するセッション k のプロセスを表す。また、S^k_{Pi} で参加者 P が実行するセッション k のプロセスの i 番目の状態を表す。

ただし、単一のセッションに関することが明らかな場合は、k を省略する。

メッセージ

1. アトミックメッセージ
   アトミックメッセージとは、参加者を表す定数、あるいは、以下のいずれかである。

   $M, M_1, M_2, \ldots$: 単なるデータを表す定数。

   $N, N_1, N_2, \ldots$: nonce(必要に応じて生成される、過去に使用されたことのないデータ) を表す定数。

   $K, K_1, K_2, \ldots$: キーを表す定数。

   x, x_1, x_2: アトミックメッセージを表す変数。

   アトミックメッセージには下記のように参加者の情報を付加することが出来る。

   M_{jP}: 参加者 P が初期値として持つ単なるデータを表す定数。

   N_{jP}: 同上の nonce を表す定数。

   K_{jPQ}; 参加者 P と Q の共用キーを表す定数。

   K_{jP}: 参加者 P の公開キーを表す定数。

   K^{-1}_{jP}: 参加者 P の秘密キーを表す定数。

   ここで、$j = 0, 1, 2, \ldots$ であり、$j = 0$ の場合は省略する。アトミックメッセージの集合を **Atom** で表す。
2. メッセージ
   メッセージは以下のように帰納的に定義される。

   - アトミックメッセージはメッセージである。
   - X_1 と X_2 がメッセージならばそれらの連接 X_1, X_2 もメッセージである。
   - X がメツセージ、Z がキーならば X を Z で暗号化した $\{X\}_Z$ もメツセージである。

   また、以降では、X_j をメッセージを表すメタ変数として、Z_j をキーを表すメタ変数として用いる。$inv(Z_j)$:Z_j の逆キーを表す関数とす

る。($inv(K_{jP}) = K_{jP}^{-1}$、$inv(K_{jP}^{-1}) = K_{jP}$ かつ $inv(K_{jPQ}) = K_{jPQ}$)
ここで、$j = 0, 1, 2, \ldots$ であり、$j = 0$ の場合は省略する。

タグ付きメッセージ

受信したメッセージ中、および初期値として保持しているメッセージ中の同一のアトミックメッセージの、複数の出現を区別するために、アトミックメッセージ (例えば M_{1P}) の右肩にタグ w を付加したタグ付きアトミックメッセージ (M_{1P}^{w}) を考える。ここで、タグ w の値を以下のように定義する。

$w = iu$: i 番目の送受信されるひとまとまりのメッセージの中で、u 番目に出現するアトミックメッセージである。

$w = 0$: 各参加者が最初から保持しているアトミックメッセージである。

送受信に出現するメッセージおよび初期値として保持しているメッセージ X_j について、その中で出現するアトミックメッセージにすべてタグを付加したものをタグ付きメッセージとよび、$\bar{X_j}$ で表す。ここで、$\bar{X_j}$ のタグ $tag(\bar{X_j})$ の値は、全てのアトミックメッセージのタグを結合したものとして定義される。X_j の異なる出現については、そのタグの値は異なることがあるため、異なるタグの値を持つ同一のメッセージを区別するために、$\bar{X_j}, \bar{X_j}'$ なる記法を用いることがある。また、キー Z_j に対して、タグ付きのキーを $\bar{Z_j}$ であらわす。さらに、タグ付のアトミックメッセージを表す変数を $\bar{x}$、$\bar{x_1}$、$\bar{x_2}$ とする。また、メッセージに対応するタグの値は、例えば、$tag(\{M_{1A}^{12}, N_{2B}^{13}\}_{K_{3AB}^{14}}) = 121314$ である。メッセージに付けられたタグからそのメッセージか何番目の送受信メッセージであるか、あるいは初期値であるかを定める関数 $st(w)$ を下記のように定義する。

- $w = iu_1iu_2 \ldots iu_m$ の場合 $st(w) = i$
- $w = 0$ の場合 $st(w) = 0$

i 番目に送受信されるひとまとまりのタグ付きメッセージを定数 U_i で表す。

メッセージのタイプ

アトミックメッセージ X には、あらかじめタイプ $Type(X)$ が定義されている。$Type_P(X)$ は P が X の部分メッセージについて判定可能な型を可能な限り付加したメッセージ X の型である。ただし、X に含まれる暗

号化メッセージで、P がそれを解読するキーを持たない場合その型は $\Box$ となる。

$$
\begin{aligned}
&Type_P(X_1, X_2) = \\
&\qquad Type_P(X_1), Type_P(X_2) \\
&Type_P(\{X\}_Z) = \\
&\quad \begin{cases}
\{Type_P(X)\}_{Type_P(Z)} \\
\cdots P\ has\ \overline{inv(Z)}\text{が導ける時} \\
\Box \\
\cdots P\ has\ \overline{inv(Z)}\text{が導けない時}
\end{cases} \\
&Type_P(X) = Type(X) \quad \cdots X \in \mathbf{Atom}
\end{aligned}
$$

$P\ has\ \bar{Z}$ は後述の論理式であり、P が $\bar{Z}$ を持っていることをあらわす。

3.2 プロトコルのモデルと初期条件

3.2.1 プロトコルの送受信と参加者の状態

プロトコルのモデルにおける $1 \sim n$ 番目の送受信のうち、i 番目の送受信を下記のように記述する。

$$i : P \to Q : U_i$$

このとき、P^k の状態は U_i の送信時に S^k_{Pi-1} から S^k_{Pi} に遷移し、Q^k の状態は U_i の受信時に S^k_{Qi-1} から S^k_{Qi} に遷移する。これらは必ずしも同時ではない。また、セッション k の開始時の参加者のプロセス P^k の状態は S^k_{P0} である。

3.2.2 メッセージ集合と役割

P が初期メッセージ集合として保持するアトミックメッセージの集合を $Init(P)$ とする。

セッション毎に異なる値 (セッション固有値と呼ぶ) が生成されるアトミックメッセージの集合を **Session** とする。セッションをまたがって共通な、または、セッション毎に生成されるが異なる保証のないアトミックメッセージの集合を **Com** とする。

セッションの参加者の集合を **Prin**、セッションのすべてのタグ付アトミックメッセージの集合を **TagAM** とする。

3.3 セキュリティプロトコルの目的

ここでは、セキュリティプロトコルの目的として以下を取り扱う。

3.3.1 メッセージの伝達を一貫性を保って行う

相手に伝えるべきアトミックメッセージの集合を **Share** とする。このとき、一貫性のある伝達とは、セッションを構成するプロセスは、すべての $X \in \mathbf{Share}$ につき、すべて相手に伝達され、セッションを構成する他のプロセスからのみ X を受信し、X に関するすり替えがあればそれを検知できることを言う。

第4章　送信者の状態推定によるメッセージ認証の検証

4.1　章のまえがき

BAN 論理および SG 論理には、公開キーまたは受信者の秘密キーにより暗号化されたメッセージが信頼できる方法、すなわち、共通キーや送信者の秘密キーにより暗号化された形式で転送される場合、受信者は内側の暗号化メッセージの送信者が自分でそのメッセージのもととなる平文を知っていてそれを暗号化したのかとうか判断ができないという問題点がある。

このような状況の例として、図 4.1(a) の例を考える。ここで、各参加者のプロセスがプロトコルを実行する。プロセスの一連の実行をセッションと呼ぶ。各プロセスはプロトコルをシーケンシャルに実行する。そして、同一の参加者の異なるプロセスは各セッションを並行して実行することができる。

問題点を明らかにするために図 4.1(b) のプロトコルの例を考える。このプロトコルでは、参加者 A は第一のメッセージとして自分の nonce N_{1A} を送信する。参加者 B は、N_{1A} と参加者 A の識別子 A を自分の公開キー K_B で暗号化し、自分の nonce N_{2B} とともに参加者 A に送る。参加者 A は B の秘密キー K_B^{-1} を持っていないので、第二のメッセージ $\{A, N_{1A}\}_{K_B}$

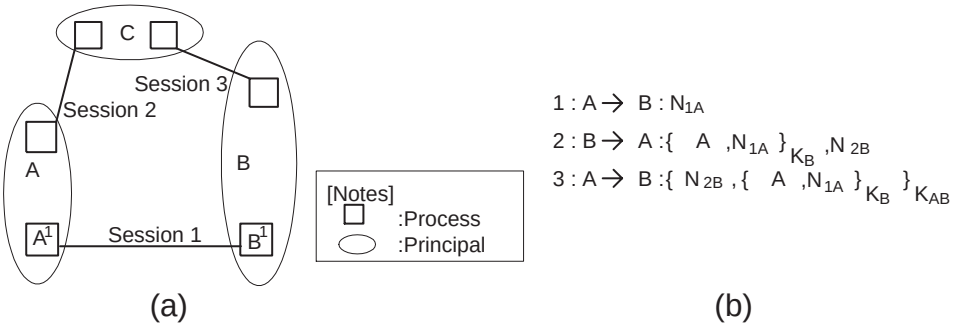

図 4.1: 参加者、プロセス、セッションとプロトコル

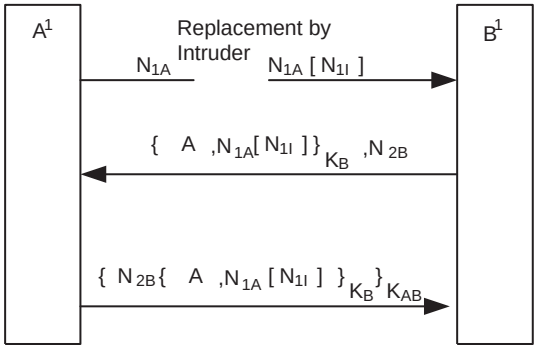

図 4.2: メッセージすり替えの例

を復号することができない。参加者 A は、この受信したメッセージを N_{2B} とともに、共用キー K_{AB} で暗号化して、B に送信する。

参加者 B は、第三のメッセージを受信し、全ての暗号を解読する。参加者 B は、A と B だけが共用キー K_{AB} を持っていることを知っているので、メッセージ $\{A, N_{1A}\}_{K_B}$ が参加者 A により作成されたことを信じる。しかし、彼が第三のメッセージの中のメッセージ N_{1A} が参加者 A により暗号化されたと結論するなら以下の例で示すように、それはいつも正しいとはかぎらない。

図 4.2 は、図 4.1(b) のプロトコルによるセッション 1 の実行を示す。ここで、A^1 および B^1 は、それぞれ参加者 A と B のプロセスを表わし、N_{1A}, N_{2B} はセッション 1 におけるプロセス A^1, B^1 の nonce を表わす。

図 4.2 において、第一のメッセージ N_{1A} は、転送の途中でイントルーダにより $N_{1A}[N_{1I}]$ とすり替えられている。(すり替えられた値をカギカッコで表わす) 各受信プロセスは自分の知っているメッセージとの食い違いがないのでこのすり替えに気が付かない。最後に、プロセス B^1 は N_{1A} の値が N_{1I} であると信じる。

このような攻撃の可能性を BAN 論理や SG 論理により安全ですり替えのない状態と区別することは出来ない。これらの論理では、受信者は公開キーにより暗号化されたメッセージの中身を信用しない。

4.1.1 問題解決のアプローチ

先に述べたような問題を解決するために、以下のアイデアを考えた。

1. プロトコル中で送信メッセージを組み立てるための同一の部分メッセージのオカレンスの間に優先順序を仮定する。この仮定はプロトコ

ルの設計者が自分の知っているメッセージのオカレンス[1]を良く知らないメッセージのオカレンス[2]のかわりに好んで使用することから自然である。

2. 送信者が送信したときに持っている各メッセージのオカレンスの集合を推定する。我々は送信メッセージ中の全ての部分メッセージについて全てのオカレンスを推定する必要がある。

これらのアイデアにより、我々は送信者がどの部分メッセージを平文から組み立てて使用したかを識別することができる。本章では、このアイデアに基づくメッセージ認証の検証方法を示す。

4.2 前提条件

3.1.1 の前提条件に加えて、本章固有の前提条件として下記がある。

プロトコル Pro7 プロトコル中で各転送における暗号化あるいは電子署名のメッセージはその内部の結合や暗号化の構造の違いにより互いにすり替えることは出来ない。

セッション S2 セッションは並行して複数同時実行可能であるが、同一参加者ペアのセッションは一度には一つしか実行されない。

参加者 Pri4 参加者は通信相手の参加者の状態 (所持するメッセージ) を推定できる。

Pri5 もし、参加者が送信しようとするメッセージの部分メッセージのオカレンスとして、他から受信したメッセージのオカレンスと自分の保持するメッセージから組み立てたオカレンスの両方をもっているなら、後者を使用しなければならない。

これら二つの仮定は、前節で述べた新しいアイデアを表わす。

4.3 論理による検証

3.3 で述べたプロトコルの目的の実現を検証するために、以下の論理的述語を用いる。

[1]メッセージの送信者はそれを平文から組み立てる。

[2]送信者はこのオカレンスを他の参加者から受信し、それは彼の持っていないキーにより暗号化されている。

4.3.1 論理式

本章で使用する論理式は以下に定義される基本命題を古典的論理結合子 $(\wedge, \vee)$ で拡張したものである。以下の基本命題式の記述方法は、BAN 論理のものを使用した。最初の 8 個の論理式は BAN 論理におけるものと同じである。但し、タグ付きのメッセージを含む論理式は、特定の出現場所のメッセージに関する論理式であり、これは本論文で導入された。

1. $P \models \varphi(X)$: P は持っているメッセージ X について、論理式 φ を信じる。論理式 φ には、$\models$ は含まれない。
2. $P \lhd \bar{X}$: P は $\bar{X}$ を受け取る。
3. $P \overset{Z}{\leftrightarrow} Q$: Z は P と Q の共用キーである。P, Q 以外は Z を知らない。
4. $\overset{Z}{\mapsto} P$: Z は P の公開キーである。P 以外は $inv(Z)$ を知らない。
5. $P\ has\ \bar{X}$: P は $\bar{X}$ を持っている。
6. $P \mid\sim \bar{X}$: P はかって $\bar{X}$ と言ったことがある。
7. $P\ says\ \bar{X}$: P からセッション内でメッセージ $\bar{X}$ を送受信する。そこで送られるメッセージ $\bar{X}$ は、その送受信のなかで改変されない。
8. $\sharp(X)$: X はこのセッションで作成された ($fresh(X)$ とも書く)。
9. $\bar{X_1} in \bar{X_2}$: $\bar{X_1}$ は $\bar{X_2}$ の構成要素。この論理式は文献 [7] の $\bar{X_2} contains \bar{X_1}$ に対応する。
10. $unforged\, X$: X はアトミックメッセージであって、イントルーダにより改変されていない。

11. $P\ canbuild_i\ \bar{X}$: P は i 番目の状態で所有するメッセージから $\bar{X}$ を作成することができる。
12. $P\ usable_i\ \bar{X}$: P は $\bar{X}$ と、i 番目の状態で所有するメッセージから U_i を作成することができる。
13. $P cbus_i \bar{X}$: P は $\bar{X}$ から $\bar{X_1}$ をとりだすことができる。ここで、$P canbuild_i \bar{X_1}$ であり、かつ、$P\ usable_i\ \bar{X_1}$ である。

4.3.2 前提となる論理式

プロトコルの初期条件から以下のように、前提となる論理式を定義する。

- $X \in Init(P)$ ならば、$P\ has\ \bar{X}$ ただし、$tag(\bar{X}) = 0$、かつ $P \models unforged\, X$ である。
- $i : P \rightarrow Q : U_i$ ならば $Q \lhd U_i$ である。
- $K_{jPQ} \in Init(P)$ ならば、$P \models P \overset{K_{jPQ}}{\leftrightarrow} Q$ である。

- $K_{jP}^{-1} \in Init(P)$ かつ $K_{jP} \in Init(Q)$ ならば $Q \models \overset{K_{jP}}{\mapsto} P$ である。
- $X \in Init(P)$ かつ $X \in \mathbf{Session}$ ならば、$P \models \sharp(X)$ である。

ここで、3 番目の論理式は、前提条件のプロトコルの項において、キーの配置が適切に行われることから定義される。また、4 番目の論理式は、これに加えて、参加者の項の他の参加者の初期状態を知っていることから定義される。

我々は BAN 論理において idealization と呼ばれる、元のプロトコルの抽象化を行わないので、転送されるメッセージにおけるキーの使い方は受信者に伝えられない。仮定より、各参加者は全ての転送されるメッセージの使われ方と構造を知っており、プロセスがキーを受信したならその使い方を知っているものとする。

4.3.3 メッセージの伝達

メッセージの確実な伝達の推論は以下の手順で行われる。本節の論理による検証は、BAN 論理と記述方式の違いはあるが同等の内容であるため、推論のあらすじのみを示し、その他の推論規則は 4.6 に採録した。

メッセージを送信したことがある

P と Q しか知り得ない共用キー Z で暗号化されたメッセージは、Q が作って送信したものである。異なる参加者ペア間のメッセージとすり替えられたなら、それを正しいメッセージと信じることはない。

$$\frac{P \lhd \overline{\{X\}_Z} \wedge P \models Q \overset{Z}{\leftrightarrow} P}{P \models Q \mid\sim \bar{X}}$$

同様にして、相手にしか作れない電子署名されたメッセージ $\bar{X}$ がある場合、以下の推論規則が使用される。

$$\frac{Q \neq P \wedge P \models \overset{inv(Z)}{\mapsto} Q \wedge P \lhd \overline{\{X\}_Z}}{P \models Q \mid\sim \bar{X}}$$

これらの規則は BAN 論理と同じである。

すり替えなく伝達

メッセージがすり替えられず、相手の参加者に伝達されることを確実な伝達と呼び、その十分条件を表す推論規則が以下のように定義される。

$$\frac{P \mid\equiv \sharp(X) \wedge P \mid\equiv Q \mid\sim \bar{X} \wedge Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X}))}{P \mid\equiv Q\ says\ \bar{X}}$$

$$\frac{P \mid\equiv Q\ says\ \bar{X} \wedge X \in Init(Q) \wedge X \notin Init(P)}{P \mid\equiv unforged\ X}$$

この第一の規則はBAN論理より拡張されており、i番目より以前の状態において、QがXを組み立てられかつ、XがU_iに使用できるという論理式が追加されている。

論理式 $says\ \bar{X}$ に関する分解規則は下記の通り。

$$\frac{P \mid\equiv Q\ says\ \overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)} \wedge Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X}))}{P \mid\equiv Q\ says\ \bar{X}}$$

$$\frac{P \mid\equiv Q\ says\ (\overline{X_1, X_2})}{P \mid\equiv Q\ says\ \bar{X_1}}$$

メッセージ生成規則

次に、送信者が送信メッセージ中のサブメッセージの内容を知っているかを決定する規則につき述べる。

1. 送信者はメッセージを作れる
   論理式 $P\ canbuild_i\ \bar{X}$ は、下記の規則による。
   - もし、Pがi番目の状態以前でXを持つなら、P can build $\bar{X}$である。

     $$\frac{P\ has\ \bar{X'} \wedge st(tag(\bar{X'})) \leq (i-1)}{P\ canbuild_i\ \bar{X}}$$

   - もし、Pが$\bar{X_1}$および$\bar{X_2}$を作れるなら、Pはその結合も作れる。

     $$\frac{P\ canbuild_i\ \bar{X_1} \wedge P\ canbuild_i\ \bar{X_2}}{P\ canbuild_i\ \overline{X_1, X_2}}$$

   - もし、Pが$\bar{X}$を作れ、かつPがi番目の状態以前で暗号化キーZを持つなら、Pは$\overline{\{X\}_Z}$を作ることが出来る。

     $$\frac{P\ canbuild_i\ \bar{X} \wedge P\ has\ \bar{Z'} \wedge st(tag(\bar{Z'})) \leq (i-1)}{P\ canbuild_i\ \overline{\{X\}_Z}}$$

2. 送信者は i 番目の送信メッセージ U_i を作るのに X を使える。
   論理式 $P\ usable_i\ \bar{X}$ は以下の規則による。

   - U_i それ自身が送信メッセージと見なせる。

$$\frac{}{\overline{P\ usable_i\ U_i}}$$

   - もし、結合 $\overline{X_1, X_2}$ がメッセージ U_i を作るのに用いられるなら、そして、P が $\bar{X_2}$ または $\bar{X_1}$ を作ることが出来るなら、$\bar{X_1}$ または $\bar{X_2}$ をそれぞれ送信メッセージ U_i に使用できる。

$$\frac{P\ usable_i\ \overline{X_1, X_2} \wedge P\ canbuild_i\ \bar{X_2}}{P\ usable_i\ \bar{X_1}}$$

$$\frac{P\ usable_i\ \overline{X_1, X_2} \wedge P\ canbuild_i\ \bar{X_1}}{P\ usable_i\ \bar{X_2}}$$

   - もし $\overline{\{X\}_Z}$ が送信メッセージ U_i に使用でき、P が i 番目の状態の前に Z を持っているなら $\bar{X}$ を送信メッセージに使用できる。

$$\frac{P\ usable_i\ \overline{\{X\}_Z} \wedge Phave\bar{Z'} \wedge st(tag(\bar{Z'})) \leq (i-1)}{P\ usable_i\ \bar{X}}$$

3. 送信者は $\bar{X}$ を知っていて、転送 U_i の一部分として $\bar{X}$ を使用できる。
   論理式 $P\ cbus_i\bar{X}$ は下記の規則による。

   - もし、P が $\bar{X}$ を作れ、$\bar{X}$ が送信メッセージを作成するのに使用できれば、P は $\bar{X}$ を作成でき、使用可能である。

$$\frac{P\ canbuild_i\ \bar{X} \wedge P\ usable_i\ \bar{X}}{P\ cbus_i\bar{X}}$$

   - もし P が $\overline{X_1, X_2}$ を作成可能で使用可能なら、P は $\bar{X_1}$ を作成可能で使用可能である。

$$\frac{P\ cbus_i\overline{X_1, X_2}}{P\ cbus_i\bar{X_1}}$$

   - もし、P が $\{X\}_Z$ を作成可能で使用可能であり、P が $inv(Z)$ を持っている (すなわち P が X を知っている) なら、P はまた X を作成可能で使用可能である。

$$\frac{P\ cbus_i\overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)} \wedge st(tag(\overline{inv(z)})) \leq (i-1)}{P\ cbus_i\bar{X}}$$

これらの規則を適用し、メッセージの伝達を、すべての $X \in \mathbf{Share}$ かつ、$X \in Init(Q)$ である X について論理式 $P \models unforged\ X$ を検証することにより、検証する。

4.4 適用例

本章の検証法の成果を、その方法を図 4.1 の例に適用し、$B \models unforgedN_{1A}$ を検証することにより示す。以下のように例のプロトコルにタグを付加する。

$$\begin{aligned} 1 : A \to B : & \quad N_{1A}^{11} \\ 2 : B \to A : & \quad \{A^{21}, N_{1A}^{22}\}_{K_B^{23}}, N_{2B}^{24} \\ 3 : A \to B : & \quad \{N_{2B}^{31}, \{A^{32}, N_{1A}^{33}\}_{K_B^{34}}\}_{K_{AB}^{35}} \end{aligned}$$

また、以下の前提条件を仮定する。

$$\begin{aligned} Init(A) = & \quad \{N_{1A}, A, K_B, K_{AB}\} \\ Init(B) = & \quad \{N_{2B}, A, K_B, K_B^{-1}, K_{AB}\} \\ \mathbf{Session} = & \quad \{N_{1A}, N_{2B}\} \\ \mathbf{Share} = & \quad \{N_{1A}\} \end{aligned}$$

この解析では、プロトコルが 4.2 の前提に従うことを仮定する。そこで、参加者 A は $\{A^{32}, N_{1A}^{33}\}_{K_B^{34}}$ を彼の知っているオカレンス A^0, N_{1A}^0 および K_B^0 (タグ 0 は参加者の初期値を表わす。) より作成して、第三の送受信を行うものとする。

第一に、A がその初期メッセージとして A^0 と N_{1A}^0 を持っていることから、$A\, canbuild_3\, A^{32}, N_{1A}^{33}$ を推論する。また、A が第 2 番目の状態において、N_{2B}^{24}, K_B^0 および K_{AB}^0 を持っていることから、$A\, usable_3\, A^{32}, N_{1A}^{33}$ を推論する。これから、$A\, cbus_3 A^{32}, N_{1A}^{33}$ を推論することが出来る。

次に、$B \lhd U_3$ および $B \models \sharp(N_{2B})$ から、$B \models A\, says\, \{A^{32}, N_{1A}^{33}\}_{K_B^{34}}$ を推論することが出来る。これにより、$B \models A\, says\, N_{1A}^{33}$ を推論出来る。

最後に、以上の結果および $N_{1A} \in Init(A)$ と $N_{1A} \notin Init(B)$ から、$B \models unforged\, N_{1A}$ を推論することが出来る。

4.5 章のまとめ

公開キーまたは受信者の秘密キーにより暗号化されたメッセージが信頼できる方法、すなわち、共通キーや送信者の秘密キーにより暗号化された形式で、転送されるようなプロトコルのカテゴリに関して研究した。このようなメッセージの認証の検証は BAN 論理 [2] や SG 論理 [5] により行うことは出来なかった。

これらのメッセージを認証するために以下のアイデアを提案した。

1. 我々は、送信者の内部でメッセージを組み立てる場合の同一のサブメッセージのオカレンス間の優先順位を仮定する。
2. セキュリティプロトコルの解析において、メッセージが送信される直前の状態での送信者の保持するメッセージのオカレンスの集合を推論する。

上記のアイデアの有効性を検証するために、基本的な前提条件と導出規則を提案した。BAN 論理の規則を拡張し、各プロトコルにおいてその検証できる範囲を広げた。

本章のこの検証規則をセキュリティプロトコルの簡単な例に適用した。本章の導出規則は、BAN 論理や SG 論理では可能でなかった、公開キーや受信者の秘密キーで暗号化され、共用キーや送信者の秘密キーによる別の暗号化に含まれるメッセージの転送の信頼性をチェックすることが出来る。

4.6 その他の論理式の定義

本文中で省略した論理式の定義を述べる。「含む」以外は BAN 論理と同じである。「含む」は文献 [7] の $\bar{X_2}\ contains\ \bar{X_1}$ に対応する。

1. メッセージの受信

$$\frac{P \lhd \overline{X_1, X_2}}{P \lhd \bar{X_1}} \quad \frac{P \lhd \overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)}}{P \lhd \bar{X}} \quad \frac{P \lhd \bar{X}}{P\ has\ \bar{X}}$$

2. 含む

$$\frac{\bar{X_1}\ in\ \bar{X_2}}{\bar{X_1}\ in\ \overline{(X_2, X_3)}} \quad \frac{\bar{X_1}\ in\ \bar{X_2}}{\bar{X_1}\ in\ \overline{\{X_2\}_Z}} \quad \overline{\bar{X}\ in\ \bar{X}}$$

3. 持っている

$$\frac{P\ has\ \overline{X_1, X_2}}{P\ has\ \bar{X_1}} \quad \frac{P\ has\ \overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)}}{P\ has\ \bar{X}}$$

4. 新鮮である。

$$\frac{\sharp(X_1)}{\sharp(X_1, X_2)}$$

第5章　メッセージすり替えの可能性のないことの検証

5.1　章のまえがき

プロトコルのセキュリティが守られているための基本的な項目として、イントルーダによってメッセージが改変され得ないことが重要である。通常はイントルーダの持っていないキーによる暗号化および電子署名を用いてガードされる。しかし、ガードされたメッセージ自体を他のメッセージとすり替えることは、データの型が同じメッセージが他に存在する場合は可能である。図 5.1 は、参加者 A から B へのガードされたメッセージの伝達を例として、４通りのすり替えの方法を示している。(1),(2),(3) に対してはすり替えの可能性のないことを検証する論理は提案されている。ここでは、(4) に関してその論理を提案する。

図 5.2 に示されるように、BAN 論理および SG 論理では、シリアルに実行される同一参加者ペアのセッション間の、キー K_{AB} によりガードされたメッセージのすり替えのチェックが行える。すなわち、当該セッション 8 において生成したメッセージ N_{1A}^{8} が含まれているならば[1]fresh あるいは in_time が成立し、ガードされたメッセージ全体が当該セッションにおいて生成されたと見なされる。従って、シリアルな他のセッションのメッセージによるすり替え (図 5.1 の (3)) が、検出できるプロトコルであるかを検証できることがわかる。同一セッション内のメッセージすり替え (図 5.1 の (2)) に関しては、BAN 論理では考慮されていないが、SG 論理では、型チェックを想定することにより、検証ができる。

従来の BAN 論理、SG 論理では、同一の参加者ペアの並行セッション間のガードされたメッセージのすり替えに関しては対応できない。例えば、図 5.3 のセッション 8 の最後で参加者 A のプロセス A^{8} が受信したメッセージに、当該セッションにおいて生成されたメッセージ N_{1A}^{8} が含まれていても、ガードされたメッセージ $\{N_{1A}^{8}, N_{2B}^{5}\}_{K_{AB}}$ の生成は、並行する別のセッション 5 において行われている。従って、N_{2B}^{8} として受信した N_{2B}^{5} を当該セッションの B^{8} が生成したものとあやまって判断してしまう。し

[1]本例では、簡単のためにキー以外のすべてのメッセージはセッションごとに異なる値を持つ nonce であるとする。

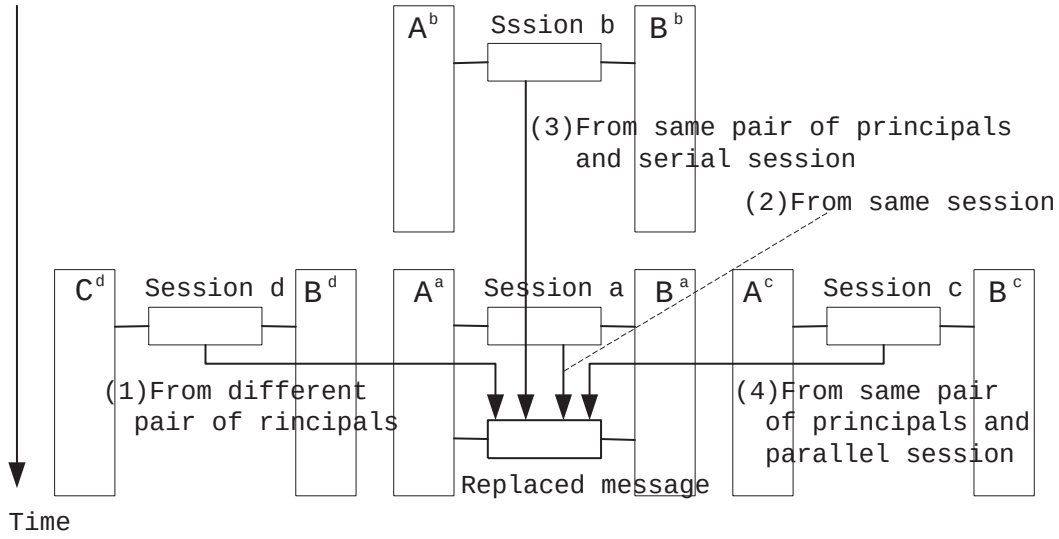

図 5.1: メッセージすり替えの 4 種類の方法

かしながら、BAN 論理ではすり替えの可能性がないと推論されてしまう。さらに型によりチェックが行われたとしても、本例の場合はこれも同一の $\{nonce, \Box\}_{Key}$ となり、すり替えを検出することはできない。にもかかわらず、SG 論理では同様にすり替えの可能性がないと推論される。

BAN 論理では、このような並行セッションの情報によるすり替えを考慮しておらず、これを検知することはできないことが指摘されている [1, 12, 13]。また、シリアルな実行では、すり替えが生じないが、パラレルな実行ではすり替えが生じるプロトコルの例も提案されている [9]。しかし、一般的な検出方法は、これまで提案されていない。

本章ではこの並行セッションのメッセージとのすり替えの検出について述べる。

5.1.1 本章のアイデアとその特徴

本章では図 5.4 に示すアイデアにより並行セッションの情報を利用したメッセージすり替えの可能性のないことの検証の問題を解決している。すなわち、すり替え元のすり替えられていないと保証された (送信者にとり unforged である) セッション k' のメッセージ (セッション毎に異なった値をもつ nonce) $N_{2B}^{k'}$ と、すり替え先のすり替えられていないと保証された (受信者にとり unforged な)、セッション k のメッセージ N_{1A}^{k} がすり替えるメッセージのまとまりの中で対応する位置にあり、$N_{1A}^{k} \neq N_{2B}^{k'}$ である

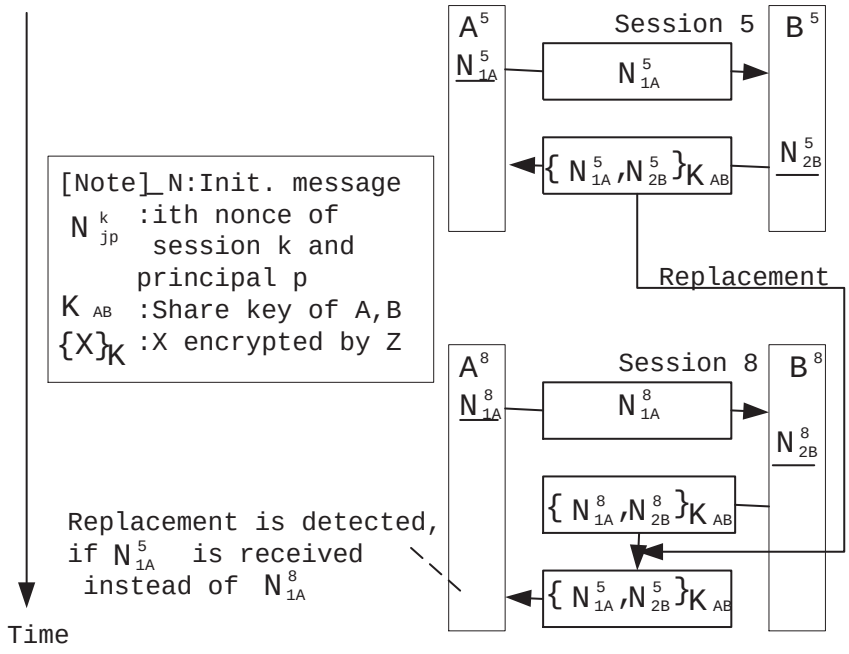

図 5.2: シリアルセッション間のメッセージすり替え

場合は、N_{1A}^{k} を含む暗号化メッセージ $\{\ldots N_{1A}^{k} \ldots\}_Z$ を、$N_{2B}^{k'}$ を含む暗号化メッセージ $\{\ldots N_{2B}^{k'} \ldots\}_Z$ ですり替えれば検知され得るので、このようにプロトコルが設計されていれば、すり替えることができない。ここで、unforged なメッセージとは、その参加者が最初から保持しているもの、あるいは他の参加者が unforged と見なしているメッセージを確実な伝達により受け取ったものを指している。また、$\{X\}_{K_{AB}}$ は X の A と B の共用キー K_{AB} による暗号化メッセージを表わす。例えば、図 5.3 の場合は、B にとって N_{2B}^{5} が unforged でなく、メッセージ $\{N_{1A}^{8}, N_{2B}^{8}\}_{K_{AB}}$ と $\{N_{1A}^{8}, N_{2B}^{5}\}_{K_{AB}}$ のすり替えの可能性のないことが言えないことがわかる。

本論文の方式の効果と特徴は、

- 同一の型の部分間のすり替えの検出を統一した方式で行うこと
- 既に参加者が保持するメッセージ自体がすり替えられている可能性を考慮し、信頼できるメッセージのみをすり替えの可能性のないことの検証に使用すること。

である。

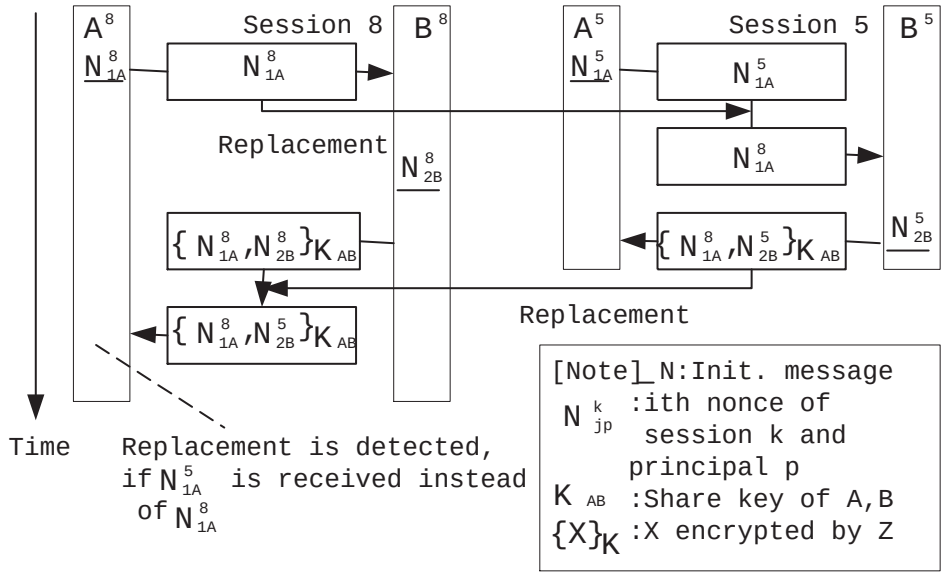

図 5.3: 並行セッション間のメッセージすり替え

5.2 プロトコル

5.2.1 前提条件

本報告プロトコルの前提条件のうち 3.1.1 と異なる点は以下の通りである。

セッション S5 セッションの初期状態 (開始時に与えられるメッセージ) は全て同一である。

参加者 前章の 4.2 の前提条件を同様に追加する。

5.2.2 論理式

本章で用いられる基本論理式のうち前章と異なるものは以下のように定義される。ただし、前章の fresh は使用しない。また、基本論理式と $\neg$, $\vee$ 等の論理演算により得られる式も論理式である。またさらに、参加者とメッセージ変数に関する全象束縛と存在束縛を加えたものも論理式である。

プロトコルば単一種別であり、すべてのセッションは同一の動作をする。従って、すべてのセッションにおいて論理式は同じ値となるので、二つのセッションで受け渡されるメッセージの比較をする論理式 (cores,replace) 以外は、セッションの識別子 k を省略して考える。また、これらの基本論

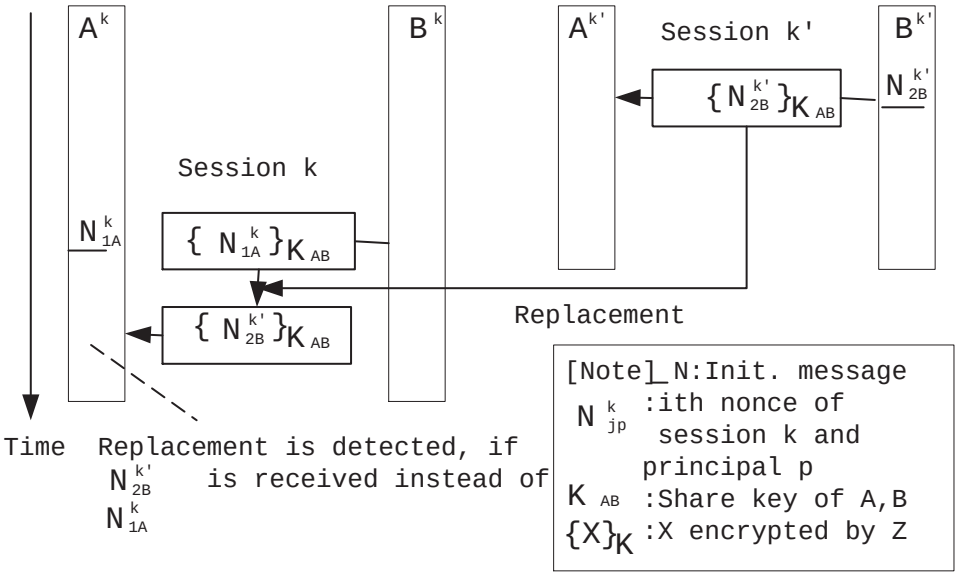

図 5.4: すり替えチェックのアイデア

理式 ($P \overset{K}{\leftrightarrow} Q, \overset{K}{\mapsto} P, unforged\, M$) は、メッセージに対して成立する論理式であり、タグを付加していないメッセージを対象とする。

6. $P \mid\!\sim \bar{X}\, to\, Q$: P はかって Q に $\bar{X}$ と言ったことがあり、Q は $\bar{X}$ を受け取った。[1]
7. $P\, says\, \bar{X}\, to\, Q$: P から Q へセッション内でメッセージ $\bar{X}$ を送受信する。そこで送られるメッセージ $\bar{X}$ は、他のセッションの同一送受信における、同一部分メッセージとのすり替えを除き、その送受信のなかで改変されない。[1] [2]
14. 欠番。
15. $auf\ X$: (almost unforged)X はアトミックメッセージであって、他のセッションの同一送受信における同一部分メッセージとのすり替えを除き、改変されていない。
16. $\bar{X}\ isto\ P$: $\bar{X}$ は宛先 P の情報を含むメッセージである。
17. $replaceunit \bar{X}$: $\bar{X}$ は一つの通信の途中で、イントルーダによりそのメッセージをすり替えの可能性がある。しかし、イントルーダにより、$\bar{X}$ の部分メッセージがその部分のみ、すり替えられることはない。
18. $unreplaceable\ \bar{X}$: 受信した $\bar{X}$ は他のセッションの同一送受信におけ

[1] $to\ Q$ は本章で拡張した部分であり、メッセージ $\bar{X}$ の送り先が Q であることを確認していることを表す。

[2] 同一部分とのすり替えに関する部分は、本章で拡張した部分である。

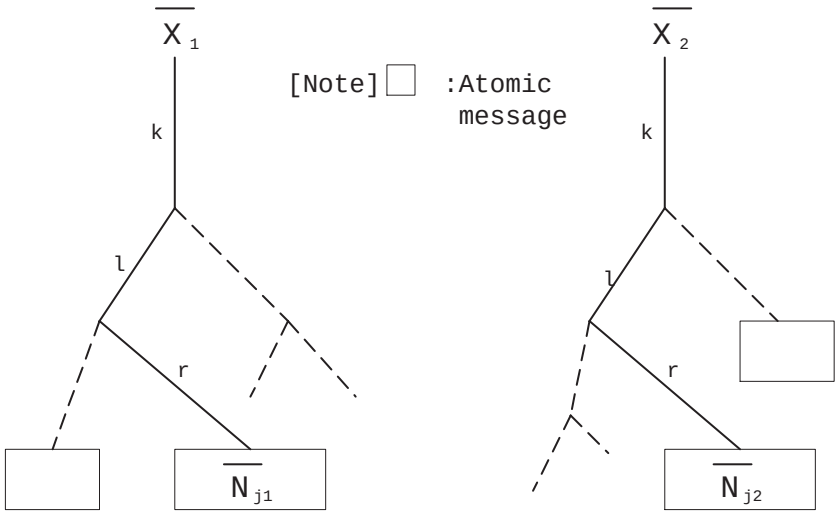

図 5.5: cores の関連

る同一部分メッセージとのすり替えを除き、すり替えられている可能性がない。

また、$\bar{X_1}$ と $\bar{X_2}$ の二つのメッセージからトップダウンに構造を解析して葉がアトミックメッセージとなる解析木を作成し、結合の左側には l、右側には r、暗号化には k のラベルをつけた場合に、ルートからのラベルの列が互いに一致する部分でかつ両者がアトミックメッセージであるペア $(\bar{X_{j_1}}, \bar{X_{j_2}})$ の集合を求める関数 $Cores(\bar{X_1}, \bar{X_2})$ を以下のように定義する (図 5.5)。

$$
Cores(\bar{X_1}, \bar{X_2}) = \begin{cases} \{(\bar{X_1}, \bar{X_2})\} & (X_1, X_2 \in \textbf{Atom} \text{ の場合}) \\ Cores(X_{11}, X_{21}) \cup Cores(X_{12}, X_{22}) & \\ \qquad (X_1 = (X_{11}, X_{12}) \wedge & \\ \qquad X_2 = (X_{21}, X_{22}) \text{ の場合}) & \\ Cores(X_{11}, X_{21}) \cup \{(Z_1, Z_2)\} & \\ \qquad (X_1 = \{X_{11}\}_{Z_1} \wedge & \\ \qquad X_2 = \{X_{21}\}_{Z_2} \text{の場合}) & \\ \{\quad\} \qquad (\text{その他の場合}) & \end{cases}
$$

5.2.3 特徴的な基本推論規則

推論規則のうち、本章の特徴を表す規則につき、以下に述べる。実際のすり替えられていないことを推論する規則は次の節以降に述べる。その他の推論規則に関しては、前章と同じである。

メッセージを送信したことがある

P と Q しか知り得ない共用キー Z で暗号化されたメッセージは、Q が作って送信したものである。異なる参加者ペア間のメッセージとすり替えられたなら、それを正しいメッセージと信じることはない。

$$\frac{P \lhd \overline{\{X\}_Z} \wedge P \models Q \overset{Z}{\leftrightarrow} P}{P \models Q \mid\sim \bar{X}\ to\ P}$$

同様にして、相手にしか作れない電子署名されたメッセージ $\bar{X}$ があり、$\bar{X}$ の中にあて先をあらわす情報を含む (isto) 場合、以下の推論規則が使用される。

$$\frac{P \models \bar{X}\ isto\ P \wedge P \models \overset{inv(Z)}{\mapsto} Q \wedge P \lhd \overline{\{X\}_Z}}{P \models Q \mid\sim \bar{X}\ to\ P}$$

メッセージのすり替え

イントルーダが作成することの出来ない部分メッセージ $\bar{X}$ 全体の、同一の参加者ペア間の他の部分メッセージとのすり替えは、$\bar{X}$ の受信者を Q とした時、同一の型 $Type_Q(X)$ のメッセージが、プロトコルの他の部分に存在しなければ不可能である。メッセージ $\bar{X}$ についてこのような部分が存在しないことを $unreplaceable\ \bar{X}$ で表す。このような X に関する推論規則は下記である。

1. まず、送受信されるひとかたまりのメッセージの中で、イントルーダが自由に更新可能な部分メッセージ X を $changeable\ \bar{X}$ とする。

$$\frac{changeable\ \overline{X_1, X_2}}{changeable\ \bar{X_1}}$$

$$\frac{changeable\ \overline{\{X\}_{K_{jP}}}}{changeable\ \bar{X}}$$

$$\overline{changeable\ U_i}$$

2. 次に $changeable\ \bar{X}$ であるが、イントルーダが作り替えることが出来

ないものを $replaceunit\ \bar{X}$ という。

$$\frac{changeable\ \overline{\{X\}_{K_{jPQ}}}}{replaceunit\ \overline{\{X\}_{K_{jPQ}}}}$$

$$\frac{changeable\ \overline{\{X\}_{K_{jP}^{-1}}}}{replaceunit\ \overline{\{X\}_{K_{jP}^{-1}}}}$$

3. $replaceunit \bar{X}$ の内で、他の部分とのすり替えが出来ない部分を $unreplaceable \bar{X}$ とする。

$$\frac{replaceunit\ \bar{X} \wedge Sametype(\bar{X}^k) = \phi}{unreplaceable\ \bar{X}}$$

$$\frac{unreplaceable\ \bar{X} \wedge \bar{X_1}\ in\ \bar{X}}{unreplaceable\ \bar{X_1}}$$

ここで、関数 $Sametype(\bar{X}^k)$ は下記のように、$\bar{X}^k$ と同じ型のタグ付きメッセージの集合として定義される。

$$\begin{aligned} Sametype(\bar{X}^k) = \{ & \bar{X_1}^{k_x} \mid \exists U_i, Q[\\ & \bar{X}\ in\ U_i \wedge Q \lhd U_i \wedge \\ & (Type_Q(X_1) = Type_Q(X)) \wedge \\ & (tag(\bar{X_1}^{k_x}) \neq tag(\bar{X}^k) \wedge \\ & ((k \neq k_x) \vee \\ & (k = k_x \wedge st(tag(\bar{X_1}^{k_x})) \leq st(tag(\bar{X}^k)))\} \end{aligned}$$

ここで、2 行目は Q が $\bar{X}^k$ の受信者であることを表わし、3 行目は同一の型であること、4 行目は同一部分でないことを表わす。5 行目は異なるセッションを表わし、6 行目は同一セッションで $\bar{X_1}^{k'}$ を含む送受信が $\bar{X}^k$ を含む送受信より後ろでないことを表わす。また、ここで、k_x は、セッション k および、セッション k とは異なるあるセッション k' を要素とするドメインとする、セッションを表わす変数である。

上記の $Sametype(\bar{X}^k) \neq \phi$ を含む推論規則は、$\bar{X}^k$ と同一の型ですり替え可能なメッセージがプロトコル内に他にない場合を表している。他に同一の型ですり替え可能のメッセージが存在する場合 ($Sametype(\bar{X}^k) \neq \phi$) については次の節で述べる。

5.2.4 セッション間のメッセージすり替えチェック

セッション k の P から Q への送受信 U_i^k 中のメッセージ $\bar{X}^k$ を対象にすり替えのないことの必要条件を考える (図 5.6)。

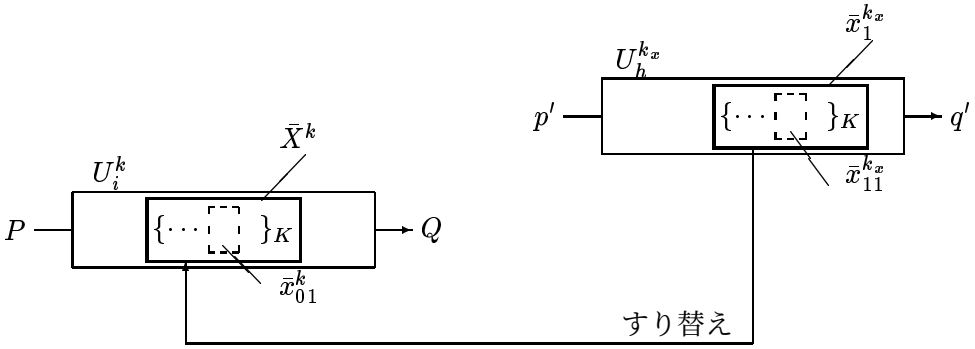

図 5.6: 同一参加者ペアによるセッション間のメッセージすり替えチェック

すり替えの元となり得るセッション k_x のメッセージ ${\bar{x_1}}^{k_x}$ の集合を前述の関数 $Sametype(\bar{X}^k)$ で定義する。すなわち、すり替えの元はすり替え先と同一型のメッセージである。また、異なるセッションの同一送受信の同一メッセージとのすり替えは次章において議論するのでここでは対象外とする。

すり替えの可能性がないためには、まず、全てのすり替え可能な元のメッセージ ${\bar{x_1}}^{k_x} \in Sametype(\bar{X}^k)$ について以下が成立することが必要である。$(\bar{x}_{01}^k, \bar{x}_{11}^{k_x}) \in Cores(\bar{X}^k, {\bar{x_1}}^{k_x})$ について以下の三つの条件を同時に満たすものが一つでも有れば、$\bar{X}^k$ と ${\bar{x_1}}^{k_x}$ のすり替えを参加者が検知可能である。

1. $\bar{x}_{01}^k$ と $\bar{x}_{11}^{k_x}$ は異なるメッセージである。

$$(x_{01} \neq x_{11} \vee (k \neq k_x \wedge x_{01} \in \mathbf{Session})) \tag{5.1}$$

   ここで、$x_{01} \in \mathbf{Session}$ は、x_{01} がセッション固有値であることを表わす。

2. Q はアトミックメッセージ $\bar{x}_{01}^k$ を見ることができ、x_{01} は Q の初期状態である。または、Q はメッセージ $\{{\bar{X_3}}^k\}_{\bar{x}_{01}^k}$ を見ることができ、$inv(x_{01})$ は Q の初期状態である。

$$\begin{aligned}(Q\ has\ \bar{x_{01}} \wedge x_{01} \in Init(Q)) \vee \\ (Q\ has\ \{\bar{X_3}\}_{\bar{x}_{01}} \wedge inv(x_{01}) \in Init(Q))\end{aligned} \tag{5.2}$$

3. p' は $\bar{x}_{11}^{k_x}$ または $\{{\bar{X_2}}^{k_x}\}_{\bar{x}_{11}^{k_x}}$ を持っていて送信した。かつ、以下のいずれかが成り立つ。x_{11} は p' の初期状態、あるいは、q' の初期状態であ

り、p' は q' から同一メッセージとのすり替え以外のない送受信 (says) で x_{11} を受取りかつ、x_{01} と x_{11} とは同一ではない。q' から says で x_{11} をもらった場合、これが同一部分同士ですり替えられている可能性があることから、$x_{01} = x_{11}$ ならすり替えの検出が出来ない可能性がある。また、$k_x \neq k$ の時、U_h から $\bar{x_{11}}^{k_x}$ を取り出した後、U_h を相手に渡さずセッション k_x を中断する可能性があるため、状態 $(h-1)$ までで、says によるメッセージの伝達がなされている必要がある。ここで、$U_h^{k_x}$ は、$\bar{x}_{11}^{k_x}$ を含む送受信の単位である。

$$\begin{aligned}
\exists p', q' \in \mathbf{Prin}[((p'\ cbus_h \bar{x}_{11} \vee \\
p'\ cbus_h \{\bar{X_2}^{k_x}\}_{\bar{x}_{11}}) \wedge h = st(tag(\bar{x}_{11}))) \wedge \\
q' \lhd U_h \wedge p' \neq q' \wedge \\
(x_{11} \in Init(p') \vee \\
(p' \models q'\ says\ \bar{x}'_{11}\ to\ p' \wedge \\
x_{11} \in Init(q') \wedge p'\ has\ \bar{x}'_{11} \wedge x_{01} \neq x_{11} \wedge \\
(k_x \neq k \vee st(tag(\bar{x}'_{11})) \leq h-1)))]
\end{aligned} \tag{5.3}$$

これらより、$Sametype(\bar{X}^k) \neq \phi$ の場合の、イントルーダが同一の型のメッセージ同士ですり替えのできない条件は、

$$\begin{aligned}
&replaceunit\ \bar{X} \wedge (\forall \bar{x_1}^{k_x} \in Sametype(\bar{X}^k) \\
&[\exists (\bar{x_{01}}^k, \bar{x_{11}}^{k_x}) \in Cores(\bar{X}^k, \bar{x_1}^{k_x}) \\
&\frac{[\text{式 (5.1)}\wedge\text{式 (5.2)}\wedge\text{式 (5.3)}]])}{unreplaceable\ \bar{X}}
\end{aligned} \tag{5.4}$$

である。

ここで、変数のドメインが有限集合 $\{t_1, \ldots, t_m\}$ であるような全象束縛と存在束縛に関して、下記の推論規則を導入する。

$$\frac{\varphi[t_1/y] \wedge \ldots \wedge \varphi[t_m/y]}{\forall y \in \{t_1, \ldots, t_m\}[\varphi]}$$

$$\frac{\varphi[t_i/y]}{\exists y \in \{t_1, \ldots, t_m\}[\varphi]}$$

ただし、φ は論理式、$\varphi[t/y]$ は論理式 φ の中の変数 y の自由出現を t で置き換えることを表す。

5.2.5 参加者間でのメッセージ伝達の保証

以下の条件を満足すれば、メッセージ X が別セッションの同一メッセージとのすり替えを除き参加者から参加者に伝達される。

$$\frac{\begin{array}{r} Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X})) \wedge \\ P \mid\equiv Q \mid\sim \bar{X}\ to\ P \wedge unreplaceable\ \bar{X} \end{array}}{P \mid\equiv Q\ says\ \bar{X}\ to\ P}$$

$$\frac{P \mid\equiv Q\ says\ \bar{X}\ to\ P \wedge X \in Init(Q) \wedge X \notin Init(P)}{P \mid\equiv auf\ X}$$

また、$says$ の分解規則として、以下のものがある。暗号化あるいは、電子署名の分解に関しては、送信者が中身のメッセージ $\bar{X}$ を見て送信し、その解読のキーを受信者が持っているときに限り分解が可能である。

$$\frac{\begin{array}{r} P \mid\equiv Q\ says\ \overline{\{X\}_Z}\ to\ P \wedge \\ P\ has\ \overline{inv(Z)} \wedge Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X})) \end{array}}{P \mid\equiv Q\ says\ \bar{X}\ to\ P}$$

結合の分解は下記による。

$$\frac{P \mid\equiv Q\ says\ (\overline{X_1, X_2})\ to\ P}{P \mid\equiv Q\ says\ \bar{X_1}\ to\ P}$$

5.2.6 従来の研究との比較

BAN 論理では以上の検証のうち、その一部である

$$\frac{P \mid\equiv Q \mid\sim X \wedge P \mid\equiv \sharp(X)}{P \mid\equiv Q\ says\ X}$$

を検証しているのみである。また、同一セッション内のメッセージとのすり替え、並行して実行されるセッションのメッセージとのすり替えの検証が不十分である。

SG 論理では、同一セッション内のメッセージのすり替えに関して型を考えて検証をしているが、並行して実行されている別のセッションのメッセージとのすり替えのチェックは不十分である。

5.3 適用例

本論文の方式でチェックできるが SG 論理の方式ではすり替えが検知出来ない例として、下記のプロトコルを説明する。

$$
\begin{aligned}
1 : A \rightarrow B : &\quad N_{1A}^{11} \\
2 : B \rightarrow A : &\quad \{N_{1A}^{21}, N_{4B}^{22}, A^{23}\}_{K_B^{-1}{}^{24}} \\
3 : A \rightarrow B : &\quad \{N_{4B}^{31}, N_{2A}^{32}, N_{3A}^{33}, B^{34}\}_{K_A^{-1}{}^{35}} \\
4 : B \rightarrow A : &\quad \{N_{2A}^{41}, N_{5B}^{42}, A^{43}\}_{K_B^{-1}{}^{44}}
\end{aligned}
$$

この例では、A から B へ nonce を送り、その応答メッセージを 3 回送りあっている。ここで、伝達すべきメッセージ、初期値、メッセージの型等の主な初期条件を下記とする。

$$
\begin{aligned}
Init(A) &= \{A, B, N_{1A}, N_{2A}, N_{3A}, K_B, K_A, K_A^{-1}\} \\
Init(B) &= \{A, B, N_{4B}, N_{5B}, K_A, K_B, K_B^{-1}\} \\
\mathbf{Session} &= \{N_{1A}, N_{2A}, N_{3A}, N_{4B}, N_{5B}\} \\
\mathbf{Com} &= \{A, B, K_B, K_A, K_B^{-1}, K_A^{-1}\} \\
\mathbf{Share} &= \{N_{2A}, N_{3A}, N_{4B}, N_{5B}\} \\
Type(N_{1A}) = Type(N_{2A}) &= Type(N_{3A}) = N \\
Type(N_{4B}) &= Type(N_{5B}) = N \\
Type(K_A) &= Type(K_B) = K \\
Type(K_A^{-1}) &= Type(K_B^{-1}) = K \\
Type(A) &= Type(B) = I
\end{aligned}
$$

本例では、第一の送受信以外は、全て送信側により電子署名がなされている。さらに、受信側の参加者名がメッセージとして含まれている。また、同一の参加者ペア間のすり替えは、メッセージの型が $\{N, N, I\}_K$ である第 2 の送受信 U_2 と第 4 の送受信 U_4 の間で考えられる。

1. U_2^k と $U_4^{k'}$ とのすり替え
   内部のメッセージで対応する位置にある異なるものは、N_{1A}^{k21} と $N_{2A}^{k'41}$、N_{4B}^{k22} と $N_{5B}^{k'42}$ である。ここで、$N_{1A} \in Init(A)$ より $A \models unforgedN_{1A}$ である。また、第 3 の送受信にはすり替えがないので、$B \models AsaysN_{2A}^{32}toB$ が言える。また、

$$
\begin{aligned}
B\ cbus_4 N_{2A}^{41} \\
N_{2A} \in Init(A) \\
B\ has\ N_{2A}^{32} \wedge st(tag(N_{2A}^{32})) = 3
\end{aligned}
$$

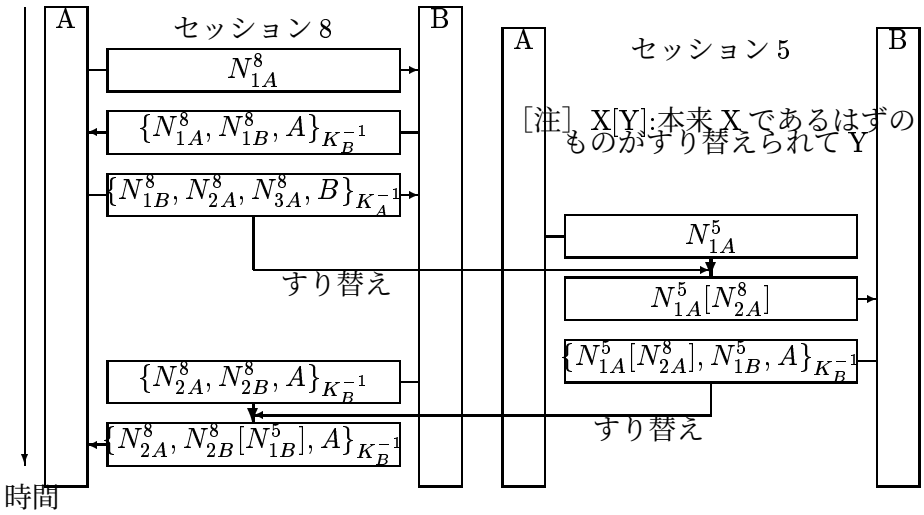

図 5.7: 適用例に存在するすり替え

であることから、N^{k21}_{1A} と $N^{k'41}_{2A}$ に式 (5.3) を適用すると真である。従って、規則 (5.4) より、

$$unreplaceable\ \{N^{23}_{1A}, N^{22}_{4B}, A^{23}\}_{K_B^{-1\,24}}$$

である。従って、すり替えは出来ない。

2. U^k_4 と U^k_2 とのすり替え
同様にして、$A \models unforged\ N_{2A}$ である。しかし、

$$unreplaceable\ \{N^{41}_{2A}, N^{42}_{5B}, A^{43}\}_{K_B^{-1\,44}}$$

が示せないので、すり替えの可能性が否定出来ない。すなわち、規則 (5.4) は適用出来ない。

以上より、すり替えの可能性のあることがわかる。実際、図 5.7 に示すような、すり替えが存在する。

一方、この例を SG 論理により検証すると、SG 論理における型の定義より

$$\tau(\{N_{1A}, N_{4B}, A\}_{K_B^{-1}}) = \{N_{1A}, \Box, A\}_{K_B^{-1}}$$
$$\tau(\{N_{2A}, N_{5B}, A\}_{K_B^{-1}}) = \{N_{2A}, \Box, A\}_{K_B^{-1}}$$

となり、型は異なる。したがってすり替えはできないという結論になる。従って、本論文の方式によれば、SG 論理では考慮されていないすり替えを見逃さないことがわかる。

5.4 章のまとめ

プロトコルのセキュリティ上重要な問題であるメッセージの確実な伝達に関して、従来チェックが不十分だった並行セッションとのメッセージすり替えの可能性を指摘し、メッセージのすり替えの可能性のないこと、すなわちこのプロトコルではすり替えが行われれば、必ず参加者が検知可能であることを検証する方法を推論規則として定式化した。

本章で述べた検証方式の効果を、従来 BAN 論理や SG 論理では攻撃の可能性がないと検証されてしまう並行セッションの情報を利用した型攻撃について、本章の方式では誤った結論を導かないことを例で示した。

5.5 論理式の定義

本文中で省略した論理式の定義を以下に示す。

1. 名前を含む

$$\frac{P \lhd P^{w'} \wedge P^{w'}\ in\ \bar{X}}{P \mid\equiv \bar{X}\ isto\ P}$$

$$\frac{P \lhd \{\bar{X_1}\}_{K_P^{w'}} \wedge \{\bar{X_1}\}_{K_P^{w'}}\ in\ \bar{X}}{P \mid\equiv \bar{X}\ isto\ P}$$

$$\frac{P \lhd K_{PQ}^{w'} \wedge K_{PQ}^{w'}\ in\ \bar{X}}{P \mid\equiv \bar{X}\ isto\ P}$$

$$\frac{\begin{array}{r} P \lhd \{\bar{X_1}\}_{K_{PQ}^{w''}} \wedge \\ \{\bar{X_1}\}_{K_{PQ}^{w''}}\ in\ \bar{X} \end{array}}{P \mid\equiv \bar{X}\ isto\ P}$$

第6章　別セッションの同一部分間のすり替えの可能性のないことの検証

6.1　章のまえがき

本章では、これまで述べた送信者の状態推定によるメッセージの認証の検証と、メッセージすり替えの可能性のないことの検証に加えて、別セッションの同一部分間のすり替えの可能性のないことの検証を行う。

また本章では、セキュリティプロトコルの目的を、電子商取引への適用を前提として、下記の 2 項目とし、イントルーダの攻撃により 、これらの目的が達成できない可能性を検証する。

プロトコルの一貫した実行 プロトコルを実行し、正常終了したプロセスがセッションの最初から最後まで同一の定められた相手のプロセスからのメッセージのみを受信し、他のセッションからメッセージとすり替えられたりしない。

正常終了の一致 あるセッションでプロトコルを実行している複数のプロセスのうち一つがセッションを正常に終了したと判断するならば、残りのプロセスも同一の判断を下す。

本章の問題点を明らかにするために、以下のプロトコルの例を考える。

$$
\begin{aligned}
1 : A \to B : &\quad \{\text{“}Doyouhave'', M_{1A}, N_{1A}\}_{K_{AB}} \\
2 : B \to A : &\quad \{\text{“}Yes'', N_{2B}\}_{K_{AB}} \\
3 : A \to B : &\quad \{\text{“}Howmuch'', N_{1A}\}_{K_{AB}} \\
4 : B \to A : &\quad \{M_{2B}, N_{2B}\}_{K_{AB}}
\end{aligned}
$$

このプロトコルは、参加者 A から参加者 B に、製品 M_{1A} の在庫、および価格 M_{2B} を問い合わせるものである。ここで、M_{1A}, M_{2B} および $''\ldots''$ は一般的なメッセージを表している。イントルーダは共用キー K_{AB} を知らないから、送受信される各メッセージを改変することは出来ない。しかし、イントルーダはメッセージをすり替えることができる。

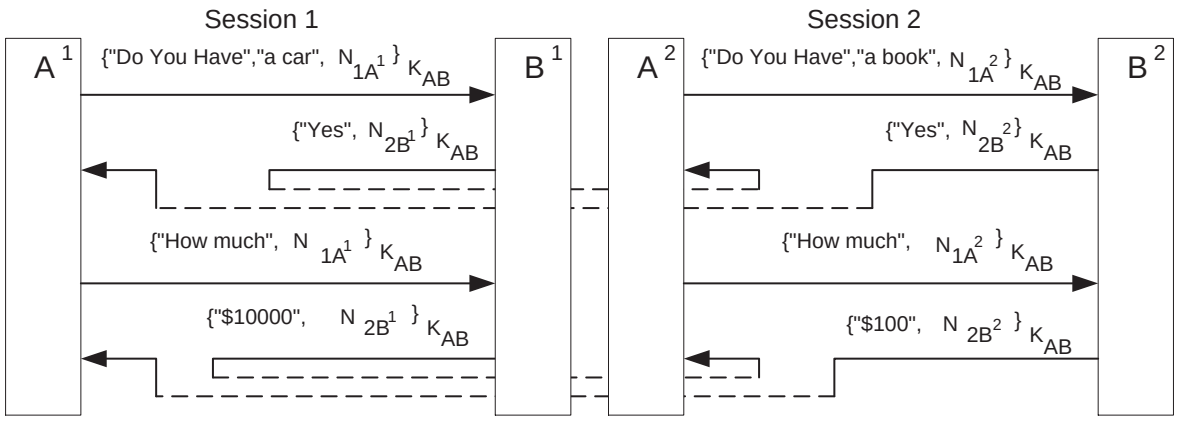

図 6.1: 同一部分同志のすり替えによる攻撃の例

図 6.1 は、二組の参加者ペア A^1, B^1 と A^2, B^2 により並行して実行される上記のプロトコルのセッション 1,2 の実行状況を表している。図 6.1 において、各問い合わせに対する応答メッセージがイントルーダによりすり替えられ (点線により表示) ているが、受信した A^1 や A^2 は自分の既に持っているメッセージと矛盾が生じないので、このすり替えに気付くことはない。この攻撃により、参加者 A は自動車の価格が 100 ドルと思い、本の価格が 10000 ドルと思ってしまう。実際上は取引がこのあと両方とも成立すれば、お金の被害は全体としてはないが、このプロトコルを使用した取引の信用は失われてしまう。また、片方の取引が取り消された場合、なんらかの被害が発生する可能性がある。

この例は、プロトコルのセッション中の同一部分同志のすり替えによる攻撃の例である。このような攻撃の可能性は、BAN 論理では検出することが出来ない。また、各参加者のメッセージのオカレンスの間に矛盾が生じないので、SG 論理においてもこれを検出することは出来ない。

6.2 セキュリティプロトコルの目的

本章では、3.3 で述べた目的に加えて、セキュリティプロトコルの目的として以下を取り扱う。

6.2.1 セッション正常または異常終了の判断の一致

セッション k を実行するプロセス P^k, Q^k は最後の 2 個のメッセージ U_{n-1}, U_n の送受信により図 6.2 のように状態が遷移する。U_{n-1} の伝達が

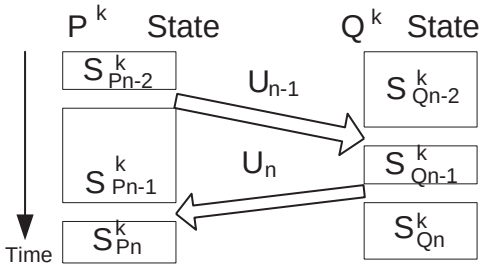

図 6.2: プロセスの最終状態遷移

イントルーダにより妨害されて、Q^k に伝わらない場合は、P^k, Q^k はそれぞれ状態 S^k_{Pn-1}, S^k_{Qn-2} にとどまり、時間がたてばセッションは異常終了し双方の判断に不一致はない。U_n の伝達がイントルーダにより妨害されて、P^k に伝わらない場合は、P^k, Q^k はそれぞれ状態 S^k_{Pn-1}, S^k_{Qn} にとどまり、P^k はセッションが異常終了、Q^k はセッションが正常終了したと判断する。このような判断の不一致を単純なメッセージの伝達のみで防止することはできない。判断の一致が必須であるオンラインシステムなどでは、2 フェーズコミットプロトコルにより、この問題を解決している [3]。しかし、通常、実際のプロトコルでは最後のメッセージに関してメッセージの消失の可能性のある場合は、受信側が送信側に問い合わせて、判断することで、送信側との食い違いを防止している。本論文においても、最後のメッセージに関しては、このような補助的な手段があるものとする。

U_{n-1}, U_n が送信されていないのにイントルーダが代理で送信することができ、受信者がこれを見破れない時、送信者の状態が進んでいない場合でも、受信者の状態が進んでしまう。これを防止するには U_{n-1}, U_n の中にイントルーダには作成できない情報が入っている必要がある。このように、U_{n-1}, U_n を本来の相手から受信したことが保証できれば、互いに相手の状態が最終状態まで進むことが保証される。

6.3 前提条件

本報告のプロトコルの前提条件は 5.2.1 と同じである。

6.4 論理による検証

6.2 の二つの目的を達成していることを検証するために、以下の形式言語を用いる。

6.4.1 論理式

本論文で使用する論理式は前章で述べた論理式および以下に定義される基本命題を古典的論理結合子 $(\wedge, \vee)$ で拡張し、さらに参加者とメッセージ変数に関する全象束縛と存在束縛を加えたものである。

19. $guard_Q\ \bar{X_1}\bar{X_2}$: $\bar{X_1}$ と $\bar{X_2}$ は暗号化または電子署名によりイントルーダが更新出来ない同一のメッセージ (ガードされたメッセージ) の一部として送信され、Q に受信される。
20. $onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2'}$: $guard_Q\ \bar{X_1}\bar{X_2}$ が成立するような **Session** の要素 $\bar{X_1}\bar{X_2}$ において、X_1 はメッセージの受信者 Q の初期メッセージ集合の要素であり、X_2 は送信者 $P(\neq Q)$ の初期メッセージ集合の要素である。また、この送受信の後、Q から P へ、ガードされたメッセージの一部として、$\bar{X_2'}$ が返信される。
21. $X_1 \sim_P X_2$: X_1 と X_2 は同一のガードされたメッセージの一部として送信され P に受信される。
22. $X \in \mathbf{Set}$: X が集合 **Set** の要素である。
23. $X \notin \mathbf{Set}$:X が集合 **Set** の要素ではない。

6.4.2 前提となる論理式

プロトコルの初期条件から前章と同様にして以下のように、前提となる論理式を定義する。

- $X \in Init(P)$ ならば、$P\ has\ \bar{X}$ ただし、$tag(\bar{X}) = 0$、かつ $P \models\!\!\models unforged\, X$ である。
- $i : P \to Q : U_i$ ならば $Q \lhd U_i$ である。
- $K_{jPQ} \in Init(P)$ ならば、$P \models\!\!\models P \overset{K_{jPQ}}{\leftrightarrow} Q$ である。
- $K_{jP}^{-1} \in Init(P)$ かつ $K_{jP} \in Init(Q)$ ならば $Q \models\!\!\models \overset{K_{jP}}{\mapsto} P$ である。

ここで、3 番目の論理式は、前提条件のプロトコルの項において、キーの配置が適切に行われることから定義される。また、4 番目の論理式は、これに加えて、参加者の項の他の参加者の初期状態を知っていることから定義される。

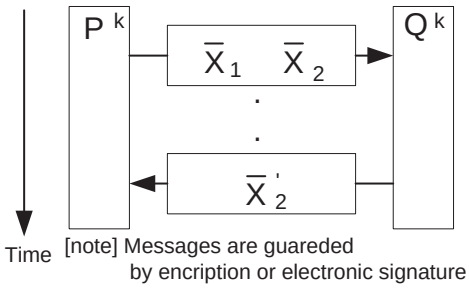

図 6.3: 1 組のプロセスによるセッション固有値の共有

6.4.3 メッセージの伝達

メッセージの確実な伝達の推論は前章の推論に加えて以下の手順で行われる。

メッセージの一貫性を保った伝達

同一参加者が他のセッションのプロセスで送信した同一部分のメッセージとのすり替えのないことの検証を以下の推論規則により行う。まず、X_1 と X_2 が P から Q へ同時に確実な伝送によって伝えられることを表す論理式 $guard_Q\ \bar{X_1}\bar{X_2}$ を下記の推論規則の結論とする。

$$\frac{\bar{X_1}\ in\ \bar{X} \wedge \bar{X_2}\ in\ \bar{X} \wedge Q \models P\ says\ \bar{X}\ to\ Q}{guard_Q\ \bar{X_1}\bar{X_2}}$$

さらに $guard_Q\ \bar{X_1}\bar{X_2}$ が成立する P から Q へ同時に伝えられるメッセージ X_1 と X_2 がそれぞれ異なる参加者 Q, P が同一部分のすり替えを除いて相手に伝達したアトミックメッセージであり、かつ **Session** の要素である場合に、この送信が相手に到着すれば、互いに相手の初期メッセージ集合の要素であり **Session** の要素であるアトミックメッセージを共有することになる。

図 6.3 [1]に示すように、上記伝達の後で、Q から P へ P の初期値である $\bar{X_2}'$ が確実な伝達で返される場合、上記の送信が相手に到着したことが確認される。上記の送受信は P^k, Q^k 以外には送信や受信ができず、他のプロセスがセッション k の X_1 や X_2 を保有していても、それ以後の状態に進むことはできない。このような伝達が存在することを $onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2}'$

[1]メッセージは暗号化または電子署名によりガードされているものとする。

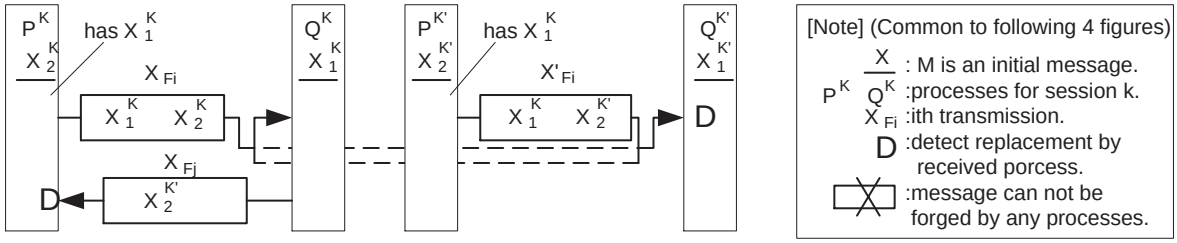

図 6.4: すり替えの検出パターン 1

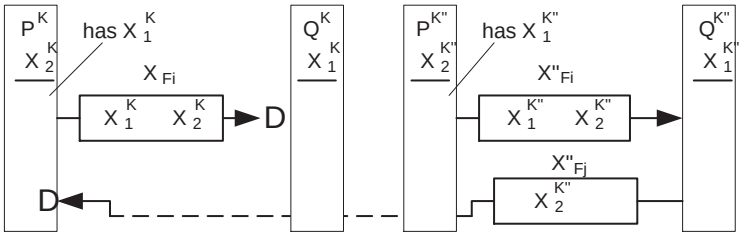

図 6.5: すり替えの検出パターン 2

と記述する。そこで、以下の推論規則を導入することができる。

$$\begin{aligned} & guard_Q\ \bar{X_1}\bar{X_2} \wedge \\ X_1 \in \mathbf{Session} \wedge X_2 \in \mathbf{Session} \wedge & \\ P \mid\equiv auf\ X_1 \wedge Q \mid\equiv auf\ X_2 \wedge & \\ st(tag(\bar{X_2})) < st(tag(\bar{X_2}')) \wedge & \\ \frac{P \mid\equiv Q\ says\ \bar{X_2}'\ to\ P}{onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2}'} & \end{aligned}$$

この規則の健全性は以下のように証明される。

定理 1 $onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2}'$ が推論されたなら、このプロトコルにもとづいた全てのセッションにおいて、$st(tag(\bar{X_2}'))$ 番目の状態以降は、イントルーダのいかなる攻撃のもとでも、そのセッションの固有値である X_1 と X_2 をともに持つプロセスは *1* 組のみであり、その他のプロセスは X_1 と X_2 のいずれも持たない。

証明

$\bar{X_1}, \bar{X_2}$ の送信と、$\bar{X_2}'$ の送信をそれぞれ

$$i : P \to Q : \quad U_i \quad \text{および} \quad j : Q \to P; \quad U_j$$

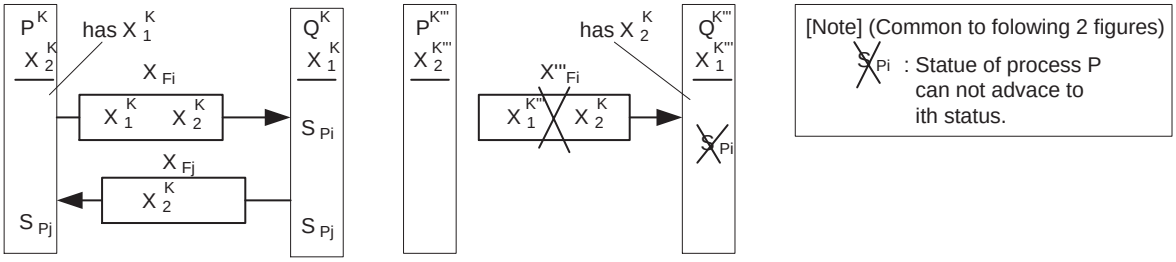

図 6.6: すり替えの検出パターン 3

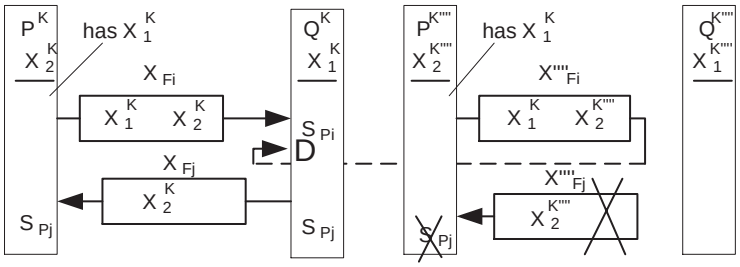

図 6.7: すり替えの検出パターン 4

とする。ここで、$onetoonecon\,\bar{X_1}\bar{X_2}\bar{X_2'}$ が推論されたと仮定する。すると、これを証明する規則は上記の規則のみであるから、前提である $Q \models auf X_1$, $P \models auf\, X_2$ が成立する。さらに、$Q \models auf\, X_1$, $P \models auf\, X_2$ を証明する規則は 1 個のみであり、その前提条件である $X_1 \in Init(Q)$、$X_2 \in Init(P)$ がそれぞれ成立している。ここで、セッション k を考え、プロセス P^k, Q^k の初期値として X_1^k, X_2^k をそれぞれ与える。またここで、本証明ではセッション k の X_1 を X_1^k と、$\bar{X_1}$ を $\bar{X_1}^k$ と表す。プロセス P^k が U_i を Q^k に送る。もし、Q^k がイントルーダの攻撃によりこの U_i を受け取らず、かわりに $P^{k'}$ の送信した $\bar{X_1}^k\, in\, U_i'$ なる U_i' を受け取っていたとすると、P^k 以外のプロセス $P^{k'}$ は、X_2^k を含む送信を行わないから、Q^k は $\bar{X_2}'^k$ を含む U_j を返すことができない (図 6.4)。また、Q^k 以外の $Q^{k''}$ は、$\bar{X_1}^k\, in\, U_i$ なる U_i を X_1^k が自分の持っているセッション固有値 $X_1^{k''}$ と異なるので、正常な相手プロセス以外からの通信であることがわかるから受けとらない (図 6.5)。従って、何かを受け取るとすれば U_i とは異なる $X_1^{k''}\, in\, U_i''$ なる U_i'' を受け取っている。このとき、$\bar{X_2}^{k''}\, in\, U_i''$、$X_2^{k''} \neq X_2^k$ (X_2^k を U_i から取り出して U_i'' に入れることはできない) より、$Q^{k''}$ も P^k に $\bar{X_2}^k\, in\, U_j$ なる U_j を送り返すことができない。従って、U_j が返ってきたということは、U_i が Q^k に受け取られたということを表す (図 6.5)。このとき、Q^k, P^k は

ともに X_1^k と X_2^k をもつ。

また、状態 $S_{Q_i}^{k'''}$ 以前で、イントルーダの介入により X_2^k を持つに至った $Q^{k'''}(\neq Q^k)$ が存在するとしても、X_2^k と $X_1^{k'''}$ を含む U_i''' を送信するプロセスが存在しないため、処理を状態 $S_{Q_i}^{k'''}$ 以降に進めることができない (図 6.6)。

さらに、状態 $S_{P_j}^{k''''}$ 以前で、同様にして X_1^k を持つ $P^{k''''}(\neq P^k)$ は処理を状態 $S_{P_j}^{k''''}$ 以降に進めることはできない (図 6.7)。なぜなら、先ほどと同様の理由により、$P^{k''''}$ の送信した U_i'''' (X_1^k および $X_2^{k''''}$ を含む) を Q^k が受信して返信しない限り、$X_2^{k''''}$ を含む返信はあり得ず、$P^{k''''}$ は状態 $S_{P_j}^{k''''}$ に進むことができないからである。もし、Q^k が受信した場合には処理が継続できるが、この場合は $P^{k''''}$ がすなわち P^k の役割を果たしていることになる。

従って、状態 $S_{P_j}^k$ 以降で、X_1^k, X_2^k の両方を持つプロセスは P^k, Q^k のみであり、その他のプロセスは X_1^k, X_2^k のいずれも持たない。**証明終り**

$guard$ を用いて、Q がメッセージ X_1 と X_2 を渡される場合の関係 $\sim_Q$ を下記のように定義する。

$$Q \models auf\ X_1 \wedge Q \models auf\ X_2 \wedge X_1 \in \mathbf{Session} \wedge X_2 \in \mathbf{Session} \wedge guard_Q\ \bar{X_1}\bar{X_2}$$
$$\frac{}{X_1 \sim_Q X_2}$$

関係 $\sim_Q$ に関して対称律が成り立つ。さらに、この関係 $\sim_Q$ に対し下記の推移律を定義する。

$$\frac{(X_1 \sim_Q X_2) \wedge (X_2 \sim_Q X_3)}{X_1 \sim_Q X_3}$$

このような関係を用いて、さらに以下の性質を推論することが出来る。

- すべての q の初期値でないセッション固有のメッセージ x は関係 $\sim_q$ を用いてグループ分けしたとき、同じグループに属する。

$$\forall q \in \mathbf{Prin}[\forall x_1, x_2 \in (\mathbf{Session} - Init(q)) \\ [x_1 \sim_q x_2]] \tag{6.1}$$

  ここで、変数のドメインが有限集合 $\{t_1, \ldots, t_m\}$ であるような全象束縛と存在束縛に関して、下記の推論規則を導入する。

$$\frac{\varphi[t_1/y] \wedge \ldots \wedge \varphi[t_m/y]}{\forall y \in \{t_1, \ldots, t_m\}[\varphi]}$$
$$\frac{\varphi[t_i/y]}{\exists y \in \{t_1, \ldots, t_m\}[\varphi]}$$

  ただし、φ は論理式、$\varphi[t/y]$ は論理式 φ の中の変数 y の自由出現を t で置き換えることを表す。

- セッションが一組のプロセスだけで成立していることを保証するため、すくなくとも一つの $onetoonecon$ が推論される必要がある。

$$\begin{aligned} &\exists \bar{x_1}, \bar{x_2}, \bar{x_2}' \in \mathbf{TagAM} \\ &[onetoonecon\ \bar{x_1}\bar{x_2}\bar{x_2}'] \end{aligned} \tag{6.2}$$

これらを用いて、$Q \models auf\ X$ を満たすメッセージが、同一部分のすり替えもない条件は、少なくとも一つの $X_1 \in \mathbf{Session}$ かつ P の初期値である X_1 と同一のガードされたメッセージで相手の参加者から送信され、同一部分とのすり替えのないことを表わす下記により定義される。

$$\frac{\begin{aligned}(Q \models auf\ X_1) \wedge (X_1 \in \mathbf{Session}) \wedge \\ (guard_Q\ \bar{X}_1\bar{X}) \wedge \\ Q \models auf\ X \wedge \text{式}\,(6.1) \wedge \text{式}\,(6.2)\end{aligned}}{Q \models unforged\, X}$$

伝達すべきメッセージの一貫性保持

伝達すべきメッセージ (**Share**) を一貫性を保って伝えている (初期値として持たない側の参加者で、$unforged$ となる。) かの検証は、以下の式により行われる。

$$\begin{aligned} &\forall x \in \mathbf{Share}[\exists p, q \in \mathbf{Prin}[x \in Init(p) \wedge \\ &q \models unforged\, x \wedge q \in (\mathbf{Prin}\ -\ \{p\})]] \end{aligned} \tag{6.3}$$

6.4.4 正常終了に関する参加者の判断の一致

セッションの最後の 2 回の送受信を、

$$\begin{aligned} n-1 : p \rightarrow q : U_{n-1} \\ n : q \rightarrow p : U_n \end{aligned}$$

としたときに、参加者 p, q の判断が一致するための条件は、U_{n-1} と U_n にイントルーダが偽造できない情報が入っていることであり、その条件は下記である。

$$\begin{aligned} &\exists p, q \in \mathbf{Prin}[\exists \bar{x_1}, \bar{x_2} \in \mathbf{TagAM}[\\ &\quad x_1 \in \mathbf{Session} \wedge x_2 \in \mathbf{Session} \\ &\quad \wedge q \models p\ says\ \bar{x_1}\ to\ q \wedge \bar{x_1}\ in\ U_{n-1} \\ &\quad \wedge p \models q\ says\ \bar{x_2}\ to\ p \wedge \bar{x_2}\ in\ U_n]] \end{aligned} \tag{6.4}$$

上記の条件 (式 (6.3)、式 (6.4)) をすべて検証することにより、セキュアなプロトコルがその目的を達成しているかどうかを検証することが出来る。

6.5 章のまとめ

セキュリティプロトコルの検証方法として、従来考慮されていなかった並行するプロセス間でのプロトコルの同一部分同士のすり替え、および正常終了判断の一致について、下記を示した。

1. 暗号や電子署名によりイントルーダが更新できない (ガードされている) メッセージによりプロセスの固有値を伝達することで、プロセス A^1, B^1 間でのみこれらのプロセス固有値を持つ条件として、プロセス A^1 からプロセス B^1 へ A^1 と B^1 のプロセス固有値を同時に送り、そのあと A^1 のプロセス固有値を返送すれば十分であることを証明し、それを推論規則として与えた。
2. プロセス間で正常終了の判断を一致させる方策として最後の 2 個のメッセージに前述のガードされたメッセージが挿入されていれば良いことを示した。

時間に関してシリアルなセッションを利用した攻撃に加えて、並行したセッションを利用した攻撃、とくに同一部分のすり替えの可能性ないことの検証方法を提案した。このことはモデルチェックによっても検証が可能であるが、本方式によれば全ての状態を作り出すことなく、より少ない計算により検証が可能である。

第7章　例題への適用

7.1　適用例

本論文の検証方式を下記のITU X.509プロトコル[6]に適用して、そのセキュリティプロトコルとしての目的達成を検証する。本例のプロトコルはメッセージのセキュリティを保った伝送に関する基本手順を定める標準である。本標準のドラフトに対して文献[2]で攻撃の可能性が示されているが、ここに示す標準[6]では、対策済みである。

$$
\begin{aligned}
1 : A \rightarrow B : \quad & A^{11}, \{N_{1A}^{12}, N_{2A}^{13}, \\
& B^{14}, M_{1A}^{15}, \{M_{2A}^{16}\}_{K_B^{17}}\}_{K_A^{-118}} \\
2 : B \rightarrow A : \quad & B^{21}, \{N_{3B}^{22}, N_{4B}^{23}, A^{24}, N_{2A}^{25}, \\
& M_{3B}^{26}, \{M_{4B}^{27}\}_{K_A^{28}}\}_{K_B^{-129}} \\
3 : A \rightarrow B : \quad & A^{31}, \{N_{4B}^{32}, B^{33}\}_{K_A^{-134}}
\end{aligned}
$$

その他の初期条件は下記の通りである。

$$
\begin{aligned}
Init(A) &= \{N_{jA}, M_{jA}, A, B, K_A, K_B, K_A^{-1}\} \\
Init(B) &= \{N_{j'B}, M_{j'B}, A, B, K_A, K_B, K_B^{-1}\} \\
\mathbf{Com} &= \{M_{jA}, M_{j'B}, A, B, K_A, K_B, K_A^{-1}, K_B^{-1}\} \\
\mathbf{Session} &= \{N_{jA}, N_{j'B}\} \\
\mathbf{Share} &= \{M_{jA}, M_{j'B}\} \\
\mathbf{Prin} &= \{A, B\} \\
\mathbf{TagAM} &= \{N_{1A}^0, N_{1A}^{12}, \ldots, K_B^{-10}, K_B^{-129}\} \\
Type(N_{jA}) &= Type(N_{j'B}) = Non \\
Type(M_{jA}) &= Type(M_{j'B}) = Msg \\
Type(A) &= Type(B) = ID \\
Type(K_A) &= Type(K_B) = Key \\
Type(K_A^{-1}) &= Type(K_B^{-1}) = Key
\end{aligned}
$$

ここで、$j = 1\ or\ 2$ かつ $j' = 3\ or\ 4$ である。

この前提条件のもとで論理による検証を行う。まず、すり替えの可能性の検証であるが、U_1, U_2, U_3 はいずれも送信者の秘密キーにより電子署名されており、かつ受信者を識別する情報が電子署名内部に含まれており、異なる参加者ペアのセッションとのすり替えの可能性はない。また、この 3 個の送受信は含むメッセージの数が異なっていることから、別の部分と互いにすり替えを行うことは出来ない。

同一参加者ペア間の同一部分とのすり替えに付いて検討する。A と B の両方のセッション固有値を含む送受信は 2 番目の U_2 のみである。U_2 内のメッセージに関しては、例えば $guard_G\, N_{4B}^{23} N_{2A}^{25}$ である。これに対して、N_{4B}^{32} が 3 番目のメッセージ U_3 に含まれるから、$onetoonecon\, N_{2A}^{25} N_{4B}^{23} N_{4B}^{32}$ が推論される。従って、式 (6.2) が推論できる。また、$N_{1A} \sim_B N_{2A} \sim_B M_{1A} \sim_B M_{2A}$ および $N_{3B} \sim_A N_{4B} \sim_A M_{3B} \sim_A M_{4B}$ より、式 (6.1) が推論できる。さらに、$P \models aufX$ が成立するのは、$N_{1A}, N_{2A}, N_{3B}, N_{4B}, M_{1A}, M_{2A}, M_{3B}, M_{4B}$ である。これらはそのまま $unforged$ となる。

ここで、以下の二つの性質を検証する。

1. 一貫性を保ったメッセージの伝達: **Share** の要素はすべて、初期値として持っていない方の参加者で $unforged$ となっているので、式 (6.3) が推論できる。
2. 終了に関する参加者の判断の一致: 2 番目および 3 番目 (最後) の送受信において、ガードされた伝達が行われており、各々有効な **Session** の要素を含むことから、式 (6.4) が推論できる。

これらにより、相手に伝達すべきメッセージ **Share** が伝達相手にすり替えられること無く伝達されている (unforged となっている) ことがわかる。

7.2 従来の研究との比較

本プロトコルの検証を従来の BAN 論理あるいは SG 論理の方式で行った場合、**Share** の要素のうち、すり替えの可能性がなく伝達されたと検証されるのは、M_{3B} のみである。M_{1A} および M_{2A} は、第一番目の送受信で伝達されているが、この場合 $fresh$ あるいは in_time と推論出来るメッセージが先行して伝達されていないため、これらを当該セッションのメッセージと推論することができない。また、M_{4B} は、A の公開キー K_A により暗号化されて伝達されているため、この内容がすり替えられていないことを検証することができない。従って、この例では、本論文の方式と比較して 1/4 の個数のメッセージしかすり替えの可能性のない伝達を検証することができない。

第8章　まとめ

8.1　すり替えの可能性のないことの検証の方法

本論文では、以下の 3 通りの攻撃に対して、各々のアイデアによるセキュリティプロトコルのイントルーダによるすり替え攻撃の可能性のないプロトコルであることの検証策を示した。

8.1.1　送信者の状態推定によるメッセージの認証の検証

公開キーまたは受信者の秘密キーにより暗号化されたメッセージが信頼できる方法、すなわち、共通キーや送信者の秘密キーにより暗号化された型式で、転送されるようなプロトコルのカテゴリに関して研究した。

これらのメッセージを認証するために以下のアイデアを提案した。

1. 送信者の内部でメッセージを組み立てる場合の同一のサブメッセージのオカレンス間の優先順位を仮定する。
2. セキュリティプロトコルの解析において、メッセージが送信される直前の状態での送信者の保持するメッセージのオカレンスの集合を推論する。

8.1.2　メッセージすり替えの可能性のないことの検証

メッセージの確実な伝達に関して、従来チェックが不十分だった並行セッションとのメッセージすり替えの可能性を指摘し、すり替え攻撃の可能性のないこと、すなわちこのプロトコルでは、すり替えが行われれば必ず参加者が検知可能であることを検証する方法を推論規則として定式化した。

本論文で述べた検証方式の効果を、従来 BAN 論理や SG 論理では攻撃の可能性がないと検証されてしまう、並行セッションの情報を利用した型攻撃について、本論文の方式では誤った結論を導かないことを例で示した。

8.1.3 別セッションの同一部分間のすり替えの可能性のないことの検証

セキュリティプロトコルの検証方法として、従来考慮されていなかった並行するプロセス間でのプロトコルの同一部分同士のすり替え、および正常終了判断の一致について、下記を示した。

1. 暗号や電子署名によりイントルーダが更新できない (ガードされている) メッセージによりプロセスの固有値を伝達することで、プロセス A^1, B^1 間でのみこれらのプロセス固有値を持つ条件として、プロセス A^1 からプロセス B^1 へ A^1 と B^1 のプロセス固有値を同時に送り、そのあと A^1 のプロセス固有値を返送すれば十分であることを証明し、それを推論規則として与えた。
2. プロセス間で正常終了の判断を一致させる方策として最後の 2 個のメッセージに前述のガードされたメッセージが挿入されていれば良いことを示した。

8.2 本論文の方法の有効性検証

上記のアイデアの有効性を検証するために、基本的な前提条件、評価項目と導出規則を提案した。BAN 論理の規則を拡張し、各プロトコルにおいてその検証できる範囲を広げた。本研究の方法の推論規則化に当たっては、同一のアトミックメッセージがプロトコル中に何度も出現する複数オカレンスのどれに推論規則を適用するかを、明確に定義するため、これらのオカレンスを識別するタグを各オカレンスに付加する方式を提案した。また、推論規則に参加者とメッセージ変数に対する全象束縛と存在束縛を加えて表現力を高めている。

本論文のこの検証規則をセキュリティプロトコルのいくつかの簡単な例に適用した。本論文の導出規則により、BAN 論理や SG 論理ではできなかった、並行したセッションを利用した攻撃、とくに同一部分のすり替えによる攻撃の可能性のないことの検証ができることを示した。このことはモデルチェックによっても検証が可能であると考えられるが、本方式によれば全ての状態を作り出すことなく、より少ない計算量により検証が可能である。

セキュリティプロトコルの例として ITU 国際標準の X.509 に本論文の検証方式を適用しその効果を実証した。とくに従来の BAN 論理や SG 論理では、検証できなかった、第一の送受信内のメッセージや、受信者の公開キーにより暗号化されたメッセージのすり替え可能性のないことの検証をすることが出来た。

8.3 その他の本研究の特徴

以上の方策とその検証の他、本研究の特徴となるのは、以下の点である。

8.3.1 プロトコルの抽象化の排除

BAN 論理では、プロトコルを検証するために、それを idealization という抽象化をしてから検証をしていた。しかし、この作業は定式化がされていないため、考え方の相違による間違いが入り込みやすく、問題であった。我々の検証方法では、SG 論理と同様に型の比較を行う必要もあって、このようなプロトコルの変形は行わない方式を採用している。これにより、送受信の流れを表すプロトコル定義をほぼそのままで、アトミックメッセージの名称の変更をすることにより、検証することが可能となり、誤りの入り込む可能性が減少した。

ここで、本研究の方式では、BAN 論理ではメッセージ中の論理式により表現していた、メッセージの機能 (共用キー、秘密キーなど) をアトミックメッセージの命名規則により表現している。このため、プロトコル本来のメッセージの名称を変更して、本研究の方式に沿ったものに変更する必要がある。

8.3.2 前提条件と評価項目の定式化

本研究では従来の BAN 論理や SG 論理では、定式化されていなかったセキュリティプロトコルの検証の前提条件、および評価項目を以下の様に定式化し、評価の自動化のための第一の条件を明確に設定した。

- 前提条件
  - 参加者のセッション開始時に保持している初期アトミックメッセージの集合。
  - セッション毎に異なる値を有するアトミックメッセージの集合。
  - アトミックメッセージの型 (物理的な特徴、メッセージ自身を調べると判明する特徴)。
- 評価項目
  - 相手の参加者に伝達すべきアトミックメッセージの集合。

上記のような特徴により、セキュリティプロトコルのイントルーダによる攻撃の可能性検証をだれが行っても同一の結果を得ることのできる、工学的に有効な手順として制定した。

8.4 今後の課題

他のセキュリティーに関する論理と同じように、本論文で提案した論理を用いて正しく情報が伝わることが証明出来れば、考えられている攻撃には耐えられることことが示せたことになる。このことを示すことは容易である。一方、証明出来なければ、考えられている範囲で必ず攻撃の方法が存在することが言えれば、この論理はこの攻撃の範囲で完全と言える。しかし、これについては、本論理体系はまだ改善の余地があるかも知れない。いずれにせよ、考えている攻撃方法自体を形式的に定義し、証明を行う必要があるが、これについては本論文の範囲を越える。

BAN 論理やここで述べた論理など、セキュリティーに関する論理の推論規則は、もう少し基本的な概念で説明され得るであろう。そのような基本の論理を定義出来れば、これらの推論規則は、証明のマクロと考えることができ、またこのような論理は、多くのセキュリティに関する論理体系を比較検討する記述言語となり得る。この方向での研究も、議論をより厳密にするためには重要である。

関連図書

[1] Ray Bird, Inder Gopal, Amir Herzburg, et al. Systematic design of two-party authentication protocols. In *LNCS 576 Advances in Cryptology CRYPTO'91*, pp. 44–61. Splinger, 1991.

[2] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. Research Report 39, DEC SRC, 1989.

[3] C.J. Date. *An Introduction to Database Systems Volume 2*, pp. 20–24. Addison-Wesley, 1983.

[4] Danny Doyle and Andrew C. Yao. On the security of public key protocols. *Trans. on Information Theory*, No. 2, pp. 198–208, 1983.

[5] Sigrid Gürgens. Sg logic - a formal analysis thechnique for authentication protocols. In *LNCS 1361 Security Protocols 5th International Workshop*, pp. 159–176. Springer, 1997.

[6] ITU. *ITU-T Rec. X.509 | ISO/IEC 9594-8 Information technology - Open Systems Interconnection - The Directory: Authentication Framework*, 1997.

[7] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using FDR. In *LNCS 1055, Tools and Algorithms for the Construction and Analysis of Systems: second internaitonal workshop, TACAS'96*, pp. 147–166. Splinger-Verlag, 1996.

[8] W. Marrero, E. Clarke, and S. Jha. Model checking for security protocols. Research report, CMU, 1997.

[9] Jonathan K. Millen. A necessarily parallel attack. In *Workshop on Formal Methods and Security Protocols*. http://www.cs.bell-labs.com/~nch/fmsp99/, 1999.

[10] Yuko Murayama. Using ban logic for the proof of a network address registration protocol. *Trans. IPSJ*, Vol. 37, No. 5, pp. 779–789, 1996.

[11] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communication of ACM*, Vol. 21, No. 12, pp. 993–999, 1978.

[12] Paul Syverson. On key distribution protocols for repeated authentication. *SIGOPS Operating Systems Reviw*, Vol. 27, No. 4, pp. 24–30, 1993.

[13] Paul F. Syverson and Paul C. van Oorschot. On unifying some cryptographic protocol logics. In *IEEE Symposium on Research in Security and Privacy*, pp. 14–28. IEEE, 1994.

[14] 月村賢治, 斎藤孝通, Wu Wen. 認証プロトコルの解析ツール. 日本ソフトウェア科学会第 16 回大会論文集, pp. 241–244. 日本ソフトウェア科学会, 1999.

付 録A　論理式と推論規則の定義

この付録では、本論文で述べたすべての検証法を統合した検証システムの論理式と推論規則の定義を示す。

A.1　論理式

本論文で使用する論理式は以下に定義される基本命題を古典的論理結合子 $(\wedge, \vee)$ で拡張し、さらに参加者とメッセージ変数に関する全象束縛と存在束縛を加えたものである。以下の基本命題式の記述方法は、BAN 論理のものを使用した。最初の 7 個の論理式は BAN 論理におけるものと同じである。但し、タグ付きのメッセージを含む論理式は、特定の出現場所のメッセージに関する論理式であり、これは本論文で導入された。

1. $P \models \varphi(X)$: P は持っているメッセージ X について、論理式 φ を信じる。論理式 φ には、$\models$ は含まれない。
2. $P \lhd \bar{X}$: P は $\bar{X}$ を受け取る。
3. $P \overset{Z}{\leftrightarrow} Q$: Z は P と Q の共用キーである。P, Q 以外は Z を知らない。
4. $\overset{Z}{\mapsto} P$: Z は P の公開キーである。P 以外は $inv(Z)$ を知らない。
5. $P\ has\ \bar{X}$: P は $\bar{X}$ を持っている。
6. $P \mid\sim \bar{X}\ to\ Q$: P はかって Q に $\bar{X}$ と言ったことがあり、Q は $\bar{X}$ を受け取った。[1]
7. $P\ says\ \bar{X}\ to\ Q$: P から Q へセッション内でメッセージ $\bar{X}$ を送受信する。そこで送られるメッセージ $\bar{X}$ は、他のセッションの同一送受信における、同一部分メッセージとのすり替えを除き、その送受信のなかで改変されない。[1] [2]
8. 欠番
9. $\bar{X_1}\ in\ \bar{X_2}$: $\bar{X_1}$ は $\bar{X_2}$ の構成要素。
10. $unforged\ X$: X はアトミックメッセージであって、イントルーダにより改変されていない。

[1] $to\ Q$ は本論文で拡張した部分であり、メッセージ $\bar{X}$ の送り先が Q であることを確認していることを表す。

[2] 同一部分とのすり替えに関する部分は、本論文で拡張した部分である。

11. $P\ canbuild_i\ \bar{X}$: P は i 番目の状態で所有するメッセージから $\bar{X}$ を作成することができる。
12. $P\ usable_i\ \bar{X}$: P は $\bar{X}$ と、i 番目の状態で所有するメッセージから U_i を作成することができる。
13. $Pcbus_i\bar{X}$: P は $\bar{X}$ から $\bar{X_1}$ をとりだすことができる。ここで、$Pcanbuild_i\bar{X_1}$ であり、かつ、$P\ usable_i\ \bar{X_1}$ である。

14. 欠番。
15. $auf\ X$: (almost unforged)X はアトミックメッセージであって、他のセッションの同一送受信における同一部分メッセージとのすり替えを除き、改変されていない。
16. $\bar{X}\ isto\ P$: $\bar{X}$ は宛先 P の情報を含むメッセージである。
17. $replaceunit\bar{X}$: $\bar{X}$ は一つの通信の途中で、イントルーダによりそのメッセージをすり替えの可能性がある。しかし、イントルーダにより、$\bar{X}$ の部分メッセージがその部分のみ、すり替えられることはない。
18. $unreplaceable\ \bar{X}$: 受信した $\bar{X}$ は他のセッションの同一送受信における同一部分メッセージとのすり替えを除き、すり替えられている可能性がない。

19. $guard_Q\ \bar{X_1}\bar{X_2}$: $\bar{X_1}$ と $\bar{X_2}$ は暗号化または電子署名によりイントルーダが更新出来ない同一のメッセージ (ガードされたメッセージ) の一部として送信され、Q に受信される。
20. $onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2'}$: $guard_Q\ \bar{X_1}\bar{X_2}$ が成立するような **Session** の要素 $\bar{X_1}\bar{X_2}$ において、X_1 はメッセージの受信者 Q の初期メッセージ集合の要素であり、X_2 は送信者 $P(\neq Q)$ の初期メッセージ集合の要素である。また、この送受信の後、Q から P へ、ガードされたメッセージの一部として、$\bar{X_2'}$ が返信される。
21. $X_1 \sim_P X_2$: X_1 と X_2 は同一のガードされたメッセージの一部として送信され P に受信される。
22. $X \in \mathbf{Set}$: X が集合 **Set** の要素である。
23. $X \notin \mathbf{Set}$:X が集合 **Set** の要素ではない。

A.2 推論規則

1. メッセージの送信

$$\frac{P \lhd \overline{\{X\}_Z} \wedge P \models Q \overset{Z}{\leftrightarrow} P}{P \models Q \ |\sim\ \bar{X}\ to\ P}$$

$$\frac{P \models \bar{X}\ isto\ P \wedge P \models \overset{inv(Z)}{\mapsto} Q \wedge P \lhd \overline{\{X\}_Z}}{P \models Q\ \mid\sim\ \bar{X}\ to\ P}$$

2. メッセージのすり替え

(a) まず、送受信されるひとかたまりのメッセージの中で、イントルーダが自由に更新可能な部分メッセージ X を $changeable\ \bar{X}$ とする。

$$\frac{changeable\ \overline{X_1, X_2}}{changeable\ \bar{X_1}}$$

$$\frac{changeable\ \overline{\{X\}_{K_{jP}}}}{changeable\ \bar{X}}$$

$$\overline{changeable\ U_i}$$

(b) 次に $changeable\ \bar{X}$ であるが、イントルーダが作り替えることが出来ないものを $replaceunit\ \bar{X}$ という。

$$\frac{changeable\ \overline{\{X\}_{K_{jPQ}}}}{replaceunit\ \overline{\{X\}_{K_{jPQ}}}}$$

$$\frac{changeable\ \overline{\{X\}_{K_{jP}^{-1}}}}{replaceunit\ \overline{\{X\}_{K_{jP}^{-1}}}}$$

(c) $replaceunit\ \bar{X}$ の内で、他の部分とのすり替えが出来ない部分を $unreplaceable\ \bar{X}$ とする。

$$\frac{replaceunit\ \bar{X} \wedge Sametype(\bar{X}^k) = \phi}{unreplaceable\ \bar{X}}$$

$$\frac{unreplaceable\ \bar{X} \wedge \bar{X_1}\ in\ \bar{X}}{unreplaceable\ \bar{X_1}}$$

ここで、関数 $Sametype(\bar{X})$ は下記のように、$\bar{X}^k$ と同じ型のタグ付きメッセージの集合として定義される。

$$\begin{aligned} Sametype(\bar{X}^k) = \{ & \bar{X_1}^{k_x} \mid \\ & \bar{X}\ in\ U_i \wedge Q \lhd U_i \wedge \\ & (Type_Q(X_1) = Type_Q(X)) \wedge \\ & (tag(\bar{X_1}^{k_x}) \neq tag(\bar{X}^k) \wedge \\ & ((k \neq k') \vee \\ & (k = k' \wedge st(tag(\bar{X_1}^{k_x})) \leq st(tag(\bar{X}^k)))\} \end{aligned}$$

3. セッション間のメッセージすり替えチェック

(a) $\bar{x}_{01}^{k}$ と $\bar{x}_{11}^{k_x}$ は異なるメッセージである。

$$(x_{01} \neq x_{11} \vee (k \neq k_x \wedge x_{01} \in \mathbf{Session})) \tag{A.1}$$

ここで、$x_{01} \in \mathbf{Session}$ は、x_{01} がセッション固有値であることを表わす。

(b) Q はアトミックメッセージ $\bar{x}_{01}^{k}$ を見ることができ、x_{01} は Q の初期状態である。または、Q はメッセージ $\{\bar{X_3}^{k}\}_{\bar{x}_{01}^{k}}$ を見ることができ、$inv(x_{01})$ は Q の初期状態である。

$$\begin{aligned}(Q\ has\ \bar{x_{01}} \wedge x_{01} \in Init(Q)) \vee \\ (Q\ has\ \{\bar{X_3}\}_{\bar{x}_{01}} \wedge inv(x_{01}) \in Init(Q))\end{aligned} \tag{A.2}$$

(c) p' は $\bar{x}_{11}^{k_x}$ または $\{\bar{X_2}^{k_x}\}_{\bar{x}_{11}^{k_x}}$ を持っていて送信した。かつ、以下のいずれかが成り立つ。x_{11} は p' の初期状態、あるいは、q' の初期状態であり、p' は q' から同一メッセージとのすり替え以外のない送受信 (says) で x_{11} を受取りかつ、x_{01} と x_{11} とは同一ではない。q' から says で x_{11} をもらった場合、これが同一部分同士ですり替えられている可能性があることから、$x_{01} = x_{11}$ ならすり替えの検出が出来ない可能性がある。また、$k_x \neq k$ の時、U_h から $\bar{x_{11}}^{k_x}$ を取り出した後、U_h を相手に渡さずセッション k_x を中断する可能性があるため、状態 $(h-1)$ までで、says によるメッセージの伝達がなされている必要がある。ここで、$U_h^{k_x}$ は、$\bar{x}_{11}^{k_x}$ を含む送受信の単位である。

$$\begin{aligned}\exists p', q' \in \mathbf{Prin}[((p'\ cbus_h \bar{x}_{11} \vee \\ p'\ cbus_h \{\bar{X_2}^{k_x}\}_{\bar{x}_{11}}) \wedge h = st(tag(\bar{x}_{11}))) \wedge \\ q' \lhd U_h \wedge p' \neq q' \wedge \\ (x_{11} \in Init(p') \vee \\ (p' \models q'\ says\ \bar{x}'_{11}\ to\ p' \wedge \\ x_{11} \in Init(q') \wedge p'\ has\ \bar{x}'_{11} \wedge x_{01} \neq x_{11} \wedge \\ (k_x \neq k \vee st(tag(\bar{x}'_{11})) \leq h-1)))]\end{aligned} \tag{A.3}$$

これらより、$Sametype(\bar{X}^{k}) \neq \phi$ の場合の、イントルーダが同一の型のメッセージ同士ですり替えのできない条件は、

$$\begin{aligned}replaceunit\ \bar{X} \wedge (\forall \bar{x_1}^{k_x} \in Sametype(\bar{X}^{k}) \\ [\exists (\bar{x_{01}}^{k}, \bar{x_{11}}^{k_x}) \in Cores(\bar{X}^{k}, \bar{x_1}^{k_x}) \\ \frac{[\text{式 } (A.1) \wedge \text{式 } (A.2) \wedge \text{式 } (A.3)]])}{unreplaceable\ \bar{X}}\end{aligned} \tag{A.4}$$

である。

4. メッセージの伝達保証

$$\frac{\begin{array}{r}Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X})) \wedge \\ P \models Q \ \mid\sim\ \bar{X}\ to\ P \wedge unreplaceable\ \bar{X}\end{array}}{P \models Q\ says\ \bar{X}\ to\ P}$$

$$\frac{P \models Q\ says\ \bar{X}\ to\ P \wedge X \in Init(Q) \wedge X \notin Init(P)}{P \models auf\ X}$$

$$\frac{\begin{array}{r}P \models Q\ says\ \overline{\{X\}_Z}\ to\ P \wedge \\ P\ has\ \overline{inv(Z)} \wedge Q\ cbus_i \bar{X} \wedge i = st(tag(\bar{X}))\end{array}}{P \models Q\ says\ \bar{X}\ to\ P}$$

結合の分解は下記による。

$$\frac{P \models Q\ says\ (\overline{X_1, X_2})\ to\ P}{P \models Q\ says\ \bar{X_1}\ to\ P}$$

5. メッセージの一貫性を保った伝達

$$\frac{\bar{X_1}\ in\ \bar{X} \wedge \bar{X_2}\ in\ \bar{X} \wedge Q \models P\ says\ \bar{X}\ to\ Q}{guard_Q\ \bar{X_1}\bar{X_2}}$$

$$\frac{\begin{array}{r}guard_Q\ \bar{X_1}\bar{X_2} \wedge \\ X_1 \in \mathbf{Session} \wedge X_2 \in \mathbf{Session} \wedge \\ P \models auf\ X_1 \wedge Q \models auf\ X_2 \wedge \\ st(tag(\bar{X_2})) < st(tag(\bar{X_2}')) \wedge \\ P \models Q\ says\ \bar{X_2}'\ to\ P\end{array}}{onetoonecon\ \bar{X_1}\bar{X_2}\bar{X_2}'}$$

$$\frac{\begin{array}{r}Q \models auf\ X_1 \wedge Q \models auf\ X_2 \wedge X_1 \in \mathbf{Session} \wedge \\ X_2 \in \mathbf{Session} \wedge guard_Q\ \bar{X_1}\bar{X_2}\end{array}}{X_1 \sim_Q X_2}$$

$$\frac{(X_1 \sim_Q X_2) \wedge (X_2 \sim_Q X_3)}{X_1 \sim_Q X_3}$$

$$\begin{array}{r}\forall q \in \mathbf{Prin}[\forall x_1, x_2 \in (\mathbf{Session} - Init(q)) \\ [x_1 \sim_q x_2]]\end{array} \qquad \text{(A.5)}$$

$$\exists \bar{x_1}, \bar{x_2}, \bar{x_2}' \in \mathbf{TagAM}$$
$$[onetoonecon\ \bar{x_1}\bar{x_2}\bar{x_2}'] \tag{A.6}$$

$$\frac{\begin{array}{r}(Q \models auf\ X_1) \wedge (X_1 \in \mathbf{Session}) \wedge \\ (guard_Q\ \bar{X_1}\bar{X}) \wedge \\ Q \models auf\ X \wedge 式\,(A.5) \wedge 式\,(A.6)\end{array}}{Q \models unforged\ X}$$

6. 存在および全象束縛の変換変数のドメインが有限集合 $\{t_1, \ldots, t_m\}$ であるような全象束縛と存在束縛に関して、下記の推論規則を導入する。

$$\frac{\varphi[t_1/y] \wedge \ldots \wedge \varphi[t_m/y]}{\forall y \in \{t_1, \ldots, t_m\}[\varphi]}$$
$$\frac{\varphi[t_i/y]}{\exists y \in \{t_1, \ldots, t_m\}[\varphi]}$$

   ただし、φ は論理式、$\varphi[t/y]$ は論理式 φ の中の変数 y の自由出現を t で置き換えることを表す。
7. 伝達すべきメッセージの一貫性保持

$$\forall x \in \mathbf{Share}[\exists p, q \in \mathbf{Prin}[x \in Init(p) \wedge$$
$$q \models unforged\ x \wedge q \in (\mathbf{Prin}\ -\ \{p\})]] \tag{A.7}$$

8. 正常終了に関する参加者の判断の一致セッションの最後の 2 回の送受信を、

$$n-1 : p \rightarrow q : U_{n-1}$$
$$n : q \rightarrow p : U_n$$

   としたときに、参加者 p, q の判断が一致するための条件は下記である。

$$\exists p, q \in \mathbf{Prin}[\exists \bar{x_1}, \bar{x_2} \in \mathbf{TagAM}[$$
$$x_1 \in \mathbf{Session} \wedge x_2 \in \mathbf{Session}$$
$$\wedge q \models p\ says\ \bar{x_1}\ to\ q \wedge \bar{x_1}\ in\ U_{n-1}$$
$$\wedge p \models q\ says\ \bar{x_2}\ to\ p \wedge \bar{x_2}\ in\ U_n]] \tag{A.8}$$

9. 受信

$$\frac{\begin{array}{r}P \lhd \overline{X_1, X_2} \\ \hline P \lhd \bar{X_1}\end{array}}{\begin{array}{r}P \lhd \overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)} \\ \hline P \lhd \bar{X}\end{array}}$$

10. 含む

$$\frac{\bar{X}\ in\ \bar{X}_1}{\bar{X}\ in\ \overline{(X_1, X_2)}}$$

$$\frac{\bar{X}\ in\ \bar{X}_1}{\bar{X}\ in\ \overline{\{X_1\}_Z}}$$

$$\frac{}{\bar{X}\ in\ \bar{X}}$$

11. 名前を含む

$$\frac{P \lhd P^{w'} \wedge P^{w'}\ in\ \bar{X}}{P \models \bar{X}\ isto\ P}$$

$$\frac{P \lhd \{\bar{X}_1\}_{K_P^{w'}} \wedge \{\bar{X}_1\}_{K_P^{w'}}\ in\ \bar{X}}{P \models \bar{X}\ isto\ P}$$

$$\frac{P \lhd K_{PQ}^{w'} \wedge K_{PQ}^{w'}\ in\ \bar{X}}{P \models \bar{X}\ isto\ P}$$

$$\frac{P \lhd \{\bar{X}_1\}_{K_{PQ}^{w''}} \wedge \{\bar{X}_1\}_{K_{PQ}^{w''}}\ in\ \bar{X}}{P \models \bar{X}\ isto\ P}$$

12. 持っている

$$\frac{P\ has\ \overline{X_1, X_2}}{P\ has\ \bar{X}_1}$$

$$\frac{P\ has\ \overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)}}{P\ has\ \bar{X}}$$

$$\frac{P \lhd \bar{X}}{P\ has\ \bar{X}}$$

13. 送信者はメッセージを作れる
論理式 $P\ canbuild_i\ \bar{X}$ は、下記の規則による。

- もし、P が i 番目の状態以前で X を持つなら、P can build $\bar{X}$ である。

$$\frac{P\ has\ \bar{X}' \wedge st(tag(\bar{X}')) \leq (i-1)}{P\ canbuild_i\ \bar{X}}$$

- もし、P が $\bar{X}_1$ および $\bar{X}_2$ を作れるなら、P はその結合も作れる。

$$\frac{P\ canbuild_i\ \bar{X}_1 \wedge P\ canbuild_i\ \bar{X}_2}{P\ canbuild_i\ \overline{X_1, X_2}}$$

- もし、P が $\bar{X}$ を作れ、かつ P が i 番目の状態以前で暗号化キー Z を持つなら、P は $\overline{\{X\}_Z}$ を作ることが出来る。

$$\frac{P\ canbuild_i\ \bar{X} \wedge P\ has\ \bar{Z}' \wedge st(tag(\bar{Z}')) \leq (i-1)}{P\ canbuild_i\ \overline{\{X\}_Z}}$$

14. 送信者は i 番目の送信メッセージ U_i を作るのに X を使える。
論理式 $P\ usable_i\ \bar{X}$ は以下の規則による。

- U_i それ自身が送信メッセージと見なせる。

$$\overline{P\ usable_i\ U_i}$$

- もし、結合 $\overline{X_1, X_2}$ がメッセージ U_i を作るのに用いられるなら、そして、P が $\bar{X_2}$ または $\bar{X_1}$ を作ることが出来るなら、$\bar{X_1}$ または $\bar{X_2}$ をそれぞれ送信メッセージ U_i に使用できる。

$$\frac{P\ usable_i\ \overline{X_1, X_2} \wedge P\ canbuild_i\ \bar{X_2}}{P\ usable_i\ \bar{X_1}}$$

$$\frac{P\ usable_i\ \overline{X_1, X_2} \wedge P\ canbuild_i\ \bar{X_1}}{P\ usable_i\ \bar{X_2}}$$

- もし $\overline{\{X\}_Z}$ が送信メッセージ U_i に使用でき、P が i 番目の状態の前に Z を持っているなら $\bar{X}$ を送信メッセージに使用できる。

$$\frac{P\ usable_i\ \overline{\{X\}_Z} \wedge Phave\bar{Z}' \wedge st(tag(\bar{Z}')) \leq (i-1)}{P\ usable_i\ \bar{X}}$$

15. 送信者は $\bar{X}$ を知っていて、転送 U_i の一部分として $\bar{X}$ を使用できる。
論理式 $P\ cbus_i\bar{X}$ は下記の規則による。

- もし、P が $\bar{X}$ を作れ、$\bar{X}$ が送信メッセージを作成するのに使用できれば、P は $\bar{X}$ を作成でき、使用可能である。

$$\frac{P\ canbuild_i\ \bar{X} \wedge P\ usable_i\ \bar{X}}{P\ cbus_i\bar{X}}$$

- もし P が $\overline{X_1, X_2}$ を作成可能で使用可能なら、P は $\bar{X_1}$ を作成可能で使用可能である。

$$\frac{P\ cbus_i\overline{X_1, X_2}}{P\ cbus_i\bar{X_1}}$$

- もし、P が $\{X\}_Z$ を作成可能で使用可能であり、P が $inv(Z)$ を持っている (すなわち P が X を知っている) なら、P はまた X を作成可能で使用可能である。

$$\frac{P\ cbus_i\overline{\{X\}_Z} \wedge P\ has\ \overline{inv(Z)} \wedge st(tag(\overline{inv(z)})) \leq (i-1)}{P\ cbus_i\bar{X}}$$

付 録B　成果物

B.1　論文

1. 根岸和義、米崎直樹、セキュリティプロトコルの一貫性および正常終了一致の同一参加者による複数セッションを考慮した検証法、情報処理学会論文誌、41 巻 8 号、pp.2281-2290、2000 年 8 月。
2. 根岸和義、米崎直樹、並行セッションの情報を用いる攻撃を考慮したプロトコルセキュリティの検証、日本ソフトウェア科学会 (投稿準備中)
3. Kazuyoshi Negishi and Naoki Yonezaki. Verification method for possiblity of parallel attack on multiple sessions with the same principals in a security protocol. In *Informal proceeding of Formal Method on Computer Security 2000*, pp.109-120, July 2000.

B.2　研究会

1. 根岸和義、米崎直樹、セキュリティプロトコルの一貫性および正常終了一致の検証法、情報処理学会コンピュータセキュリティ研究会、情処研報 2000-CSEC-8、Vol.2000,No.30、pp.37-42、2000 年 3 月
2. 根岸和義、米崎直樹、セキュリティプロトコルにおける暗号化メッセージの送信者による認知に関する検証法、情報処理学会コンピュータセキュリティ研究会、情処研報 2000-CSEC-11、Vol.2000,No.80、pp.25-30、2000 年 9 月
3. 根岸和義、米崎直樹、並行セッションの情報を用いる攻撃を考慮したプロトコルセキュリティの検証、日本ソフトウェア科学会第 17 回大会予稿集、2000 年 9 月 (発表予定)