マルチメディア通信と分散処理 97－7
コンピュータセキュリティ 8－7
（2000. 3. 21）

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

根岸 和義[†‡] 米崎 直樹[‡]

[†] 日立製作所 ビジネスソリューション開発本部
[‡] 東京工業大学大学院 情報理工学研究科計算工学専攻

従来のセキュリティプロトコルの検証では，(1) すり替えられることなくメッセージを伝達する，(2) 秘密を保持する，ことの 2 点を検証することが一般的である．ただし，この中で考慮される攻撃のパターンでは，同一参加者の複数セッション間におけるプロトコルの，同一部分メッセージ同士のすり替えが考慮されていなかった．本論文ではこのような攻撃を考慮した一貫性の検証方法を与え，さらに，そこで用いられる推論規則の健全性を証明した．一方，電子商取引において，第三者の攻撃があっても，プロトコルを実行する 2 つのプロセスが，同期して正常終了する必要がある．そこで，セッションを構成するプロセスの間で正常終了の判断が一致するようなプロトコルになっているか否かを検証する方法を提案し，適用例を示した．

Verification method for consistency and normal termination agreement on multiple sessions with the same principals in a security protocol

Kazuyoshi Negishi[‖§] Naoki Yonezaki[§]

[‖]Business Solution Systems Development Division, Hitachi Ltd.
[§]Department of Computer Science, Graduate School of Information Science and Engeneering, Tokyo Institute of Technology

Generally verification of security protocols has focused on the subjects, (1)transmission of the messages without replacement, and (2)secret maintenance of messages. The pattern of attack considered in these researches does not take the following situation into account, that is the case where the replacement of messages might be done between the different sessions of a protocol. We give verification rules for consistency that take account of these attacks. We also verify the soundness of the verification rule. In electronic commerce trading, protocol must be executed between two processes and terminate normally under attacks by the malicious principals. We propose the method to verify the agreement of the decision of normal termination for processes which consist the session. Finally, we show the effectiveness of our methods by applying it to the example of ITU X.509 protocol.

1 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，通信のセキュリティを守ることによる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．

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 [1]，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 [3] 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．[4] 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 [5, 6, 7]，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 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．

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

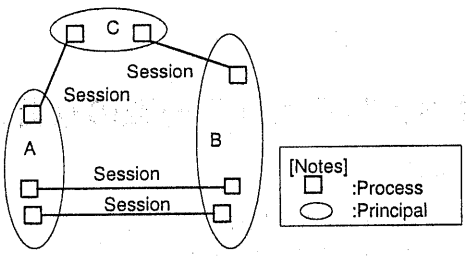

図 1: 参加者，プロセスとセッション

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．

本論文の第 2 章では，制約条件，用語定義を述べる．第 3 章では，プロトコルのモデルとその初期条件を示し，第 4 章ではプロトコルの目的を定義し，第 5 章では論理式による検証の手順を示すとともにその正しさを証明する．第 6 章では適用例を示す．

2 前提条件と用語

2.1 前提条件

プロトコル
- 扱うプロトコルは 1 種類のみとする．
- 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．
- 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．

セッション
- セッションはプロトコルの一連の実行である．
- セッションは並行して複数同時実行可能である．
- セッション間でメッセージの受け渡しはない．
- 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
- 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．
- 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
- 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．
- 暗号化メッセージは完全で，キーなしでは解読できない．
- 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．
- 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
- 対象とするセッション全体の開始時点では，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．
- 送受信されたメッセージは全て傍受出来るとする．
- 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)．
- 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．)

2.2 用語

2.2.1 参加者，役割，プロセス およびその状態

A, B, C: 特定の参加者を表す定数．

p, q 参加者を表す変数．

P, Q, R, T, V: 参加者を表すメタ変数．

P^k: 参加者 P が実行するセッション k のプロセス．

$S^k_{P,i}$: 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．

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．

2.2.2 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，以下のいずれかである．

   $M, M_1, M_2, \ldots$: 単なるデータ．

   $N, N_1, N_2, \ldots$: 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)．

   $K, K_1, K_2, \ldots$: キー．

   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メッセージ．

   N_{jP}: 同上の nonce.

   K_{jPQ}; 参加者 P と Q の共用キー．

   K_{jP}: 参加者 P の公開キー．

   K^{-1}_{jP}: 参加者 P の秘密キー．

   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．

2.2.3 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．(. は $K_{1p}^{-1 \cdot w}$ のように秘密キーの -1 と区別するためのものである) ここで，タグ w の値を以下のように定義する．

$w = iu$: i 番目の送受信の 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で，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，メタ変数 X_j^w で表す．また，タグ付きのキーを表すメタ変数を Z_j^w 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 $x^{\cdot w}$, $x_1^{w_1}$, $x_2^{w_2}$ とする．タグ付きの変数 (例えば X_1^w) に対して，対応するタグなしの変数 X_1 の値が，前者の全てのタグを取り去ったものとして定義される．また，対応するタグ w の値は，前者の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．例えば，$X_1^w = \{M_{1A}^{12}, N_{2B}^{13}\}_{K_{3AB}^{14}}$ とすると，これに対応して，$X_1 = \{M_{1A}, N_{2B}\}_{K_{3AB}}$，および，$w = 121314$ である．タグから i 番目の送受信，あるいは初期値であることを求める関数 $st(w)$ を下記のように定義する．

- $w = i_1 u_1 i_2 u_2 \ldots i_m u_m$ の場合 $st(w) = i$
- $w = 0$ の場合 $st(w) = 0$

i 番目に送受信されるタグ付きメッセージの全体を変数 U_i で表す．

2.2.4 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．$Type_P(X^{\cdot w})$ 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タグ付きメッセージ $X^{\cdot w}$ の型である．ただし，$X^{\cdot w}$ 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 □ となる．

$$Type_P(X^{\cdot w}, X'^{\cdot w'}) = Type_P(X^{\cdot w}), Type_P(X'^{\cdot w'})$$

$$Type_P(\{X^{\cdot w}\}_{Z'^{\cdot w'}}) = \begin{cases} \{Type_P(X^{\cdot w})\}_{Type(Z'^{\cdot w'})} & \cdots P \text{ has } Z''^{\cdot w''} \text{が導ける時} \\ \square & \cdots P \text{ has } Z''^{\cdot w''} \text{が導けない時} \end{cases}$$

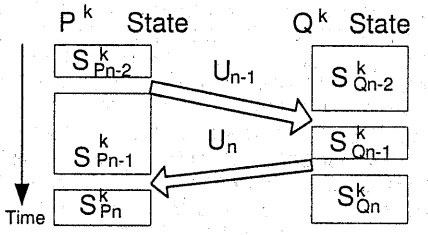

図 2: プロセスの最終状態遷移

$$Type_P(X^{\cdot w}) = Type(X) \quad \cdots X \in \mathbf{Atom}$$

ここで，$Z'' = inv(Z')$ である．$P\ has\ Z''^{\cdot w''}$ は後述の論理式であり，P が $Z''^{\cdot w''}$ を持っていることをあらわす．

3 プロトコルのモデルと初期条件

3.1 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_{Pi-1}^k から S_{Pi}^k に遷移し，Q^k の状態は U_i の受信時に S_{Qi-1}^k から S_{Qi}^k 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_{P0}^k である．

3.2 メッセージ集合と役割

P の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を $Init(P)$ とする．

セッション毎に異なる値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．

4 セキュリティプロトコルの目的

セキュリティプロトコルの目的は以下の通りである．

4.1 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 $X \in$ **Share** につき，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．

4.2 セッションの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 2 のように

状態が遷移する．U_{n-1} の伝達が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 [8]．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．

5 論理による検証

前節の二つの目的を達成していることを検証するために，以下の論理的言語を用いる．

5.1 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．

$P \models Y(X)$: P は持っているメッセージ X について，論理式 Y を信じる．

$P \lhd X^{\cdot w}$: P は $X^{\cdot w}$ を受け取った．

$P \overset{Z}{\leftrightarrow} Q$: Z は P と Q の共用キーである．

$\overset{Z}{\mapsto} P$: Z は P の公開キーである．

$P\ has\ X^{\cdot w}$: P は $X^{\cdot w}$ を持っている．

$P \mid\sim X^{\cdot w}\ to\ Q$: P はかって Q に $X^{\cdot w}$ と言ったことがあり，Q は $X^{\cdot w}$ を受け取った．[1]

$P\ says\ X^{\cdot w}\ to\ Q$: P から Q へセッション内でメッセージ $X^{\cdot w}$ を送受信する．そこで送られるメッセージ $X^{\cdot w}$ 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．

$X^{\cdot w}\ in\ X'^{\cdot w'}$: $X^{\cdot w}$ は $X'^{\cdot w'}$ の構成要素．

$unforged\ X$; 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．

$auf\ X$: (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．

$X^{\cdot w}\ isto\ P$: $X^{\cdot w}$ は P あてのメッセージである．

$unreplaceable\ X^{\cdot w}$: 受信した $X^{\cdot w}$ 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．

$guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}$: $X_1^{\cdot w_1}$ と $X_2^{\cdot w_2}$ は同一の暗号または電子署名によりイントルーダが更新出来ないメッセージ (ガードされたメッセージ) として送信され，Q に受信される．

$onetoonecon\ X_1^{\cdot w_1} X_2^{\cdot w_2} X_2^{\cdot w_2'}$: $guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}$ が成立するような $X_1^{\cdot w_1} X_2^{\cdot w_2}$ において，$X_1^{\cdot w_1}$ はガードされた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，$X_2^{\cdot w_2}$ 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，同様にガードされたメッセージの一部として，$X_2^{\cdot w_2'}$ が返信される．

$X \sim_P X'$: X と X' はセッション固有値 (いずれも，初期状態では送信者が保持) であり，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．

$X \in \mathbf{Set}$: X が集合 Set の要素である

$X \notin \mathbf{Set}$: X が集合 Set の要素ではない

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

5.2 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，$P\ has\ X^{\cdot 0}$ かつ $P \models unforged\ X$ 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，$P \models P \overset{K_{jPQ}}{\leftrightarrow} Q$ 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．

5.3 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のうち，5.3.3 までは，SG 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す．

5.3.1 メッセージを送信したことがある

共用キーで暗号化したメッセージを受信した場合，以下の推論により，異なる参加者ペア間のメッセージからの再使用はなく，送信元がメッセージを送信としたことがあるという論理式 ($\mid\sim\ to$) が定義される．

$$\frac{P \lhd \{X^{\cdot w}\}_{Z^{\cdot w'}} \wedge P \models Q \overset{Z}{\leftrightarrow} P}{P \models Q \mid\sim X^{\cdot w}\ to\ P}$$

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 $X^{\cdot w}$ があり，$X^{\cdot w}$ 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．

$$\frac{P \models X^{\cdot w}\ isto\ P \wedge Q \neq P \wedge \quad P \models \overset{Z}{\mapsto} Q \wedge P \lhd \{X^{\cdot w}\}_{Z'^{\cdot w'}} \wedge Z = inv(Z')}{P \models Q \mid\sim X^{\cdot w}\ to\ P}$$

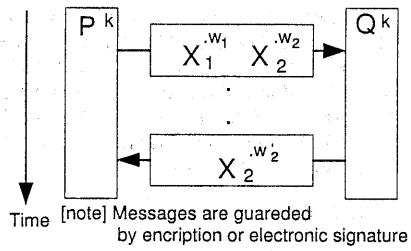

図 3: 1組のプロセスによるセッション固有値の共有

5.3.2 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 $X^{\cdot w}$ 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，$X^{\cdot w}$ の受信者を P とした時，同一の型 $Type_P(X^{\cdot w})$ 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出来ない．メッセージ $X^{\cdot w}$ についてこのような部分が存在しないことを *unreplaceable* $X^{\cdot w}$ で表す．

5.3.3 すり替えなく伝達

他のセッションの同一部分とのすり替えを除き，メッセージがすり替えられず，確実に相手の参加者に伝達される条件を表す推論規則が定義される．

$$\frac{P \models Q \vdash X^{\cdot w}\ to\ P \wedge unreplaceable\ X^{\cdot w}}{P \models Q\ says\ X^{\cdot w}\ to\ P}$$

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

5.3.4 メッセージの一貫性を保った伝達

同一参加者の他のプロセスの送信した同一部分のメッセージとのすり替えに関する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 $guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}$ を下記の推論規則の結果とする．

$$\frac{\begin{array}{r} R,T \in \mathbf{Prin} \wedge R \models auf\ X_1 \wedge T \models auf\ X_2 \wedge \\ Q \models P \vdash X^{\cdot w}\ to\ Q \wedge \\ X_1^{\cdot w_1}\ in\ X^{\cdot w} \wedge Q \models P\ says\ X_1^{\cdot w_1}\ to\ Q \\ X_2^{\cdot w_2}\ in\ X^{\cdot w} \wedge Q \models P\ says\ X_2^{\cdot w_2}\ to\ Q \end{array}}{guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}}$$

ここで，R,T は P,Q とは独立して参加者を表す変数であり，どちらかの参加者で X_1 および X_2 が確実に伝達されて受信されたことを表す．さらに $guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}$ 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 のセッション固有値であって，かつ初期値である場合に，この送受信が相手に到着すれば，互いに相手のセッション固有値を共有することになる．図 3 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 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，それ以後の状態に進むことはできない．それ以後は，セッション k の X_1 と X_2 のいずれかを保有するプロセスがこの一組以外にないことが保証される．このような伝達が存在することを *onetoonecon* $X_1^{\cdot w_1} X_2^{\cdot w_2} X_2^{\cdot w_2'}$ 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．

$$\frac{\begin{array}{r} guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2} \wedge \\ X_1 \in \mathbf{Session} \wedge X_2 \in \mathbf{Session} \wedge \\ X_1 \in Init(Q) \wedge X_2 \in Init(P) \wedge \\ st(w_2) < st(w_2') \wedge P \models Q\ says\ X_2^{\cdot w_2'}\ to\ P \end{array}}{onetoonecon\ X_1^{\cdot w_1} X_2^{\cdot w_2} X_2^{\cdot w_2'}}$$

この規則の健全性は以下のように定義される．

定理 1 *onetoonecon* $X_1^{\cdot w_1} X_2^{\cdot w_2} X_2^{\cdot w_2'}$ が推論されたなら，$st(w_2')$ 番目の状態以降で，このセッションの X_1 と X_2 を持つプロセスは正常終了しないプロセスを入れても *1* 組しかない． **証明略**

ここで，*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．

$$\frac{Q \models auf\ X_1 \wedge Q \models auf\ X_2 \wedge X_1 \in \mathbf{Session} \wedge X_2 \in \mathbf{Session} \wedge guard_Q\ X_1^{\cdot w_1} X_2^{\cdot w_2}}{X_1 \sim_Q X_2}$$

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 $q \models auf\ x$ なる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．

$$\forall q \in \mathbf{Prin}[\forall x_1, x_2 \in (\mathbf{Session} \cap \{x \mid q \models auf\ x\}) [x_1 \sim_q x_2]] \quad (1)$$

ここで，下記の推論規則を導入する．

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

ただし，ϕ は論理式，$\phi[t/y]$ 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．

$$\exists x_1^{\cdot w1}, x_2^{\cdot w_2}, x_2^{\cdot w_2'} \in \mathbf{TagAM} [onetoonecon\ x_1^{\cdot w_1} x_2^{\cdot w_2} x_2^{\cdot w_2'}] \quad (2)$$

これらを用いて，すり替えのない条件は，下記により定義される．

$$\frac{(Q \models auf\ X') \wedge (X' \in \mathbf{Session}) \wedge (guard_Q\ X'^{\cdot w'} X^{\cdot w}) \wedge Q \models auf\ X \wedge 式\ (1) \wedge 式\ (2)}{Q \models unforged\ X}$$

5.3.5 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を一貫性を保って伝えている (一組のプロセスの間で，互いに相手からのメッセージのみを受信) かの検証は，以下の式により行われる．

$$\forall x \in \mathbf{Share}[\exists p,q \in \mathbf{Prin}[x \in Init(p) \wedge q \models unforged\, x \wedge p \neq q]] \quad (3)$$

5.4 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，

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

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．

$$\exists p,q \in \mathbf{Prin}[\exists x_1^{\cdot w_1}, x_2^{\cdot w_2} \in \mathbf{TagAM}[x_1 \in \mathbf{Session} \wedge x_2 \in \mathbf{Session} \wedge q \models p\ says\ x_1^{\cdot w_1}\ to\ q \wedge x_1^{\cdot w_1}\ in\ U_{n-1} \wedge p \models q\ says\ x_2^{\cdot w_2}\ to\ p \wedge x_2^{\cdot w_2}\ in\ U_n]] \quad (4)$$

上記の条件 (式 (3)，式 (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 適用例

本論文の検証方式を下記の ITU X.509 プロトコル [9] 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 [9] では，対策済みである．

$$1 : A \rightarrow B : \quad A^{\cdot 11}, \{N_{1A}^{12}, N_{2A}^{13}, B^{\cdot 14}, M_{1A}^{15}, \{M_{2A}^{16}\}_{K_B^{17}}\}_{K_A^{-1.18}}$$
$$2 : B \rightarrow A : \quad B^{\cdot 21}, \{N_{3B}^{22}, N_{4B}^{23}, A^{\cdot 24}, N_{2A}^{25}, M_{3B}^{26}, \{M_{4B}^{27}\}_{K_A^{28}}\}_{K_B^{-1.29}}$$
$$3 : A \rightarrow B : \quad A^{\cdot 31}, \{N_{4B}^{32}, B^{\cdot 33}\}_{K_A^{-1.34}}$$

その他の初期条件は下記の通りである．

$$Init(A) = \{N_{jA}, M_{jA}, A, B, K_A, K_B, K_A^{-1}\}$$
$$Init(B) = \{N_{j'B}, M_{j'B}, A, B, K_A, K_B, K_B^{-1}\}$$
$$\mathbf{Com} = \{M_{jA}, M_{j'B}, A, B, K_A, K_B, K_A^{-1}, K_B^{-1}\}$$
$$\mathbf{Session} = \{N_{jA}, N_{j'B}\} \quad \mathbf{Share} = \{M_{jA}, M_{j'B}\}$$
$$\mathbf{Prin} = \{A, B\}$$
$$\mathbf{TagAM} = \{N_{1A}^0, N_{1A}^{12}, N_{2A}^0, N_{2A}^{13}, \ldots, K_B^{-1.0}, K_B^{-1.29}\}$$
$$Type(N_{jA}) = Type(N_{j'B}) = Non$$
$$Type(M_{jA}) = Type(M_{j'B}) = Msg$$
$$Type(A) = Type(B) = ID$$
$$Type(K_A) = Type(K_B) = Key$$
$$Type(K_A^{-1}) = Type(K_B^{-1}) = Key$$

ここで，$j = 1\ or\ 2$ かつ $j' = 3\ or\ 4$ である．

本例につき，以下の二つの性質を検証する (詳細略)．

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 (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 (4) が推論できる．

7 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提案するとともに証明した．
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．

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．

参考文献

[1] Marrero, W., Clarke, E. and Jha, S.: Model Checking for Security Protocols, Research report, CMU (1997).

[2] Burrows, M., Abadi, M. and Needham, R.: A Logic of Authentication, Research Report 39, DEC SRC (1989).

[3] Gürgens, S.: SG Logic - A Formal Analysis Thechnique for Authentication Protocols, *LNCS 1361 Security Protocols 5th International Workshop*, Springer, pp. 159–176 (1997).

[4] 月村賢治, 斎藤孝通, Wen, W.: 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, 日本ソフトウェア科学会, pp. 241–244 (1999).

[5] Bird, R., Gopal, I., Herzburg, A. et al.: Systematic Design of Two-Party Authentication Protocols, *LNCS 576 Advances in Cryptology CRYPTO'91*, Splinger, pp. 44–61 (1991).

[6] Syverson, P.: On Key Distribution Protocols for Repeated Authentication, *SIGOPS Operating Systems Reviw*, Vol. 27, No. 4, pp. 24–30 (1993).

[7] Syverson, P. F. and van Oorschot, P. C.: On Unifying Some Cryptographic Protocol Logics, *IEEE Symposium on Research in Security and Privacy*, IEEE, pp. 14–28 (1994).

[8] Date, C.: *An Introduction to Database Systems Volume 2*, Addison-Wesley, pp. 20–24 (1983).

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