コンピュータセキュリティ 11－5
（2000．9．8）

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

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

メッセージの認証の検証において，受信者が受信メッセージを全体として認証していても，その中の公開キーあるいは受信者の秘密キーにより暗号化された部分メッセージを送信者自身が暗号化して送信したかどうかを知ることは出来ない．我々はこの問題を解決するために以下のアイデアを提案する．(1) 同一のメッセージの複数の出現の間に優先順序のあるプロトコルのみを対象とする．(2) セキュリティプロトコルの解析において，送信者が各送信時に保有しているメッセージの集合を推定する．これらのアイデアにより，送信者が送信された部分メッセージの内容を知っていたかどうかを決定することができる．このアイデアの実用性の検証のため，我々はBAN 論理を拡張して我々のアイデアを導入した検証システムを作成した．このシステムを簡単な例題の検証に適用することにより，我々の検証法の有効性をを示した．本論文の方式により，従来の研究では考慮されていなかった，公開キーあるいは受信者の秘密キーにより暗号化され，共用キーや送信者の秘密キーによる暗号化メッセージに含まれる，部分メッセージの信頼性の検証が可能となった．

Verification method for the authentication of encrypted messages by its sender 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

In a verification of authentication for messages, even if a receiver could authenticate a received message as a hole, the receiver cannot decide whether the sender encrypted a sub-message by public key or receiver's secret key and sent the message or not. We introduce the following ideas to solve this problem: (1) We only consider the protocol which has the precedence among the occurrences of the same sub-message, (2) In the analysis of security protocol, we infer the set of occurrences of a message that a sender possesses at each transmission. With this idea, we can decide whether the sender know the content of sub-messages of the message.
In order to investigate the realizability of this idea, we define a verification system which is an extension of BAN logic with some additional rules reflecting our idea. We apply our verifier to a simple example to show the effectiveness of the method. With this system, we can verify the reliability of a message which is encrypted by public key or receiver's secret key and is included in another encrypted message by shared key or sender's secret key, that has not been considered in previous researches.

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]，SG 論理および，筆者らの論文 [8] はその解決策を提案してきた．しかし，BAN 論理，SG 論理では，さらに次の節に示すような問題点がある．

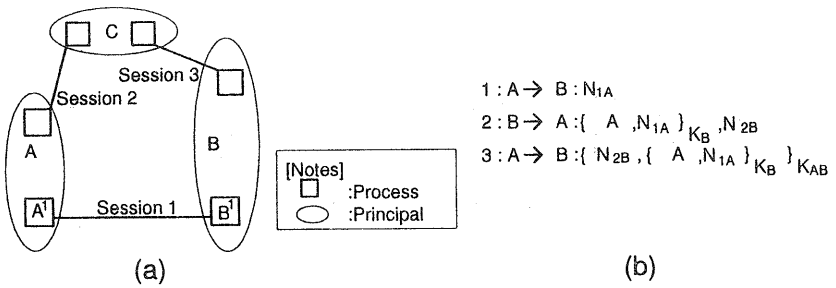

図 1: 参加者，プロセス，セッションとプロトコル

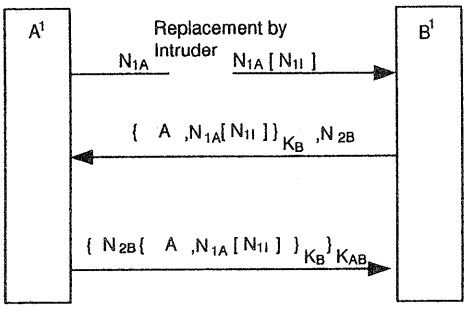

図 2: メッセージすり替えの例

1.1 問題点

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 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 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 $\{A, N_{1A}\}_{K_B}$ 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 2 は，図 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 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．

1.2 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．本論文では，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．

2 前提条件と用語

以下に前提条件と用語を示す．

[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．

2.1 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 **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.

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.

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.

Pri4,Pri5 の二つの仮定は, 前節で述べた新しいアイデアを表わす.

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.)

2.2 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.

2.2.1 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, $S^k_{P,i}$ 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.

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, 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．

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．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．

3 プロトコルのモデルと初期条件

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．

$$i : P \rightarrow Q : U_i$$

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．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．

4 論理による検証

プロトコルを検証するために，以下の論理的述語を用いる．

4.1 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 [9] 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．

4.2 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 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す.

4.3.1 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.

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

これらの規則は BAN 論理と同じである.

4.3.2 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.

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

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

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.

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

$$\frac{P \models Q\ says\ \overline{(X_1, X_2)}}{P \models Q\ says\ \bar{X}_1}$$

4.3.3 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.

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

   - 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．

5 適用例

本論文の検証法の成果を，その方法を図 1 の例に適用し，$B \models unforged\ N_{1A}$ 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．

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

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 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．

6 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 [3] 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．

本論文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．本論文の導出規則は，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．

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] 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, Vol. 41, No. 8 (2000).

[9] Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR, *LNCS 1055, Tools and Algorithms for the Construction and Analysis of Systems: second internaitonal workshop, TACAS'96*, Splinger-Verlag, pp. 147–166 (1996).