情報処理学会論文誌 Vol.56 No.3 903–910 (Mar. 2015)

# 鍵交換プロトコルにおける推論的形式分析手法

我妻 和憲[1] 後藤 祐一[1] 程 京徳[1,a)]

受付日 2014年6月30日, 採録日 2014年12月3日

**概要**：暗号プロトコルの安全性を事前に検討するために形式分析が使用されている．モデル検証や定理証明などの従来の証明的形式分析手法において，ある暗号プロトコルに，ある欠陥がないことを検証する際には，その欠陥を分析前に分析者が列挙できていなければならない．これら証明的形式分析手法では，分析者が分析前に列挙していない欠陥は検証できない．このため，分析前に分析対象の欠陥を列挙せずに前向き推論を用いてプロトコルの仕様に暗黙的に含まれる欠陥を導出するという推論的形式分析手法のアイデアが提案され，また，分析における前向き推論を基礎づける論理体系が示された．しかし，推論的形式分析手法の具体的な手法はいまだ確立されていない．本論文では，暗号プロトコルの一種である鍵交換プロトコルにおいて，推論的形式分析手法を提案し，提案手法が有用であることを実証した．証明的形式分析手法で分析者が分析前に欠陥を列挙していなかった鍵交換プロトコルにおいて，提案手法により，分析前に列挙していなかった欠陥を検出できることを確認した．したがって，提案手法は分析者が見落とした欠陥の検出に有用であるといえる．さらに，提案手法は様々な暗号プロトコルに適用させるように拡張が可能であることを示した．

**キーワード**：暗号プロトコル，鍵交換プロトコル，形式分析，前向き推論

# A Formal Analysis Method with Reasoning for Key Exchange Protocols

Kazunori Wagatsuma[1] Yuichi Goto[1] Jingde Cheng[1,a)]

Received: June 30, 2014, Accepted: December 3, 2014

***Abstract:*** Formal analysis of cryptographic protocols is used to find flaws before using the protocols. In traditional formal analysis method with proving such as model checking and theorem proving, analysts must enumerate flaws before analysis when they verify that a cryptographic protocol has not those flaws. In other words, those methods can detect only flaws that analysts enumerate. An idea of formal analysis to use forward reasoning to detect flaws of cryptographic protocols was proposed and logic systems underlying forward reasoning were presented. However, its concrete method is not established. This paper presents a concrete method of formal analysis with reasoning for key exchange protocols and shows its effectiveness. By the proposed method, we analyzed a key exchange protocol and detect flaws that analysts did not enumerate in traditional formal analysis method with proving. Therefore, the proposed method is effective for detecting flaws that analysts overlooked. Finally, we discussed that the proposed method can be extended to various cryptographic protocols.

***Keywords:*** cryptographic protocols, key exchange protocols, formal analysis, forward reasoning

## 1. はじめに

ネットワーク上での安全かつ公平な情報のやりとりのため，鍵交換，認証，デジタル署名，秘密分割，電子投票，ゼロ知識証明などの多くの暗号プロトコルが提案されている [16]．特に，鍵交換プロトコルについては多くのプロトコルが提案されている [2], [7]．

暗号プロトコルの安全性を事前に検討するために形式分析が使用されている．ネットワーク上には，やりとりされているメッセージの盗聴や改ざんなどを行う攻撃者が存在している．安全性が保証されていない暗号プロトコルで

[1] 埼玉大学大学院理工学研究科
Graduate School of Science and Engineering, Saitama University, Saitama 338–8570, Japan
[a)] cheng@ics.saitama-u.ac.jp

は，攻撃者による盗聴や改ざんによるなりすましや，機密情報の漏洩が発生する場合がある．モデル検証手法および定理証明手法が鍵交換プロトコルのための形式分析手法として利用されている．たとえば，定理証明手法においては Isabelle [14]，CafeOBJ [9] など，モデル検証手法においては Scyther [8]，ProVerif [1] などが主にあげられる．従来のモデル検証や定理証明などの証明的形式分析手法において，ある暗号プロトコルに，ある欠陥がないことを検証する際には，その欠陥を分析前に分析者が列挙できていなければならない．これら証明的形式分析手法では，分析者が分析前に列挙していない欠陥は検証できない．

一方で，分析前に分析対象の欠陥を列挙せずに，前向き推論を用いてプロトコルの仕様に暗黙的に含まれる欠陥を導出するという推論的形式分析手法のアイデアが提案され，また，分析における前向き推論を基礎づける論理体系が示された [5]．しかし，推論的形式分析手法の具体的な手法はいまだ確立されていない．

本論文では，暗号プロトコルの一種である鍵交換プロトコルにおいて，推論的形式分析手法を提案し，提案手法が有用であることを実証する．証明的形式分析手法で分析者が分析前に欠陥を列挙していなかった鍵交換プロトコルにおいて，提案手法により，分析前に列挙していなかった欠陥を検出できることを確認する．それによって，提案手法は分析者が見落とした欠陥の検出に有用であるといえる．最後に，提案手法は様々な暗号プロトコルに適用させるように拡張が可能であることを示す．

## 2. 鍵交換プロトコルのための証明的形式分析手法

鍵交換プロトコルとは，暗号技術を用いて参与者同士のみがセッション鍵を交換するための手順である [2]．鍵交換プロトコルの仕様は，手順ごとに以下の形式で表現される．

$$N.X_1 \rightarrow X_2 : Y_1, Y_2, \cdots, Y_n \quad (1)$$

上記の形式において，鍵交換プロトコルの仕様は，$N$ 番目の手順において，参与者 $X_1$ が $X_2$ に対して $Y_1, Y_2, \cdots, Y_n$ というデータを送信することを表す．$X_1$，$X_2$ には鍵交換プロトコルにおける登場人物が代入される．登場人物は通常，参与者 $A$，$B$，および攻撃者 $I$ で，セッション鍵の配布者 $S$ が存在する場合もある．また，$Y_1, Y_2, \cdots, Y_n$ には送信されるデータが代入される．具体的には以下のとおり定義される [2]．

- $A$，$B$，$S$，$I$（登場人物 $A$，$B$，$S$，$I$ の識別子．ただし $I$ は攻撃者の識別子である）
- $SK$（セッション鍵）
- $K_{X_1X_2}$（$X_1$ と $X_2$ の共通鍵）
- $E_X$（$X$ の公開鍵）
- $Sig_X$（$X$ の秘密鍵）

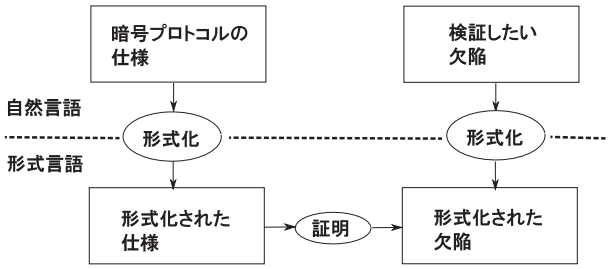



図 1 証明的形式分析手法の概要
**Fig. 1** Overview of proving formal analysis method.

- $N_X$（$X$ のナンス）
- $T_X$（$X$ のタイムスタンプ）
- $\{Y\}_K$（$Y$ を任意のデータとして，暗号化鍵 $K$ によって暗号化された $Y$．複数のデータを暗号化することも可能）
- $Y+1$（値をインクリメントされたデータ $Y$）
- $Y'$（古いデータ $Y$．ここでは $Y$ はナンス，タイムスタンプ，セッション鍵のいずれかである）

ただし，鍵交換プロトコルの仕様は通常，参与者を $S$（ただし，$S$ のないプロトコルもある），$A$，$B$ とした場合における，それぞれの参与者の正常な振舞いが記載されている．すなわち，仕様には以下のことが記載されていない．

- 攻撃中における各登場人物の振舞い
- 攻撃者の振舞い
- データの送信以外の参与者の振舞い
  - データの受信
  - （暗号化されたデータの復号化による）メッセージの取得

モデル検証手法および定理証明手法は，証明的形式分析手法である．証明的形式分析手法とは，検証したい欠陥を分析前に列挙し，形式化された鍵交換プロトコルの仕様の下で列挙した欠陥が存在すること，あるいは存在しないことを証明する手法である [5]．鍵交換プロトコルにおける欠陥とは，そのプロトコルを用いた通信において，攻撃が成立するということである．**図 1** は証明的形式分析手法の大まかな手順を説明した図であり，その手順は 3 つに分割できることが示されている．まず分析者は検証したい欠陥を列挙し，それを形式的に記述する．次に，分析者は分析対象となる鍵交換プロトコルの仕様を形式的に記述する．最後に，分析者は鍵交換プロトコルの仕様を前提として，列挙した欠陥が存在することが証明されるか確認する．もし列挙した欠陥が存在することが証明されたら，分析対象となる鍵交換プロトコルは安全ではないといえる．鍵交換プロトコルに 1 つでも欠陥が見つかった場合，その鍵交換プロトコルは安全ではない．そのため，ある鍵交換プロトコルが安全であることを証明的形式分析手法を用いて検証するためには，分析者はその鍵交換プロトコルの欠陥を分析前に列挙できていなければならない．なぜなら，分析対象

とされていない欠陥について，証明的形式分析手法は何も主張できないからである．つまり，証明的形式分析手法では，分析者が列挙していない欠陥は検証できない．

## 3. 鍵交換プロトコルのための推論的形式分析手法

### 3.1 鍵交換プロトコルのための推論的形式分析手法の概要

前向き推論を用いてプロトコルの仕様に含まれる暗黙的な欠陥を導出するという推論的形式分析手法のアイデアが提案された [5]．推論的形式分析手法とは，形式化された鍵交換プロトコルの仕様を前向き推論の前提として利用し，その前提の下で成り立つ性質や特徴を前向き推論によって導出し，その導出結果を分析し，欠陥を検出する手法である．前向き推論を用いた形式分析において，分析者は分析前に分析対象の欠陥を列挙する必要がない．分析者は，前向き推論により仕様から導出された性質や特徴を分析する．分析者が列挙できる欠陥の場合，分析者は導出された性質の中から探すことができる．分析者が分析開始前に列挙できなかった欠陥も，分析者は導出された性質を分析し，検出できる可能性がある．よって，推論的形式分析手法は，分析者が分析前に列挙していなかった欠陥を検出することを支援することができる．

**図 2** は推論的形式分析手法における 3 つのステップを説明した図である．現在，推論的形式分析手法には，鍵交換プロトコルの仕様の形式化，前向き推論，導出された結論の自然言語化および分析という 3 つのステップが含まれることが分かっている [5]．そして，前向き推論を基礎付ける論理体系として，強相関論理 [3], [4] が適していることが理論的に示されている [5]．また，推論的形式分析手法のステップにおける前向き推論によって，形式化された鍵交換プロトコルの仕様から導出できる限りの結論を導出する必要がある [11]．そのため，自動的に前向き推論を行うためのプログラムである前向き推論エンジンが提案・開発されている [6]．しかし，推論的形式分析手法の具体的な手法はいまだ確立されていなかった．

### 3.2 鍵交換プロトコルのための推論的形式分析手法の手順

推論的形式分析手法の具体的手順のうち，形式化手順はすでに提案されている [17]．本論文では，推論的形式分析手法における前向き推論，および前向き推論によって導出された結論の分析手順を提案する．分析者は鍵交換プロトコルの攻撃に対する知識に関係なく，本提案手法に従い形式分析を行うことができる．鍵交換プロトコルのための推論的形式分析手法における形式化，前向き推論，導出された結論の分析の具体的手順は以下のとおりである．

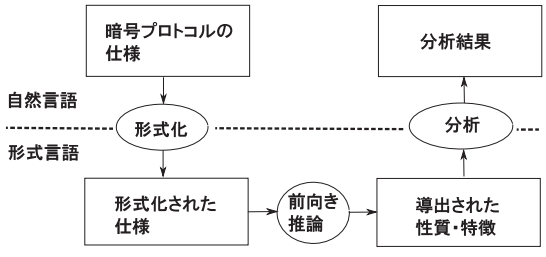



**図 2** 推論的形式分析手法の概要
**Fig. 2** Overview of reasoning formal analysis method.

**鍵交換プロトコルの形式化手順**

2 章の形式 (1) で表現される鍵交換プロトコルの仕様から，一階述語論理式への書き換えを行うための手順は以下のとおりである．以下 2 つの事柄について形式化を行ったら，鍵交換プロトコルの形式化は終了とする．

- 仕様に示されている参与者の振舞い
- 仕様に暗黙に示されている事柄

形式化に用いる一階述語論理式における述語は以下の 7 個である．

- $Eq(x_1, x_2)$：$x_1$，$x_2$ は同一のものである．
- $Get(p, x)$：$p$ は $x$ を取得する．
- $Parti(p)$：$p$ はプロトコルの参与者である．
- $Recv(p, x)$：$p$ は $x$ を受け取る．
- $Send(p_1, p_2, x)$：$p_1$ は $p_2$ に $x$ を送る．
- $Sesk(x)$：$x$ はセッション鍵である．
- $Start(p_1, p_2)$：$p_1$ と $p_2$ がセッションを開始する．

また，形式化に用いる一階述語論理式における関数は以下の 10 個である．

- $data(x_1, \ldots, x_n)$：送受信するデータ群 $x_1, \ldots, x_n$
- $enc(k, x_1, \ldots, x_n)$：鍵 $k$ によって暗号化されたデータ群 $x_1, \ldots, x_n$
- $id(p)$：$p$ の識別子
- $nonce(p)$：$p$ のナンス
- $old(x)$：$x$ の古いデータ
- $pk(p)$：$p$ の公開鍵
- $plus(x)$：$x$ の値をインクリメントしたデータ
- $sk(p)$：$p$ の秘密鍵
- $symk(p_1, p_2)$：$p_1$ と $p_2$ の共通鍵
- $tstamp(p)$：$p$ のタイムスタンプ

さらに，形式化に用いる一階述語論理式における個体定数は以下の 5 個である．

- $a$，$b$：プロトコル参与者
- $i$：攻撃者
- $s$：鍵配布サーバ
- $sesk$：セッション鍵

**仕様に示されている参与者の振舞い**

(1) それぞれの手順における登場人物の振舞いを一階述語論理式で表現する．何番目の手順を形式化するかに応じて，以下の規則により論理式を作成する．

- 1 番目の手順の場合，
  (a) $Start(p_1, p_2) \Rightarrow Send(p_1, p_3, data(x_1, \ldots, x_n))$

という論理式を作成する．$p_i$ は登場人物を表す個体変数，$x_i$ はデータを表す個体変数，関数 $data$ における引数の数 $n$ は，対応する手順で送信されるデータの数とする．ただし，1つの鍵で暗号化された複数のデータの数は1つとする．

- 2番目以降の手順の場合，
  (a) $Recv(p_1, data(x_1, \ldots, x_m)) \Rightarrow Parti(p_1) \Rightarrow Send(p_1, p_2, data(x_1, \ldots, x_n))$ という論理式を作成する．ただし，対応する手順の直前で受信されるデータの数を $m$，その直後に送信されるデータの数を $n$ する．

(2) 対応する手順におけるデータの送信者および受信者に応じて，手順 (1) で作成した論理式の個体変数 $p_1$，$p_2$，$p_3$ を置換する．
  (a) $Start(p_1, p_2) \Rightarrow Send(p_1, p_3, data(x_1, \ldots, x_n))$ という論理式において $p_1$，$p_2$，$p_3$ を置換する．
    - 1番目の手順におけるデータの受信者が $S$ の場合，$p_3$ を $s$ へ置換する．
    - 1番目の手順におけるデータの受信者が $B$ の場合，$p_3$ を $p_2$ へ置換する．
  (b) $Recv(p_1, data(x_1, \ldots, x_m)) \Rightarrow Parti(p_1) \Rightarrow Send(p_1, p_2, data(x_1, \ldots, x_n))$ という論理式において，対応する手順におけるデータの送信者あるいは受信者に応じて $p_1$，$p_2$ を置換する．
    - 送信者が $S$ の場合，$p_1$ を $s$ へ置換する．
    - 送信者が $A$ の場合，$p_1$ は置換しない．
    - 送信者が $B$ の場合，$p_1$ を $p_2$ へ置換する．
    - 受信者が $S$ の場合，$p_2$ を $s$ へ置換する．
    - 受信者が $A$ の場合，$p_2$ を $p_1$ へ置換する．
    - 受信者が $B$ の場合，$p_2$ は置換しない．

(3) 対応する手順で送信される各データに応じて，手順 (1) で作成した論理式の個体変数 $x_1, \cdots, x_n$ を置換する．$i = 1, \cdots, n$ に対して，以下の作業を行う．
- $Y_i$ が暗号化されていないデータの場合，送信されるデータの種類に応じて，$x_i$ を関数または個体変数へ置換する．
  - $A$ のナンスであるなら，$nonce(p_1)$ へ置換する．
  - $B$ のナンスであるなら，$nonce(p_2)$ へ置換する．
  - $S$ のナンスであるなら，$nonce(s)$ へ置換する．
  - $A$ の識別子であるなら，$id(p_1)$ へ置換する．
  - $B$ の識別子であるなら，$id(p_2)$ へ置換する．
  - $S$ の識別子であるなら，$id(s)$ へ置換する．
  - $A$ のタイムスタンプであるなら，$tstamp(p_1)$ へ置換する．
  - $B$ のタイムスタンプであるなら，$tstamp(p_2)$ へ置換する．
  - $S$ のタイムスタンプであるなら，$tstamp(s)$ へ置換する．
  - セッション鍵 $SK$ であるなら，個体変数 $k$ へ置換する．
  - その他のデータであるなら，$x_i$ は置換しない．
- $Y_i$ が値をインクリメントされたデータの場合，$x_i$ を $plus(x'_i)$ へ置換する．
- $Y_i$ が鍵 $k$ で暗号化されたデータ $\{Y'_1, \cdots, Y'_n\}_k$ の場合，
  (a) $x_i$ を $enc(k, x'_1, \cdots, x'_n)$ へ置換する．
  (b) 鍵を表す変数 $k$ について，暗号化に使われている鍵の種類から $k$ を置換する．
    - $A$ と $S$ の共通鍵であるなら，$symk(p_1, s)$ へ置換する．
    - $B$ と $S$ の共通鍵であるなら，$symk(p_2, s)$ へ置換する．
    - $A$ と $B$ の共通鍵であるなら，$symk(p_1, p_2)$ へ置換する．
    - $A$ の公開鍵であるなら，$pk(p_1)$ へ置換する．
    - $B$ の公開鍵であるなら，$pk(p_2)$ へ置換する．
    - $S$ の公開鍵であるなら，$pk(s)$ へ置換する．
    - $A$ の秘密鍵であるなら，$sk(p_1)$ へ置換する．
    - $B$ の秘密鍵であるなら，$sk(p_2)$ へ置換する．
    - $S$ の秘密鍵であるなら，$sk(s)$ へ置換する．
    - セッション鍵 $SK$ であるなら，$k$ は置換しない．

(4) 手順 (3) と同様に，送信されるデータの種類に応じて，$x'_i$ を関数または個体定数へ置換する．

(5) 手順 (1)〜(3) で作成した論理式 $Start(p_1, p_2) \Rightarrow Send(p_1, p_3, data(x_1, \ldots, x_n))$，$Recv(p_1, data(x_1, \ldots, x_m)) \Rightarrow Parti(p_1) \Rightarrow Send(p_1, p_2, data(x_1, \ldots, x_n))$ における論理結合子 $\Rightarrow$ の前件と後件について，以下の作業を行う．
  (a) 前件と後件に含まれている共通の個体定数どうしを，同じ個体変数 $x_n$ に書き直す（ただし，$s$ は鍵配布サーバであるため書き直す対象から除外する）．
  (b) 前件と後件のどちらか一方のみに含まれている個体変数があれば，対応する手順でやりとりされるデータに応じて個体定数を代入する．
    - セッション鍵 $SK$ であるなら，$sesk$ を代入する．
    - その他のデータであるなら，任意の個体定数を定義し，代入する．

(6) 作成した論理式に存在する個体変数に対応させて，量化子 $\forall$ を追加する．

(7) $Start(p_1, p_2)$ という論理式における個体変数 $p_1$，$p_2$ を個体定数 $a$，$b$，$i$ のいずれかに置換する．ただし，$p_1$，$p_2$ は異なる個体定数とする．

**仕様に暗黙に示されている事柄**

仕様に記載されていない事柄のうち，プロトコルごとに異なる部分の形式化手順を記述する．以下 2 つの形式化が完了したら，この作業は完了となる．

- 攻撃者の振舞い
  ただし，今回の提案手法における攻撃者の振舞いは Dolev-Yao モデル [10] に基づくものとする．
- 攻撃者を含めた，全登場人物に共通する振舞い

**攻撃者の振舞い**

攻撃者は以前のセッションでやりとりした古いデータ，および古いセッション鍵を取得していることが想定されている [2]．それについて形式化を行う．

(1) 上述の “仕様に記載されている参与者の振舞い” における形式化手順で作成した論理式から，送信されているデータ（述語 $Send$ の引数 $data(x_1, \cdots, x_n)$ で表される）を表す項を列挙する．
(2) 列挙された項を $T_1, \cdots, T_n$ とすると，$m = 1, \cdots, n$ の場合すべてにおいて，$Start(p_1, p_2) \Rightarrow Get(i, T_m)$ という論理式を作成する．
(3) $T_m$ において，ナンス，タイムスタンプ，セッション鍵を表す関数が含まれていれば，該当する関数（仮に $f(y)$ とする）を $old(f(y))$ へ置き換える．
(4) 以下の論理式を生成する．
  - 攻撃者が別の参与者 $p$ にデータ $x$ を送信した場合，$p$ がそのデータを受信する．
    $\forall p \forall x(Send(i, p, x) \Rightarrow Recv(p, x))$
  - 攻撃者がプロトコルで送信されるデータをすべて取得（盗聴）する．
    $\forall p_1 \forall p_2 \forall x(Send(p_1, p_2, x) \Rightarrow Get(i, x))$

**攻撃者を含めた，全登場人物に共通する振舞い**

対象の鍵交換プロトコルで使われているデータの種類に応じて，以下の論理式を生成する．

- ある登場人物 $p_1$（攻撃者を含む）が自分の共通鍵で暗号化された $x_1, \cdots, x_n$ を取得した場合，元々のメッセージを取得する（共通鍵を使わないプロトコルの場合，これらの論理式は生成しない）．
  $\forall p_1 \forall p_2 \forall x_1 \ldots \forall x_n(Get(p_1, enc(symk(p_1, p_2), x_1, \cdots, x_n)) \Rightarrow Get(p_1, data(x_1, \cdots, x_n)))$
- ある登場人物 $p$（攻撃者を含む）が自分の公開鍵で暗号化された $x_1, \cdots, x_n$ を取得した場合，元々のメッセージを取得する（公開鍵を使わないプロトコルの場合，これらの論理式は生成しない）．
  $\forall p \forall x_1 \ldots \forall x_n(Get(p, enc(pk(p), x_1, \cdots, x_n)) \Rightarrow Get(p, data(x_1, \cdots, x_n)))$
- $A$, $B$, $S$ は参与者であり，攻撃者 $I$ は参与者ではない．
  $Parti(a), Parti(b), Parti(s), \neg Parti(i)$
- $SK$ および古い $SK$（$SK'$）はセッション鍵である．
  $Sesk(sesk), Sesk(old(sesk))$
- ある登場人物 $p$（攻撃者を含む）がデータ $x$ を受け取った場合，そのデータを取得する．
  $\forall p \forall x(Recv(p, x) \Rightarrow Get(p, x))$
- $symk(p_1, p_2)$ と $symk(p_2, p_1)$ は同一のものである（提案手法において共通鍵に関する前向き推論を行うために必要な前提である）．
  $\forall p_1 \forall p_2(Eq(symk(p_1, p_2), symk(p_2, p_1)))$
- ある登場人物 $p$（攻撃者を含む）が暗号化された $x_1, \cdots, x_n$ を取得した場合，それがセッション鍵で暗号化したものであり，かつ $p$ がそのセッション鍵を取得していれば，元々のメッセージを取得する．
  $\forall k \forall p \forall x_1 \ldots \forall x_n(Get(p, enc(k, x_1, \cdots, x_n)) \Rightarrow ((Sesk(k) \wedge Get(p, k)) \Rightarrow Get(p, data(x_1, \cdots, x_n))))$

以上で形式化の具体的手順は完了となる．ここで生成された論理式を前向き推論における前提として使用する．

**前向き推論手順**

前向き推論において，前提から導き出せるだけの結論を導出する．しかし，前提から導き出せるだけの結論を手作業で導出するのは分析者にとって手間がかかる．FreeEnCal [6] は，命題，一階述語，二階述語レベルの論理式とそれらの図式を扱うことのできる前向き推論エンジンである．FreeEnCal は，論理式を構成する語彙，論理式の構成規則，公理，推論規則を入力として自由に与えられるようにすることで，任意の論理体系や形式理論を扱うことができる．さらに，入力された論理式を前提として，その前提から成り立つ結論を自動的に導出できる．そのため，前向き推論による，形式化手順において生成された論理式（前提）から成り立つ性質や特徴の導出に FreeEnCal を利用することができる．今回は，FreeEnCal を用いて前向き推論を自動的に行う．

次に，参与者が正しく送信データを受け取った場合と，攻撃者が送信データを受け取った場合に場合分けし，以下の手順で論理式を付け加え，それぞれの場合において，前向き推論を繰り返す．

- $p$, $q$ を任意の参与者を表す個体定数として，導出された結論に $Send(p, q, x)$ が含まれ，かつ，$Recv(q, x)$, $Recv(i, x)$ のいずれも含まれていない場合，$Recv(p, x)$ または $Recv(i, x)$ のいずれかを新しい前提として追加する．
- $Recv(i, x)$ が結論に含まれ，かつ “$Send(i,$” で始まる原子論理式と “$Recv(i,$” で始まる原子論理式の数が等しくない場合，以下を行う．
  (1) 導出された論理式のうち，$Get(i, z)$（$z$ は任意の項）という形式の論理式を列挙する．
  (2) 導出された論理式のうち，$Parti(q)$（$q$ は任意の項）という形式の論理式を列挙する．
  (3) $Send(i, q, z')$ を新しく追加する前提として作成する．

(4) 列挙された $Parti(q_1), \cdots, Parti(q_n)$ に基づき，$Send(i, q, z')$ における $q$ を $q_1, \cdots, q_n$ のいずれかに置換する．

(5) 列挙された論理式 $Get(i, z_1), \cdots, Get(i, z_n)$ に基づき，$z_1, \cdots, z_n$ および攻撃者自身が元々所有しているデータ（鍵，ナンス，タイムスタンプなど）を組み合わせて送信するデータを表す項を作成する．

上述の手順において，新たな論理式が付け加えられない場合は分析手順へ進む．

**分析手順**

攻撃者の攻撃が成立したことを表す論理式が導出された結論に含まれているのかを調べる．

(1) 導出された結論から攻撃の成立と関係ある論理式が導出されているかチェックするための準備として，攻撃の成立の有無とは関係ない論理式を除去する．
- 変数を含む論理式
- 二項演算子 $\wedge, \Rightarrow$ を含む論理式

(2) 前の手順で除去した論理式において，攻撃の成立と関係ある論理式が導出されているかチェックする．
- プロトコルが正常終了したことを表す論理式
- 攻撃者が機密情報を得たことを示す論理式
- 参与者が，攻撃者によって改ざんされたデータを受信したことを示す論理式

(3) 攻撃者が機密情報を得たか，または参与者が攻撃者によって改ざんされたデータを受信したことを示す論理式が導出された結論に含まれ，かつプロトコルの一番最後の手順で送信するべきデータを，いずれかの参与者が送信，あるいは取得したことを示す論理式が含まれているなら，攻撃の成立，すなわち欠陥ありと判定する．

上記の分析手順における，プロトコルの正常終了を表す論理式とは，プロトコルの一番最後の手順で送信するべきデータを，いずれかの参与者が送信，あるいは取得したことを示す論理式である．ある参与者が，最後に送信あるいは取得すべきデータの値を事前に知っている場合，分析者は，前向き推論によって得られた論理式の中から，送信あるいは取得されたデータの値が一致するかどうか確認する．一致する場合，一番最後の手順で送信するべきデータを，いずれかの参与者が送信，あるいは取得した，すなわちプロトコルが正常終了したといえる．

## 4. 有用性の実証

今回提案した手法は鍵交換プロトコルの形式分析に有用であることを実証する．証明的形式分析手法で分析者が分析前に欠陥を列挙していなかった鍵交換プロトコルにおいて，提案手法により，分析前に列挙していなかった欠陥を検出できることを確認する．そのための事例として，今回は鍵交換プロトコルの一種である Otway-Rees プロトコル [13] を用いる．プロトコルの仕様は以下のとおりである．

1. $A \rightarrow B : M, A, B, \{N_A, M, A, B\}_{K_{AS}}$
2. $B \rightarrow S : M, A, B, \{N_A, M, A, B\}_{K_{AS}}, \{N_B, M, A, B\}_{K_{BS}}$
3. $S \rightarrow B : M, \{N_A, SK\}_{K_{AS}}, \{N_B, SK\}_{K_{BS}}$
4. $B \rightarrow A : M, \{N_A, SK\}_{K_{AS}}$

この鍵交換プロトコルには 2 つの欠陥が指摘されている．第 1 に，攻撃者が参与者の暗号化データを自分の鍵で暗号化したデータに改ざんし，セッション鍵を取得できてしまう [2]．以下の順番でデータをやりとりすることにより，攻撃が成立する．

1. $A \rightarrow B : M, A, B, \{N_A, M, A, B\}_{K_{AS}}$
2'. $B \rightarrow I(S) : M, A, B, \{N_A, M, A, B\}_{K_{AS}}, \{N_B, M, A, B\}_{K_{BS}}$
2''. $I(B) \rightarrow S : M, A, I, \{N_A, M, A, B\}_{K_{AS}}, \{N_I, M, A, B\}_{K_{IS}}$
3'. $S \rightarrow I(B) : M, \{N_A, SK\}_{K_{AS}}, \{N_I, SK\}_{K_{IS}}$
4'. $I(B) \rightarrow A : M, \{N_A, SK\}_{K_{AS}}$

第 2 に，参与者 $A$ の暗号化メッセージを攻撃者が送り返した場合，$A$ が $M$，$A$，$B$ をセッション鍵として誤認識してしまう [7]．以下の順番でデータをやりとりすることにより，攻撃が成立する．

1. $A \rightarrow I(B) : M, A, B, \{N_A, M, A, B\}_{K_{AS}}$
4'. $I(B) \rightarrow A : M, \{N_A, M, A, B\}_{K_{AS}}$

1 番目の欠陥についての検証 [15]，および 2 番目の欠陥についての検証 [12] が行われた．しかし，各検証手法において，もう一方の欠陥はそれぞれ列挙されていなかったため検出されなかった．証明的形式分析手法は分析者が列挙していない欠陥は検出できないため，この事例は分析者が検証すべき欠陥を列挙していなかった事例であるといえる．このようなことは，今回の事例以外にも，証明的形式分析手法において発生する可能性がある．

提案手法でこのプロトコルに存在する上記 2 つの欠陥を両方とも検出できたならば，提案手法は，分析者が分析前に欠陥を列挙しているかどうかにかかわらず欠陥を検出できる手法であるといえる．

提案手法では Otway-Rees プロトコルに存在する 2 つの欠陥を表す論理式が導出できることが確認できた．まず，第 1 の欠陥について，$Get(a, data(nonce(a), sesk))$, $Get(i, sesk)$ という論理式が導出された結論に含まれていた．1 番目の論理式は，参与者 $A$ がプロトコルの最後の手順で取得すべきメッセージである，$A$ のナンスとセッション鍵を取得したことを表す．2 番目の論理式は，攻撃者が機密情報であるセッション鍵を取得したことを表す．すなわち，攻撃者が機密情報であるセッション鍵を取得し，かつ $A$ はそれに気付かずにプロトコルにおける手順を完了したことを示している．この結論は，指摘され

ている第1の欠陥と一致する．次に，第2の欠陥について，$Get(a, data(nonce(a), m, id(a), id(b)))$ という論理式が導出された結論に含まれていた．この論理式は，$A$ はプロトコルで決められた手順を完了したことを示す．本来は，$A$ はプロトコルで決められた最後の手順を完了したことを表す論理式として，$Get(a, data(nonce(a), sesk))$ が導出されるべきである．しかし，$A$ は $nonce(a)$ の値は元々知っているが，$sesk$ の値を事前に知らない．そのため，$Get(a, data(nonce(a), sesk))$ のような改ざんされたメッセージを取得した場合でも，$A$ はメッセージの改ざんに気付かない可能性がある．さらに，取得したメッセージが，セッション鍵として誤認識する可能性がある．この結論は，指摘されている第2の欠陥と一致する．以上により，提案手法では Otway-Rees プロトコルに存在する2つの欠陥を表す論理式が導出できることが確認できた．したがって，提案手法は分析者が分析前に欠陥を列挙しているかどうかにかかわらず欠陥を検出できる手法であるといえる．

## 5. 考察

現在の提案手法には，2つの課題がある．第1に，提案手法を実行するのに非常に手間がかかることである．提案手法における前向き推論手順において，参与者が正しく送信データを受け取った場合と，攻撃者が送信データを受け取った場合に場合分けし，それぞれの場合において前向き推論を繰り返すという作業がある．提案手法に従い形式分析を行うために，選択可能な論理式を追加したすべての場合について，前向き推論を行わなければならない．しかし，現状では選択可能な論理式を追加したすべての場合ついて前向き推論を行うのは非常に手間がかかる．4章の事例研究では，攻撃が成立する場合についてのみ前向き推論を行った．また，鍵交換プロトコルの形式化や導出された大量の論理式を分析する作業も，分析者にとって手間がかかる．このため，提案手法の自動化が必要である．第2に，検出できる欠陥の種類に制限があることである．提案手法における攻撃者の振舞いは Dolev-Yao モデル [10] に基づいている．このモデルでは，攻撃者による盗聴，改ざん，攻撃者が持っている鍵を用いた復号化が想定されている．そのため，提案手法ではこれらに由来する欠陥は検出可能であるが，Dolev-Yao モデルで扱っていない攻撃は検出できない．たとえば，このモデルでは暗号技術自体は安全であるという想定のため，暗号技術自体の欠陥を検出できない．このため，攻撃者の振舞いのモデルを変更あるいは拡張し，より多くの欠陥を検出できるようにする必要がある．

今回提案した手法はあらゆる鍵交換プロトコルに汎用的に用いることができる．なぜならば，鍵交換プロトコルにおいて，登場人物およびやりとりするデータの種類は定義された範囲に限定されており，登場人物がデータを送受信する順番だけがプロトコルごとに異なるためである．したがって，今回の事例の対象とした鍵交換プロトコル以外に対しても，当該鍵交換プロトコルに適した攻撃者モデルを用意することで，提案手法を用いて形式分析を行うことができる．

今回提案した手法は様々な暗号プロトコルに適用させるように拡張が可能である．提案手法において定義された，一階述語論理式における述語，個体定数，関数，論理式は鍵交換プロトコルに限らず，様々な暗号プロトコルにも利用できる．暗号プロトコルにおいて，共通する事柄が2つある．第1に，参与者は暗号技術を用いてデータを送受信することである．なぜなら，暗号プロトコルとは，参与者同士が暗号技術を用いてデータをやりとりする手順の集合だからである．第2に，参与者はデータの送信，受信，復号化によるメッセージの取得という動作を繰り返すことである．暗号プロトコルにおいて，送信されたデータは誰かが受信する．そして，受信したデータが暗号化されている場合，暗号鍵に対応する復号鍵を受信者が持っていれば，元々のメッセージを取得できる．受信者は，取得したデータに応じて，プロトコルで定義された次のデータを送信する．したがって，提案手法で定義した述語および関数，すなわち，データの送信，受信，取得をそれぞれ表す $Send$，$Recv$，$Get$ という3つの述語，および公開鍵，秘密鍵，共通鍵をそれぞれ表す $pk(p)$，$sk(p)$，$symk(p_1, p_2)$ という3つの関数は，他の種類の暗号プロトコルにも使用することができる．提案手法において，定義されていないデータは，新しく個体定数あるいは関数を定義すればよい．また，攻撃者の振舞いのモデルは，当該暗号プロトコルの種類に応じて，適切な攻撃者の振舞いのモデルを用意することで，提案手法を用いて形式分析を行うことができる．

## 6. おわりに

本論文では，鍵交換プロトコルのための推論的形式分析手法を提案した．次に，証明的形式分析手法で分析者が分析前に欠陥を列挙していなかったという事例において，提案手法を用いてそれらの欠陥を表す論理式が導出できることが確認できた．これによって，提案した手法を用いることで，分析者が分析前に欠陥を列挙しているかにかかわらず欠陥を検出できることを示した．したがって，提案手法は分析者が見落とした欠陥の検出に有用であるといえる．

今後の課題は以下の4つである．第1に提案手法の有用性の実証に用いた鍵交換プロトコルは Otway-Rees プロトコルのみである．そのため，今後は現在提案されている他の鍵交換プロトコルにも提案手法が有用であることを実証する．第2に，作業自体に手間がかかり，本論文でも提案手法におけるすべての手順を実行できていないという課題がある．そのため，今後は提案手法の自動化および効率化を行う．第3に，検出できる欠陥の種類に制限がある．そのため，今後はより多くの欠陥を検出できるように適切な

攻撃者の振舞いのモデルについて検討する．第 4 に，今回提案した手法は，形式分析の対象は鍵交換プロトコルに限定されている．そのため，今後は様々な暗号プロトコルに適用できるように提案手法を拡張し，その手法が暗号プロトコルの形式分析に有用であることを実証する．

## 参考文献


[1] Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules, *14th IEEE Computer Security Foundations Workshop*, pp.82–96, IEEE (2001).

[2] Boyd, C. and Mathuria, A.: *Protocols for authentication and key establishment*, Springer (2003).

[3] Cheng, J.: A strong relevant logic model of epistemic processes in scientific discovery, *Information modelling and knowledge bases XI*, Vol.61, pp.136–159 (2000).

[4] Cheng, J.: Strong relevant logic as the universal basis of various applied logics for knowledge representation and reasoning, *Frontiers in Artificial Intelligence and Applications*, Vol.136, pp.310–320 (2006).

[5] Cheng, J. and Miura, J.: Deontic relevant logic as the logical basis for specifying, verifying, and reasoning about information security and information assurance, *1st International Conference on Availability, Reliability and Security*, pp.601–608, IEEE-CS (2006).

[6] Cheng, J., Nara, S. and Goto, Y.: FreeEnCal: A forward reasoning engine with general-purpose, *Knowledge-Based Intelligent Information and Engineering Systems*, LNAI, Vol.4693, pp.444–452, Springer (2007).

[7] Clark, J. and Jacob, J.: A survey of authentication protocol literature: Version 1.0. http://www.cs.york.ac.uk/~jac/ (1997).

[8] Cremers, C.J.F.: The Scyther tool: Verification, falsification, and analysis of security protocols, *Computer Aided Verification*, LNCS, Vol.5123, pp.414–418, Springer (2008).

[9] Diaconescu, R. and Futatsugi, K.: Cafeobj report, *World Scientific* (1998).

[10] Dolev, D. and Yao, A.C.: On the security of public key protocols, *IEEE Trans. Information Theory*, Vol.29, No.2, pp.198–208, IEEE (1983).

[11] Gao, H., Goto, Y. and Cheng, J.: A systematic methodology for automated theorem finding, *Theoretical Computer Science*, Vol.554, pp.2–21, Elsevier (2014).

[12] Nesi, M. and Nocera, G.: Deriving the type flaw attacks in the Otway-Rees protocol by rewriting, *Nordic Journal of Computing*, Vol.13, No.1, pp.78–97, Publishing Association Nordic Journal of Computing (2006).

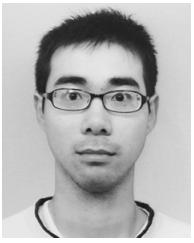
[13] Otway, D. and Rees, O.: Efficient and timely mutual authentication, *ACM SIGOPS Operating Systems Review*, Vol.21, No.1, pp.8–10, ACM (1987).

[14] Paulson, L.C.: Isabelle: A generic theorem prover, LNCS, Vol.828, Springer (1994).

[15] Paulson, L.C.: Proving properties of security protocols by induction, *10th IEEE Computer Security Foundations Workshop*, pp.70–83, IEEE (1997).

[16] Schneier, B.: *Applied cryptography: Protocols, algorithms, and source code in C*, John Wiley and Sons, Inc (1996). 暗号技術大全第 2 版，ソフトバンクパブリッシング (2003).

[17] Wagatsuma, K., Anze, S., Goto, Y. and Cheng, J.: Formalization for formal analysis of cryptographic protocols with reasoning approach, *Future Information Technology*, LNEE, Vol.309, pp.211–218, Springer (2014).


**我妻 和憲** （学生会員）

平成 23 年埼玉大学工学部情報システム工学科卒業．平成 25 年同大学大学院理工学研究科博士前期課程数理電子情報系専攻情報システム工学コース修了．現在，同博士後期課程理工学専攻数理電子情報コース在学中．暗号プロトコルの形式分析の研究に従事．



**後藤 祐一** （正会員）

平成 13 年埼玉大学工学部情報システム工学科卒業．平成 15 年同大学大学院理工学研究科博士前期課程情報システム工学専攻修了．平成 17 年同博士後期課程情報数理科学専攻修了．博士（工学）．平成 17 年同大学工学部助手．平成 19 年同大学大学院理工学研究科助教．知識工学の研究に従事．人工知能学会，ACM，IEEE-CS 各会員．



**程 京徳** （正会員）

昭和 57 年中国清華大学計算機科学技術系卒業．昭和 61 年九州大学大学院工学研究科修士課程情報工学専攻修了．平成元年同博士後期課程情報工学専攻修了．工学博士．平成元年九州大学工学部助手．平成 3 年同助教授．平成 8 年九州大学大学院システム情報科学研究科教授．平成 11 年埼玉大学大学院理工学研究科教授．ソフトウェア工学，知識工学および情報セキュリティ工学の研究に従事．ACM 上級会員（Senior Member），IEEE-CS，IEEE-SMC，IEEE 各会員．