公立鳥取環境大学紀要
第14号（2016. 3）pp.95-103

１．はじめに

参考文献４）および５）において、当時開発されていた食品産業向け商品処理装置のソフトウェア更新システムの検証事例が報告されている。そのシステムは工場内のネットワーク中継装置を介して工場内の商品処理装置が工場外のネットワーク上にある管理サーバに接続し、更新ソフトウェアが管理サーバから商品処理装置へダウンロードされるということを実現するものである。また、その検証事例における検証手法は形式手法と呼ばれる数理論理学を基礎とした幾つかの検証手法のうち、BAN 論理に基づく定理証明と LTL モデル検査が用いられている。ここで、定理証明とはシステムに与えられた条件・性質から安全性など検証したい性質の数理論理学的な証明可能性を検討するという手法である。モデル検査はシステムの振る舞いを数理論理学的モデルとして表現し、その振る舞いを網羅的に探索することでそのモデルが検証したい性質を満たすか否かを確認するという手法である。BAN 論理は信念の論理に分類されるものであり、プロトコルの安全性検証のために提案された論理推論体系としては最も早期のものの１つである（参考文献１））。BAN 論理が安全性などの性質について形式的な分析を可能にしたことや、また、それによる認証に関する安全性証明の形式化は後のプロトコルの安全性検証のための論理の研究に大きな影響を与えたが、一方、その無矛盾

盾

した状態遷移図および S4におけるモデルと公理系において脆弱性発見および修正を行った。つまり、検証対象そのものまで立ち戻ることなく検証対象からの抽象物において検証対象の脆弱性に関わる本質的な部分のみの考察によってそれらを行った。また、BAN 論理において証明された性質について、S4という無矛盾性の証明および意味論が確立された論理体系を用いてその性質を証明することにより、その証明と結果の信頼性を高めた。

本研究で用いた S4は古典命題論理に次のK、T、４と呼ばれる公理および必然化の規則と呼ばれる推論規則を加えたものである。

$$K：\Box(A\rightarrow B)\rightarrow(\Box A\rightarrow\Box B).$$

$$T：A\rightarrow A.\quad 4：\Box A\rightarrow\Box\Box A.$$

$$\frac{A}{\Box A}$$

これらを加えることで、論理式□A は「その時点を含む以降の時点においてA」という解釈ができる。また、◇A は¬□¬A の略記として与えることができ、それは「その時点を含む未来のある時点においてA」という解釈ができる。そして、論理式Aは「１つの時点においてA」という解釈となる。実際、S4のクリプキ意味論は半順序関係（反射的かつ推移的関係）を持つ構造に対して完全である（S4においてAが証明可能であるとき、かつそのときのみ任意の半順序構造においてAが恒真である）ことが示される。本研究において用いる体系について、構文論は参考文献３）および６）において与えているシーケント計算の体系を用いる。また、意味論は参考文献２）および６）において与えられているクリプキ意味論を用いる。

２．ソフトウェア更新システム

図１は対象となるソフトウェア更新システムの概念図である。

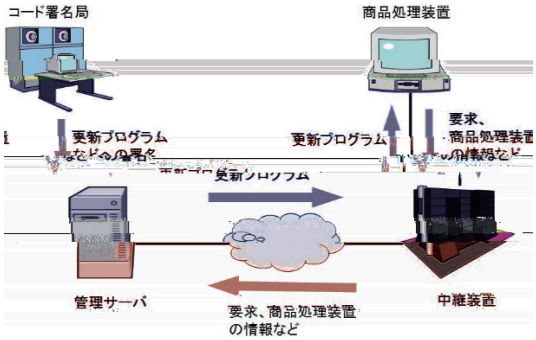

図１　ソフトウェア更新システム概要

このシステムは商品処理装置、中継装置、管理サーバ、コード署名局の４つのエージェントからなる。商品処理装置は工場内にある中継装置（プロキシ）を経由してインターネットに接続し、管理サーバに INF ファイル（このシステムでは、現在のソフトウェアの版情報などを記載したもののこと）や、更新要求を送信する。管理サーバはコード署名局から認証を得た更新 INF ファイルおよび更新プログラムを商品処理装置から受信した INF ファイルを基に商品処理装置へ向けて送信する。その際、それらのファイルはインターネットを介して、中継装置を経由して商品処理装置に受信される。ここで、中継装置は商品処理装置と管理サーバの間にあって通信の受け渡しを行うのだが、例えば、商品処理装置から送信された INF ファイルや管理サーバから送信された更新 INF ファイルおよび更新プログラムを記憶して置き、商品処理装置に対しては管理サーバの代わりに更新 INF ファイルと更新プログラムの送信を行い、管理サーバに対しては商品処理装置の代わりに更新 INF ファイルと更新プログラムの要求を送信し、それらを受信するという役割を持つ。

図２は商品処理装置、中継装置、管理サーバという３つのエージェントの通信プロトコルの仕様である。商品処理装置が中継装置に向けて INF ファイルを送信し、中継処理装置がそれを受信した後、管理サーバに送信する。その後、更新プログラムがある場合、中継装置に対して更新ありとの回答を送信し、中継装置はそれを商品処理装置に送信する。そして、中継装置は商品処理装置の代わりに更新 INF ファイルおよび更新プログラムの要求を管理サーバに対して送信する。管理サーバは中継装置から要求を受信した後、更新 INF ファイルと更新プログラムを中継装置に送信する。その後、商品処理装置から更新 INF ファイルおよび更新プログラムの要求を中継装置が受信した場合、中継処理装置はそれらを商

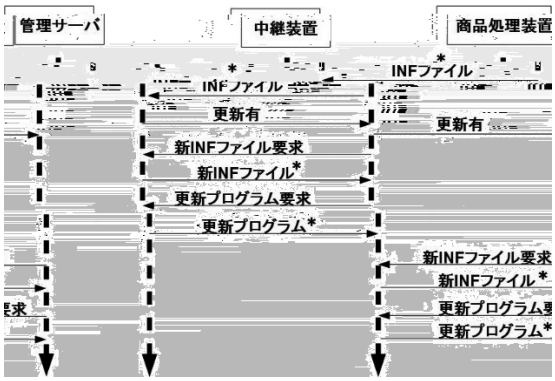

図２　プロトコル仕様

品処理装置に送信する。

図２のプロトコルはインターネット等を介しての通信は行わないコード署名局は省かれている。

図２における通信は実際には共通鍵暗号系による暗号通信が用いられる。また商品処理装置は INF ファイルを送信する際、公開鍵暗号系を用いた署名を行う。管理サーバも更新 INF ファイルと更新プログラムに対して署名を行い、これによって、受信したファイルの送信元の確認を可能にしている。

参考文献４）と５）では、このシステムのプロトコルに従って商品処理装置が２つのファイルの更新を要求したとき最新版を得ることができるかということについて検証を行っている。その際、そこでは検証対象のプロトコルをその検証内容に関係する部分を抽出して定理証明およびモデル検査を行っている。そこで抽出されたものは図２における（＊）の箇所である。それらはその検証項目に関係する INF ファイルおよび更新ファイルの送信の箇所である。また、先行研究では暗号通信に関して共通鍵暗号による通信は正常に機能していることを前提としており、その性質は検証の対象から除外している。一方、ファイルに付された署名については検証の対象としている。本研究では、この抽象化された対象の状態遷移図の作成から行う。

３．状態遷移図

状態遷移図とは、状態と呼ばれる点と、それらを結ぶ有向辺からなる図的表現である。出発点となる状態があり、そこから有向辺によって結ばれる状態へと遷移する。ただし、有向辺にはガードと呼ばれる含意命題が付随しており、その前条件を満たしたときのみその遷移が可能となる。前条件を満たし状態が遷移したとき、その遷移先の状態においてその後条件が成り立つ（状態遷移図の詳細については参考文献７）１章１節など)。

前節の抽象化された対象を状態遷移図で表すために図２における各要素を表す記号や述語を定める。ここで、INF ファイルおよびプログラムの版数は簡単のためそれぞれ３つとする。

- INF_i：商品処理装置にインストールされているファイルの版情報などを記載したファイル。ここでは３つ用意されているものとし、i＝1, 2, 3とする。
- F_i：プログラム。i＝1, 2, 3。INF_1、INF_2、INF_3のそれぞれに対応したソフトウェア。
- M：管理サーバ。P：中継装置。C：商品処理装置。
- send（A, B, F）：エージェントＡからエージェントＢへのファイルＦの送信を表す述語。

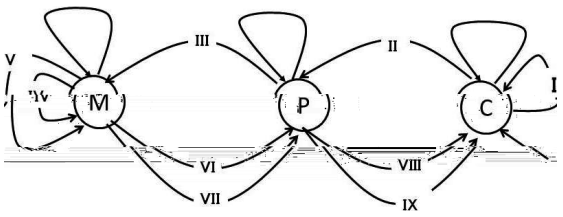

図３　状態遷移図

- receive（A, F）：エージェントＡによるＦの受信を表す述語。
- sign（A, F）：ファイルＦがエージェントＡに署名されていることを表す述語。

上記の図３の丸で囲まれたM、P、Cは状態遷移図における状態である。ここでは、システムにおける３つのエージェントを状態としている。状態から他の状態への遷移を表す矢印は対応するエージェントからエージェントへのファイルの送信に対応する。以下、図３の各遷移におけるガード（Ⅰ）から（Ⅸ）の詳細を述べる。

（Ⅰ）send(C, P, INF_i) ∧ sign(C, INF_i)（i =1, 2）

（Ⅱ）send(C, P, INF_i) ∧ sign(C, INF_i)
→ receive(P, INF_i)（i =1, 2）

（Ⅲ）receive(P, INF_i) ∧ sign(C, INF_i)
→ receive(M, INF_i)（i =1, 2）

（Ⅳ）receive(M, INF_i) ∧ sign(C, INF_i)
→ send(M, P, INF_{i+1}) ∧ sign(M, INF_{i+1}) (i =1, 2)

（Ⅴ）send(M, P, INF_{i+1})
→ send(M, P, F_{i+1}) ∧ sign(M, F_{i+1})（i =1, 2）

（Ⅵ）send(M, P, INF_{i+1}) ∧ sign(M, INF_{i+1})
→ receive(P, INF_{i+1})（i = 1, 2）

（Ⅶ）send(M, P, F_{i+1}) ∧ sign(M, F_{i+1})
→ receive(P, F_{i+1})（i = 1, 2）

（Ⅷ）receive(P, INF_{i+1}) ∧ sign(M, INF_{i+1})
→ receive(C, INF_{i+1})（i = 1, 2）

（Ⅸ）received(P, F_{i+1}) ∧ sign(M, F_{i+1})
→ receive(C, F_{i+1})（i = 1, 2）

この状態遷移図が図２のシーケンスチャートの通信内容を表現していることは、そのシーケンスチャートの各通信と状態遷移図における各状態遷移が対応していることを示すことで確認できる。ただし、状態遷移図において、中継装置Ｐの送信に関するガードが含まれていない。これは、中継装置が受け取って渡すという機能だけのものであることから、Ｐによる送信は省略してガード（Ⅲ）、（Ⅶ）、（Ⅷ）はＰがファイルを受信したらそのファイルをそのまま管理サーバMまたは商品処理装置Ｃが受信するということにしている。

（Ⅰ）について、前提条件無しで send(C, P, INF_i) と sign(C, INF_i) が成り立つ場合があることを表している。また、この状態遷移図の各状態には自分自身の状態に戻ってくる矢印が記載されている。これは図２などには明示されていないことであるが、実際にプロトコルにそのような冗長性を与え得るという観点と、S4のモデルを与えるための便宜を図る観点からそのようにしている。

４．モデル検査

与えた状態遷移図においてＣは適切な更新プログラムを受信できない振る舞いを探索する。ここで、様々な状態遷移の仕方のうち、悪意のある者にＰが乗っ取られた場合を考える。例えば、ＰはＣから受信した INF ファイルをそのまま管理サーバＭに送信せず、以前Ｃから受信した古い INF ファイルをＭに送信するという状況を考える。

１．既にＰが INF_1を持っているとする。
２．Ｃが INF_2をＰに送信する。
３．Ｐは INF_2ではなく、INF_1をＭに送信する。
４．Ｍは INF_1を受信し、INF_2および F_2をＰに送信する。
５．Ｐは既に持っている INF_2と F_2を受信する。
６．Ｐは INF_2と F_2をＣに送信する。
７．Ｃは INF_2と F_2を受信する。

このような振る舞いを図３の状態遷移図は許容することを確認することができる。このことの検査を行うために、その検証項目を表す論理式を与える。

$$\Box(\mathrm{send}(C, P, INF_2) \wedge \mathrm{sign}(C, INF_2) \rightarrow \Diamond \mathrm{receive}(C, INF_3))$$

この論理式において、Ｃが INF_2に署名しそれを送信すれば、INF_2に対して適切な更新 INF ファイルである INF_3を受信するということを表す。

ここで、S4のクリプキ意味論を与える。S4フレームとは、空でない半順序集合のことである。空でない集合Ｓとその上の半順序関係Ｒに対して、本稿では S4フレームの元を時点または可能世界と呼ぶ。与えられた時点ａと論理式Ａに対して、「ａにおいてＡ」であるとき、

$$a \models A$$

と表記する。次に、すべての原子命題に対して各時点における真偽が定義されたとき、原子命題から作られる各論理式の真偽は論理式の構成に関する帰納法により定義される。特に、

$a \models \Box A \overset{\text{定義}}{\Leftrightarrow}$ 任意の時点ｂに対して、aRb ならばｂ⊨Ａ

である。さらに、◇Ａは¬□¬Ａの略記であることから、

$a \models \Diamond A \Leftrightarrow$ ある時点が存在して、aRb かつｂ⊨Ａ

が成り立つ。S4フレームにおいて各原子論理の真偽が定義されたものを S4モデルと呼ぶ。与えられた S4モデルと論理式に対して、その論理式がそのモデルのすべての時点において真であるとき、その論理式はそのモデルにおいて真であるという。また、与えられた S4フレームと論理式に対して、任意の原子論理式の真偽の定義の仕方による S4モデルにおいてその論理式が真である場合、その論理式はその S4フレームにおいて恒真であるという。

前述の１～５の振る舞いを S4モデルとして表現する。集合について、図３の状態遷移図において前述の１～７の振る舞いを表す遷移の実行の順番を選ぶ。半順序関係について、遷移実行の順番の前後関係を選ぶ。そして、Ｐが既に INF_1を受信しそれを保持していることを表現するため、すべての時点において receive(P, INF_1) と sign(C, INF_1) が成り立つものとする。以下がその S4モデルの詳細である。

① send(C, P, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
② receive(P, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
③ receive(M, INF_1),
receive(P, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
④ send(M, P, INF_2), sign(M, INF_2),
receive(P, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
⑤ send(M, P, F_2), sign(M, F_2),
receive(P, INF_2), sign(M, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
⑥ receive(P, INF_2), sign(M, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
⑦ receive(P, F_2), sign(M, F_2),
receive(P, INF_2), sign(M, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓
⑧ receive(C, INF_2), sign(M, INF_2),

receive(P, INF_2), sign(M, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)
↓

⑨ receive(C, F_2), sign(M, F_2),
receive(C, INF_2), sign(M, INF_2),
receive(P, INF_2), sign(M, INF_2), sign(C, INF_2),
receive(P, INF_1), sign(C, INF_1)

このS4モデルについて、各時点において真である原子命題が記述されている。署名は一度なされたら常にそれがどの状態においても常に保持されることから、署名を表す原子命題 sign はある時点で真である場合、それ以降のすべての時点で真であるため、上記の S4モデルのある時点で現れている sign はそれ以降の時点においても現れている。③ではPが INF_2ではなく INF_1をMに送信した状況を示す。④ではMが INF_1に対する更新ファイルとして INF_2に署名してそれをPに送信している。

さて、この S4モデルにおいて、

$$①\models\Box\begin{pmatrix}\mathrm{send(C,\ P,\ INF_2)}\ \wedge\ \mathrm{sign(C,\ INF_2)}\\ \rightarrow\Diamond\mathrm{receive(C,\ INF_3)}\end{pmatrix}$$

は成立しない。実際、

$$①\models\mathrm{send(C,\ P,\ INF_2)}\ \wedge\ \mathrm{sign(C,\ INF_2)}$$

が成り立つがことは明らかであるが、receive(C, INF_3) はどの時点においても偽であることから、

$$①\models\Diamond\mathrm{receive(C,\ INF_3)}$$

は成り立たない。したがって、検証項目の論理式はこのモデルにおいて偽である。

同様に、Cが INF_2に署名しそれを送信すれば INF_2に対して適切な更新プログラムである F_3を受信するということを表す次の論理式も、この S4モデルによって偽になることを示すことができる。

$$\Box\begin{pmatrix}\mathrm{send(C,\ P,\ INF_2)}\ \wedge\ \mathrm{sign(C,\ INF_2)}\\ \rightarrow\Diamond\mathrm{receive(C,\ F_3)}\end{pmatrix}$$

5．形式化

脆弱性の発見としては3節で述べたことで十分であるが、さらにここで与えた2つの論理式がこのプロトコルにおいて証明できないということまで述べる。そのために、プロトコルを表現する公理系を与え、ここで与えたモデルがその論理体系における公理を満たすということを確認する。

公理系は図3の状態遷移図のガードを基に与える。

〔公理系1〕

(1) $\Box(\mathrm{sign(P,}\ X)\ \rightarrow\Box\mathrm{sign(P,}\ X))$
$(X\in\{\mathrm{INF_1,\ INF_2,\ INF_3,\ N_1,\ N_2,\ N_3}\})$

(2) $\Box(\mathrm{sign(M,}\ X)\ \rightarrow\Box\mathrm{sign(M,}\ X))$
$(X\in\{\mathrm{INF_1,\ INF_2,\ INF_3,\ F_2,\ F_3,}\})$

(3) $\Box\begin{pmatrix}\mathrm{send(C,\ P,\ INF}_i)\ \wedge\ \mathrm{sign(C,\ INF}_i)\\ \rightarrow\Diamond\mathrm{receive(P,\ INF}_i)\end{pmatrix}$
$(i=1,\ 2)$

(4) $\Box\begin{pmatrix}\mathrm{receive(P,\ INF}_i)\ \wedge\ \mathrm{sign(C,\ INF}_i)\\ \rightarrow\Diamond\mathrm{receive(M,\ INF}_i)\end{pmatrix}$
$(i=1,\ 2)$

(5) $\Box\begin{pmatrix}\mathrm{receive(M,\ INF}_i)\ \wedge\ \mathrm{sign(C,\ INF}_i)\\ \rightarrow\Diamond\mathrm{send(M,\ P,\ INF}_{i+1})\ \wedge\ \mathrm{sign(M,\ INF}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

(6) $\Box\begin{pmatrix}\mathrm{send(M,\ P,\ INF}_{i+1})\\ \rightarrow\Diamond(\mathrm{send(M,\ P,\ F}_{i+1})\ \wedge\ \mathrm{sign(M,\ F}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

(7) $\Box\begin{pmatrix}\mathrm{send(M,\ P,\ INF}_{i+1})\ \wedge\ \mathrm{sign(M,\ INF}_{i+1})\\ \rightarrow\Diamond\mathrm{receive(P,\ INF}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

(8) $\Box\begin{pmatrix}\mathrm{send(M,\ P,\ F}_{i+1})\ \wedge\ \mathrm{sign(M,\ F}_{i+1})\\ \rightarrow\Diamond\mathrm{receive(P,\ F}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

(9) $\Box\begin{pmatrix}\mathrm{receive(P,\ INF}_{i+1})\ \wedge\ \mathrm{sign(M,\ INF}_{i+1})\\ \rightarrow\Diamond\mathrm{receive(C,\ INF}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

(10) $\Box\begin{pmatrix}\mathrm{receive(P,\ F}_{i+1})\ \wedge\ \mathrm{sign(M,\ F}_{i+1})\\ \rightarrow\Diamond\mathrm{receive(C,\ F}_{i+1})\end{pmatrix}$
$(i=1,\ 2)$

ここで、各公理は $\Box(A\rightarrow\Diamond B)$ という形をしている。これは「Aをトリガーとして Bが生じる」ということが常に成り立つということを表している。公理系1の(1)について、1度なされた署名はその後常に成り立ち続けることを表している。

公理系1の各論理式が3節で与えた S4モデルにおいて成り立つことは容易に確認できる。例えば、公理系1の(1)について、sign(P, INF_1)、sign(P, INF_2) は①から⑤のいずれにおいても成り立っており、それ以外は成り立っていない。つまり、sign(A, X) が成り立っていない箇所においては常に□sign(A, X) が成り立っていないので、公理系1の(1)は①から⑤のいずれにおいても成り立つ。公理系1の他の公理も同様にその S4モデルで成り立つことが確認できる。

S4の健全性（与えられた論理式が S4の形式体系において証明可能ならば任意の S4フレームにおいてその論

理式は恒真）より、与えられた論理式がS4に公理系１を加えた形式体系において証明可能ならば公理系１の各論理式を真とする任意のS4モデルにおいてその論理式は真となる。したがって、次の２つの論理式

1　$\Box(\mathrm{send(C, P, INF_2)} \wedge \mathrm{sign(C, INF_2)} \rightarrow \Diamond \mathrm{receive(C, INF_3)})$

2　$\Box(\mathrm{send(C, P, INF_2)} \wedge \mathrm{sign(C, INF_2)} \rightarrow \Diamond \mathrm{receive(C, F_3)})$

はS4に公理系１を加えた形式体系において証明不可能であることが導かれる。

６．常に適切な更新ソフトウェアを得るための修正

古いINFファイルにすり替えられることを防ぐために、商品処理装置はINFファイルと共に自身がランダムに生成した文字列（これをナンスと呼ぶ）を送信する形に修正することを参考文献４）では提案している。この節では公理系１をナンスの署名と送信を考慮した次の修正版を与える。そのために、新たに記号および述語を定義する。

— N_i：ナンス。i =1, 2, 3。

— fresh(N_i)：ナンス N_i は新規に生成されたことを表す述語。i =1, 2, 3。

〔公理系２〕

(1)　$\Box(\mathrm{sign(P}, X) \rightarrow \Box \mathrm{sign(P}, X))$
　　($X \in \{\mathrm{INF_1}, \mathrm{INF_2}, \mathrm{INF_3}, \mathrm{N_1}, \mathrm{N_2}, \mathrm{N_3}\}$)

(2)　$\Box(\mathrm{sign(M}, X) \rightarrow \Box \mathrm{sign(M}, X))$
　　($X \in \{\mathrm{INF_1}, \mathrm{INF_2}, \mathrm{INF_3}, \mathrm{F_2}, \mathrm{F_3}\}$)

(3)　$\Box(\neg \mathrm{fresh(X)} \rightarrow \Box \neg \mathrm{fresh(X)})$
　　($X \in \{\mathrm{N_1}, \mathrm{N_2}, \mathrm{N_3}\}$)

(4)　$\Box\left(\begin{array}{c}\mathrm{send(C, P, INF}_i\mathrm{)} \wedge \mathrm{sign(C, INF}_i\mathrm{)} \\ \wedge \mathrm{send(C, P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(P, INF}_i\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i = 1, 2)

(5)　$\Box\left(\begin{array}{c}\mathrm{receive(P, INF}_i\mathrm{)} \wedge \mathrm{sign(C, INF}_i\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(M, INF}_i\mathrm{)} \\ \wedge \mathrm{receive(M, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i = 1, 2)

(6)　$\Box\left(\begin{array}{c}\mathrm{receive(M, INF}_i\mathrm{)} \wedge \mathrm{sign(C, INF}_i\mathrm{)} \\ \wedge \mathrm{receive(M, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{send(M, P, INF}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{send(M, P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i = 1, 2)

(7)　$\Box\left(\begin{array}{c}(\mathrm{send(M, P, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{send(M, P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{send(M, P, F}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, F}_{i+1}\mathrm{)} \\ \wedge \mathrm{send(M, P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i = 1, 2)

(8)　$\Box\left(\begin{array}{c}\mathrm{send(M, P, INF}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{send(M, P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(P, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i = 1, 2)

(9)　$\Box\left(\begin{array}{c}\mathrm{send(M, P, F}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, F}_{i+1}\mathrm{)} \\ \wedge \mathrm{send(M, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(P, F}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i=1, 2)

(10)　$\Box\left(\begin{array}{c}\mathrm{received(P, INF}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(C, INF}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(C, N}_i\mathrm{)} \wedge \neg\mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i=1, 2)

(11)　$\Box\left(\begin{array}{c}\mathrm{received(P, F}_{i+1}\mathrm{)} \wedge \mathrm{sign(M, F}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(P, N}_i\mathrm{)} \wedge \mathrm{fresh(N}_i\mathrm{)} \wedge \mathrm{sign(C, N}_i\mathrm{)} \\ \rightarrow\Diamond\left(\begin{array}{c}\mathrm{receive(C, F}_{i+1}\mathrm{)} \\ \wedge \mathrm{receive(C, N}_i\mathrm{)} \wedge \neg\mathrm{fresh(N}_i\mathrm{)}\end{array}\right)\end{array}\right)$
　　(i=1, 2)

ここで、ナンスNは商品処理装置Cが生成するものであり、それが新たに生成したものであることが確認できる場合においてファイルの送受信が可能であるものとしている。そして、その新規性は更新ファイルをCが受信した時点で否定される。公理系２の(3)は新規性が否定されたら以降は常にそのナンスは新規ではないということを表している。公理系２の(11)はその新規性の終了時点を示している。

CからCの署名付き $\mathrm{INF_2}$ および署名付きナンス $\mathrm{N_2}$ を管理サーバMに向けて送信すれば、Cはいずれ更新ファイル $\mathrm{INF_3}$ および $\mathrm{F_3}$ を受信するという検証項目を次の論理式で表す。

1　$\Box\left(\begin{array}{c}\mathrm{send(C, INF_2)} \wedge \mathrm{sign(C, INF_2)} \\ \wedge \mathrm{send(C, N_2)} \wedge \mathrm{fresh(N_2)} \wedge \mathrm{sign(C, N_2)} \\ \rightarrow\Diamond\mathrm{receive(C, INF_3)}\end{array}\right)$

2　$\Box\left(\begin{array}{c}\mathrm{send(C, INF_2)} \wedge \mathrm{sign(C, INF_2)} \\ \wedge \mathrm{send(C, N_2)} \wedge \mathrm{fresh(N_2)} \wedge \mathrm{sign(C, N_2)} \\ \rightarrow\Diamond\mathrm{receive(C, F_3)}\end{array}\right)$

これらの論理式はS4に公理系２を加えた体系において証明可能である（詳細は付記の証明図を参照せよ）。

すなわち、ナンスの導入によって適切な更新がなされることが示される。

７．まとめと課題

本稿における研究では、LTL モデル検査による脆弱性解析と BAN 論理による修正の妥当性に対する定理証明をそれぞれ S4に基づくモデル検査および定理証明よって行った。特に、検証対象に立ち戻らずに S4モデルと公理系に基づいて脆弱性発見と修正を行う事例を与えた。

ところで、基本的に定理証明とモデル検査を両方実施することが望ましく、さらに理論的にはそれぞれ同一の論理に基づく方がより整合性のある結果と分析が得られと思われる。その実施手順としては次のようになると考えられる。

１．適当な記述言語（仕様記述言語、状態遷移図など）による対象の記述。
２．既存のツールによるモデル検査と定理証明の実施。
３．S4や古典述語論理など、理論的に既に明らかになっている論理によるモデル検査（充足可能性判定も含む）と定理証明。

しかし、定理証明とモデル検査の両方を実施することも一般的にはコストが大きく、ましてや同一の論理に基づくものにすることはさらにコストを要する。以上を踏まえると、上記１から着手し、必要に応じて２、３と進めることにならざるを得ない。しかし、１の工程のみ実施するにしても、２、３への視野を持って行うことが検証対象への深い理解につながると思われる。

謝辞

査読者の有用なコメントに感謝を申し上げます。なお、本研究は公立鳥取環境大学特別研究費の助成を受けたものです。

参考文献

１）Burrows, M., Abadi, M., & Needham, R.（1990）. A logic of authentication. DEC SRC Research Report.
２）Goldblatt, R.（1992）. Logics of time and computation second edition. CSLI.
３）Troelstra, A., & Schwichtenberg, H.（2000）. Basic Proof theory, second edtion. Cambrige University Press.
４）吉田聡・山形頼之（2007）「ソフトウェア更新システムプロトコルの BAN logic による安全性検証」, 独立行政法人産業技術総合研究所システム検証研究センター算譜科学研究速報.
５）山形頼之・斎藤正也（2007）「ソフトウェア更新システムのモデル検査によるセキュリティ」, 独立行政法人産業技術総合研究所システム検証研究センター算譜科学研究速報.
６）小野寛晰（1994）「情報科学における論理」, 日本評論社.
７）中島震（2008）「SPIN モデル検査―検証モデリング技法―」, 近代科学社.
８）長谷部浩二・BANA Gergei・岡田光弘（2010）「セキュリティプロトコルの論理的検証法」, 萩谷昌巳・塚田恭章（著）『数理的技法による情報セキュリティ』, 共立出版, pp. 185-202.

（受付日2015年８月28日　受理日2015年11月11日）

付記：証明図

次の論理式が S4および公理系２の下で証明可能であることを以降に示す。

$$1 \quad \Box\left(\begin{array}{c}\mathrm{send(C, INF_2)} \wedge \mathrm{sign(C, INF_2)} \\ \wedge\ \mathrm{send(C, N_2)} \wedge \mathrm{fresh(N_2)} \wedge \mathrm{sign(C, N_2)} \\ \rightarrow \Diamond \mathrm{receive(C, INF_3)}\end{array}\right)$$

なお、次の論理式に対する証明図は以降に示すものと同様であるため、ここでは割愛する。

$$2 \quad \Box\left(\begin{array}{c}\mathrm{send(C, INF_2)} \wedge \mathrm{sign(C, INF_2)} \\ \wedge\ \mathrm{send(C, N_2)} \wedge \mathrm{fresh(N_2)} \wedge \mathrm{sign(C, N_2)} \\ \rightarrow \Diamond \mathrm{receive(C, F_3)}\end{array}\right)$$

ところで、以下に与える証明図には明らかと思われる推論を省略しており、ギャップが存在する個所を二重線で記している。以下がその省略のある推論である。

$$\frac{\Box\,\mathrm{sign}(A,X),\Gamma \Longrightarrow B}{\mathrm{sign}(A,X),\Gamma \Longrightarrow B}\ (1) \qquad \frac{\Longrightarrow \Box\wedge\Gamma\to A}{\Gamma \Longrightarrow A}\ (2) \qquad \frac{\Gamma \Longrightarrow \Diamond\Diamond A}{\Gamma \Longrightarrow \Diamond A}\ (3)$$

$$\frac{A \Longrightarrow B}{\Diamond A \Longrightarrow \Diamond B}\ (4) \qquad \frac{\Gamma \Longrightarrow A}{\Box\Gamma \Longrightarrow \Box A}\ (\mathrm{box}) \qquad \frac{\Gamma \Longrightarrow A,\Delta \qquad A,\Pi \Longrightarrow \Sigma}{\Gamma,\ \ \Longrightarrow}\ (\mathrm{cut})$$

ここで、Γは論理式の集合を表し、∧Γはそこに含まれるすべての論理式を論理積∧によって結合した論理式を表す。⑴は公理系２の⑴からS4のシーケント計算の体系において可能な証明可能な推論である。⑵、⑶、⑷はS4のシーケント計算において証明可能な推論である。（box）について、古典命題論理のシーケント計算の体系にこの推論規則を加えた体系は古典命題論理のヒルベルト流体系に公理Kおよび必然化の規則を加えた体系と同値である。（cut）は命題論理の推論規則の１つである。以下、証明図を与える。

証明図 **6** の終式　　証明図 **5** の終式

証明図 7