National Institute of
Advanced Industrial Science
and Technology
AIST

AIST-PS-2007-008

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

山形頼之 斎藤正也

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
Programming Science
Technical Report

Research Center
for Verification and Semantics

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 発行
Published by Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology

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

山形頼之　　斎藤正也

2007 年 7 月 17 日

概要

近年のコンピューターネットワークの発達に従い、そのセキュリティの確保が改めて課題となっている。ネットワーク上でのセキュリティを確保するためには、まずシステムの設計自体が正しいことが必要である。したがって、設計上の誤りをあらかじめ発見する手法が必要とされる。形式手法は、設計を数学的にモデル化し、モデルが期待される性質を数学的に証明する。これにより、形式手法は原理的にはすべての誤りを検出できる。本研究では、株式会社イシダと共同で、形式手法を用いて、同社の大型商品処理装置内蔵プログラムの遠隔更新システムプロトタイプのセキュリティを検証した。まず BAN-logic を用いて、通信される情報がセキュリティを確保するのに十分であることを確かめた。次に、システムの詳細仕様書から作成したモデルを対象にモデル検査を行い、送受信された情報が正しく取り扱われているか検証した。本報告ではこの詳細仕様の検証を取り扱う。

1　導入

近年のコンピューターネットワークの発達に従い、そのセキュリティの確保が改めて課題となっている。ネットワーク上でのセキュリティを確保するためには、まずシステムの設計自体が正しいことが必要である。したがって、設計上の誤りをあらかじめ発見する手法が必要と考える。中でも、設計を数学的にモデル化し、モデルが期待される性質を数学的に証明する形式手法は、原理的にはすべての誤りを検出できるため重要である。

本研究では、株式会社イシダと共同で、同社の大型商品処理装置内蔵プログラムの遠隔更新システムのプロトタイプに形式的手法を適用した。検証の対象となったプロトタイプシステムは、同社顧客先に設置された大型商品処理装置について、その内蔵プログラムを更新する必要が生じたときに、同社に設置された管理サーバからネットワークを介して更新プログラムを送信することを目的としたものである。システムは商品処理装置、管理サーバ、および両者の通信を中継する中継装置 (プロキシー) から構成される。中継装置は各顧客工場に 1 台づつ設置されることを想定している。商品処理装置が誤った更新プログラムを読み込むと、生産ラインの停止や人命に関わる事故も予想される。したがって、正しい更新プログラムだけが読み込まれなくてはならない。また、顧客ごとにプログラムをカスタマイズすることが考えられるため、更新プログラムに顧客固有の秘密情報が含まれる可能性がある。このことから、更新プログラムを対象となる顧客以外が取得できないようにすることが好ましい。

ネットワークシステムの形式手法によるセキュリティ検証には複数の技法がある。これらは大別すると、セキュリティ検証に特化した論理を用いるもの (たとえば [1])、項書き換え系 (たとえば [2]) を用いる方法と、汎用の定理証明器 (たとえば [3]) やモデル検査器 (たとえば [4]) を用いる方法がある。

本研究ではまず BAN-logic[1] を用いて、通信される情報がセキュリティを確保するのに十分であることを

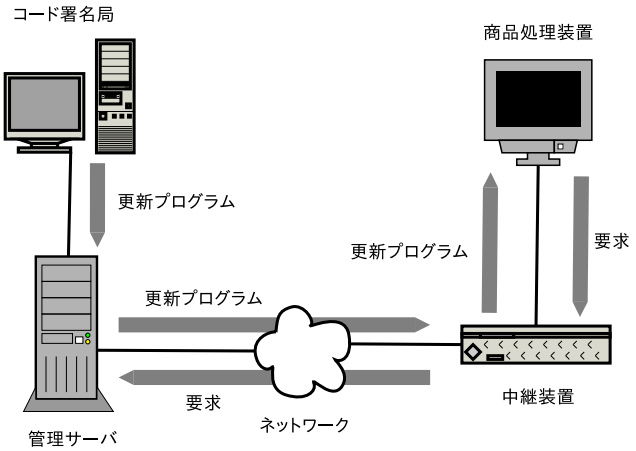

図 1　更新プログラムの流れ

確かめた [5]。次に、システムの詳細仕様書から作成したモデルを対象にモデル検査を行い、送受信された情報が正しく取り扱われているか検証した。本報告ではこの詳細仕様の検証を取り扱う。

検証は装置間の通信規約を表したシークエンスチャートとコマンド仕様書を対象とし、これを人手で Promela 言語 [6] に変換したうえ、Spin モデル検査器 [6] により検証を行った。検証項目は次の 2 点である。

1. 商品処理装置が取得した更新プログラムがその装置に対応した最新のものであること。(安全性)
2. 不正な商品処理装置が自己と異なる装置向けの更新プログラムを取得できないこと。(秘匿性)

これを LTL 式 [6] で表現することにより検証を行う。中継装置および商品処理装置は顧客先に設置されるため、秘密鍵の漏洩やクラッキングを受ける可能性が無視できない。そこで、中継装置、商品処理装置それぞれが信頼できない場合について検証項目を検討した。

この結果、最初の仕様レビューにより、不正な中継装置が更新プログラムを他の商品処理装置のものと偽って送信することができることが分かった。そこでセッションの始めに商品処理装置がランダムなセッション番号を生成して管理サーバに送信し、更新プログラムをこのセッション番号とともに署名して送信するよう改善した。Spin による検証では元の仕様の問題点を再現でき、また改良した仕様では問題が起きないことを示せた。

同様に仕様レビューにより、不正な商品処理装置が他の商品処理装置の更新プログラムを取得できることが分かった。この問題点を Spin による検証で再現することができた。

本報告の構成を述べる。まず 2 節では本システムの概要を紹介する。次に 3 節では Promela によるモデル化の方針を示す。4 節と 5 節では上記の 2 つの検証項目についての結果を示す。最後に 6 節で結論を述べる。

2　本システムの構成

1 節で述べたように本プロトタイプは、商品処理装置の更新プログラムを配布することが目的である。図 1 に更新プログラムの流れを簡略化して図示する。

更新プログラムは、まずコード署名局により電子署名がおこなわれ、管理サーバに置かれる。セッションは商品処理装置が要求を中継装置に送信することから開始される。中継装置は商品処理装置から要求を受信する

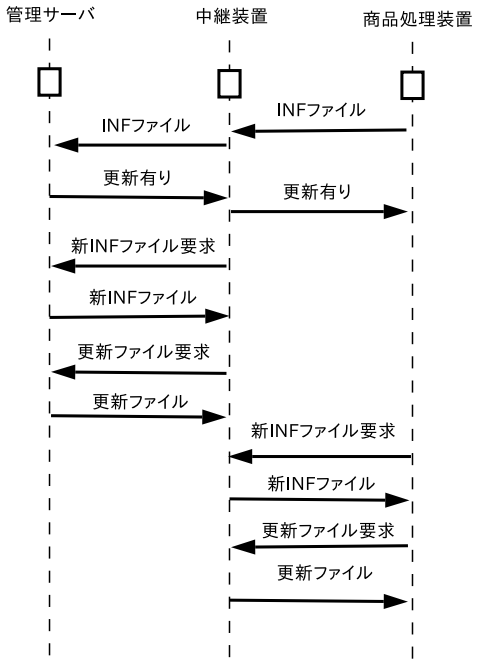

図 2　通信シークエンス

と、ネットワークを通して管理サーバに要求を送信する。管理サーバは要求を受けて、更新プログラムを中継装置へ送信する。中継装置は更新プログラムを受け取ると、これを商品処理装置に送信し、セッションが終了する。

商品処理装置が直接管理サーバに接続せず、中継装置が媒介するのは複数の理由がある。

1. 商品処理装置に能力の限界があるとき、電子署名やインターネット接続等を商品処理装置に代わって行う。
2. 商品処理装置が直接外部に接続しないことで、商品処理装置からの情報の漏洩や、商品処理装置への侵入を防止する。
3. 商品処理装置がどの工場に設置されているかを追跡する。

商品処理装置、中継装置、管理サーバ間の通信は SSL により保護される。PKI や証明書設計に付いてはここでは触れないが、接続時にはサーバ、クライアントともピアの固有番号および種別を判定することができるとする。ここで種別とは処品処理装置、中継装置、管理サーバ、コード署名局、各種 CA のいずれかである。

通信シーケンスを図 2 に簡略化して示す。商品処理装置は自己の機種やインストール済みプログラムを記述した INF ファイルとよばれるファイルを保持している。セッション開始時には、まず商品処理装置が、中継装置を経由して、INF ファイルを管理サーバに送信する。管理サーバは、INF ファイルから更新の必要性を判断し、更新の有無を中継装置に送信する。中継装置は更新の有無情報を商品処理装置に転送した後、更新がある場合には更新された INF ファイルを管理サーバに要求する。中継装置は管理サーバから新たな INF ファイルを受け取ると、その内容から更新しなければならないプログラムを判定し、管理サーバに要求する。一方、商品処理装置は中継装置から更新があることを知らされると、更新された INF ファイルを中継装置に要求する。中継装置は管理サーバから新 INF ファイルと更新プログラムを取得した後、新 INF ファイルを商品

処理装置に向けて送信する。商品処理装置は新 INF ファイルの内容から更新しなければならないプログラムを判定し、必要があれば操作画面に表示して、操作者に判断を求める。操作者の了承が得られると、商品処理装置は中継装置に更新プログラムの送信を求める。中継装置は全ての更新プログラムを管理サーバから受信した後、商品処理装置の要求に答え更新プログラムを送信する。中継装置は更新プログラムをキャッシュしておき、同じ更新プログラムが他の商品処理装置に必要になったときには、管理サーバに更新プログラムを要求せず、単にキャッシュを送信する。

3 モデル作成の手法

本研究では、プロトタイプの開発に用いられたメッセージシークエンスチャート及びメッセージ仕様書を Promela 言語によりモデル化し、Spin モデル検査器により検証を行った。本節では Promela によるモデル化の方法に付いて述べる。

モデル検査用のモデル作成については、検査したい性質に本質的に関わる部分のみを取り出した、できるだけ単純なモデルを作成し検証する方法と、元のシステムに忠実なモデルを作成し、必要に応じて抽象化を行う方法の 2 つが考えられる。本研究では最初に詳細なモデルを作成し、後から必要に応じて抽象化を行う方法をとった。機器の固有番号、プログラム名、プログラムバイナリ、バージョン番号、秘密鍵、公開鍵、IP アドレス、ホスト名、その他一般の文字列は 1 つのバイトとしてモデル化を行った。SSL に関連する操作などライブラリパッケージの機能は Promela のインライン関数として実現されている。また、秘密鍵による暗号化など特定の機器のみが可能な操作は同様にインライン関数として表現され、その関数を特定機器のモデルだけが呼び出すよう注意してモデル化を行った。

最後に、検証項目に無関係な変数や構造体のフィールドをモデルから削除することで、状態数を抑える抽象化を行った。抽象化の正当性は、目視による簡単な検討を行い確認した。

モデル化の過程では、仕様書に記述されているエラー処理のあいまい性が問題となった。仕様では、エラーが発生した際に ERRO メッセージを通信ピアに送信することになっていたが、ERRO メッセージ送受信時のエラーへの対処法や、中継装置がエラーを受け取った場合に他のピアへ中継するべきかどうかが定められていなかった。そこでモデル化に際してエラー処理の流れを定義し直した。具体的には次のようにエラー処理をモデル化した。

まず仕様書にあるエラーコード表からエラーを抽出し、モデルのどの箇所で発生し得るかを検討した。この検討結果に従い、エラーを発生する可能性のあるモデル中の箇所に、ランダムにエラーを発生させるコードを埋め込んだ。そしてエラーが発生した場合には機器の種類に応じた処理を行うこととした。

管理サーバおよび商品処理装置は次のようにエラー処理を行う。

自己がエラーを発生させた場合 エラー処理部にジャンプする。エラー処理部ではエラー内容をログに書き込み、ERRO を通信ピアに通知したのち終了する。通信ピアに ERRO を通知する際にエラーが発生した場合は、そのエラーを記録し、ERRO を再度送信することなく、そのままセッションを終了する。

ERRO **コマンドを受信した場合** ERRO コマンドを受信した時点で、エラー処理部にジャンプする。エラー処理部では、エラー内容をログに書き込んだ後、そのままセッションを終了する。

中継装置は、セッション毎に管理サーバへの通信を担当するスレッドと商品処理装置への通信を行うスレッドを生成するが、各スレッドは次のようにエラー処理を行う。

自己がエラーを発生させた場合　エラー処理部にジャンプする。エラー処理部ではエラー内容をログに書き込んだ後、スレッド間通信キューに ERRO コマンドを書き込む。その後、通信ピアに向けて ERRO コマンドを送信する。ERRO コマンドの送信に失敗した場合は、エラーをログに書き込んだ上、そのまま終了する。

スレッド間通信キューから ERRO を受け取った場合　エラー処理部にジャンプしたのち通信ピアに ERRO を転送する。ERRO の送信に失敗した場合は、エラーをログに書き込んだ上、そのまま終了する。

通信ピアから ERRO を受け取った場合　エラー処理部にジャンプしたのち、ログに記録する。さらに、スレッド間通信キューに受信した ERRO コマンドを書き込み、そのまま終了する。

ただし、通信ピアとのセッションが開始されていない段階では、ERRO の送信は行わない。

4　安全性についての検証

本節では不正な中継装置が存在していたとしても、商品処理装置が不正な更新プログラムを検出できるか否か検証した結果について述べる。検証すべき性質は次のいずれかが成り立つことである。これを性質 (A) と呼ぶ。

1. 商品処理装置が正規の更新プログラムを受信する。
2. エラーが生じる。
3. 商品処理装置が何も受信しない。

更新プログラムの受信はモデル上では商品処理装置のプログラムを格納する変数の変化として表現される。また、エラーの発生はエラーコードを格納する変数の変化として同様に表現される。

不正な中継装置は次のようにモデル上で表現される。不正な中継装置が生成するメッセージの種類や順序は正常の中継装置と同じである。同様に受信するメッセージの種類や順序も同じである。しかし不正な中継装置は生成するメッセージのパラメータをランダムに決定する。また不正な中継装置は INF ファイルや更新プログラムをキャッシュ中のものからランダムに選んで送信する。このようにモデル化することにより、多様な攻撃に対する耐性を検証することができる。

加えて不正な中継装置がメッセージの順序をランダムに選ぶモデルに対しても検証を試みたが、メモリ使用量の限界により検証を完了することができなかった。

本システムは任意の数の中継装置、商品処理装置が存在し得るが、モデル検査を行うためには装置の数には上限がなければならない。ここでは管理サーバ、不正な中継装置がそれぞれ 1 台づつあり、これに加えて商品処理装置が 2 台ある場合を想定した。2 台の商品処理装置が同時に更新を試み、正常に更新が行われればそれぞれ異なった更新プログラムを受け取ることとする。

プロトタイプ仕様の検討から、性質 (A) は満たされないことが明らかになった。商品処理装置は新たな INF ファイルの内容が予想できず、更新ファイルは中継装置から送信された INF ファイルの情報によってのみ知ることができる。よって中継装置が、本来送信すべきファイルではないファイルを、商品処理装置に送信したとしても、商品処理装置はこれを検出することができない。また、更新ファイルは電子署名されているが、署名された部分にはファイル名が含まれていない。従って、ファイル名を偽ることも可能である。この問題点をモデル検査によって再現することができた。

いずれもキャッシュにある他のバージョンまたは商品処理装置への更新プログラムや INF ファイルを、本来とは異なった商品処理装置に送信するものである。そこでセッション開始時に商品処理装置が乱数によりセッション ID を生成しておき、新 INF ファイルや更新ファイルを送信する際にこの ID とともに電子署名を行うよう改良した。この改良により上記のシナリオで性質 (A) が満たされることをモデル検査で検証できた。

5 秘匿性に関する検証

本節では、不正な商品処理装置を介して更新プログラムが漏洩する可能性に付いて、検討した結果を述べる。商品処理装置がクラッキングを受けた場合、更新プログラムもふくめ全データが漏洩することが予想され、これを防ぐことは不可能である。また、商品処理装置の認証用秘密鍵が漏洩し不正な機器に利用された場合、これを発見することは困難である。従ってクラッキングや秘密鍵の漏洩の可能性を想定した場合には、更新プログラムの秘匿性を保つことは困難である。

しかし、商品処理装置のクラッキングや認証用秘密鍵の漏洩があったとしても、他の商品処理装置の更新プログラムが漏洩しないように設計することで、被害を最小限にとどめることができる。したがて、不正な商品処理装置が自己以外の更新プログラムを取得できないことを要請し、検証を行った。この性質を性質 (B) と呼ぶことにする。

しかし、Promela によるモデルの作成段階で、性質 (B) は成り立たないことが分かった。管理サーバは商品処理装置の INF ファイルに基づいて、送信すべき更新プログラムを決定するが、INF ファイルは電子署名等の保護が行われておらず、偽造が容易だからである。この問題はモデル検査によっても再現できた。

6 結論

本研究では株式会社イシダが共同で、遠隔ソフトウェア更新システムのプロトタイプについて、その詳細仕様書を対象に、セキュリティ上の性質について検証を行った。まず仕様書レビューの段階で、不正な中継装置が存在するとき、誤った更新プログラムを商品処理装置が受理する可能性が明らかになった。また、モデル作成の過程で、不正な商品処理装置が他の商品処理装置の更新プログラムを取得できることが判明した。一方、モデル検査によっては新たな問題は発見されなかったが、上記の問題を再現することができた。モデル検査器によるエラーの再現は、研究参加者の間で問題点の理解を共有する上で有益であった。レビューおよびモデル検査結果に基づき、プロトタイプの改良を行った。

なお、本研究の成果は、製品の改善の機会を与え、安全性をより一層向上させるための示唆を与えたとイシダでは評価しているものの、製品への検証は、テストなど通常の手法に従って別途行なわれており、本研究での検証は、それに加えて製品の仕様書等について行なわれたものであること、本研究の結果だけからは、最終製品全体としての信頼性、安全性について言及できないことを強調しておきたい。

7 謝辞

本研究は株式会社イシダの協力のもとに行われた。ここに感謝したい。また、モデル検査に用いた Promela モデルは、高井利憲氏が作成したモデルを元に作成した。ここに感謝したい。

参考文献

[1] Michael Burrows, Martin Abadi, Roger Needham, A logic of Authentication, DEC SRC Research Preport 39, 1990.

[2] H. Ohsaki,T. Takai, ACTAS : A System Design for Associative and Commutative Tree Automata Theory, Electronic Notes in Theoretical Computer Science, 2005

[3] Giampaolo Bella, Fabio Massacci, Lawrence C. Paulson, The Verification of an Industrial Payment Protocol: The SET Purchase Phase, Proceedings of the 9th ACM conference on Computer and Communications Security, ACM Press, 2002

[4] John M. Mitchell, Mark Mithcell, Ulrich Stern, Automated Analysis of Cryptographic Protocols Using Murϕ, Proceedings of the 1997 IEEE Symposium on Security and Privacy, IEEE Computer Society, 1997

[5] 吉田聡, 山形頼之, ソフトウェアアップデートシステムプロトコルの BAN Logic による安全性検証, 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, 2007

[6] Gerard J. Holzmann, The Spin Model Checker, Addison Wesley, 2004

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
(算譜科学研究速報)
発行日 2007 年 7 月 17 日
編集・発行：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
同連絡先：〒563-8577 大阪府池田市緑丘 1-8-31
e-mail：informatics-inquiry@m.aist.go.jp

Security verification through model checking: A case study using software update system
July 17, 2007
Research Center for Verification and Semantics (CVS)
Ikeda Site
National Institute of Advanced Industrial Science and Technology (AIST)
1-8-31 Midorigaoka, Ikeda, Osaka, 563-8577, Japan
e-mail: informatics-inquiry@m.aist.go.jp

AIST01-J00022-71