平成２２年度研究開発成果概要書
「通信プロトコルとその実装の安全性評価に関する研究開発」
副題：「形式手法によるプロトコル実装の検証技術と
形式仕様に基づく網羅的ブラックボックス検査技術の開発」

（１）研究開発の目的

インターネット標準は現実の実装においては必ずしも仕様がその通りに反映されていない場合が多く、暗号通信プロトコルなどにおいて、その実装の誤りが安全性の脆弱性に繋がるケースが複数見受けられている。このような誤りはもちろん実装の作成者の注意不足に起因する場合も多いが、一方で誤りの一端が標準そのものにある場合もある。

本プロジェクトでは、形式的手法による通信プロトコルの規格の記述を元にその実装の様々な評価・検証手法を統一的に扱うことのできる手法および、ブラックボックスになっている実装においてもテストケースの網羅性などを保証し、仮想マシンモニタを利用した実行のトレース及びロールバックにより脆弱な実装を解析できる環境の開発を行う。

（２）研究開発期間

平成２２年度から平成２４年度（３年間）

（３）委託先企業

独立行政法人産業技術総合研究所＜幹事＞、株式会社レピダム

（４）研究開発予算（百万円）

| | |
|---|---|
| 平成２２年度 | ４１ |
| 平成２３年度 | ３９ |
| 平成２４年度 | ３８ |

（５）研究開発課題と担当

課題ア：形式的手法によるプロトコル実装の検証技術の開発

1.　形式検証のフレームワーク技術（産総研）

検証対象プログラムの形式モデルと形式仕様を記述するために、C 言語のサブセットを決定する。このサブセットに対して定理証明支援器 Coq 上で必要な型や演算子等を形式化し、形式検証のフレームワークを準備する。また、レピダムによって抽出された検証対象の TLS の部分プログラムに対して、インターネット標準（RFC 5246）との関係を明らかにしながら、Coq 上での形式仕様を記述し、形式検証を行う。

2.　形式検証ケーススタディーの作成（レピダム）

既存の TLS 実装を調査し、形式的検証の対象として適当な部分を産総研と共同で選定する。更に、選定部分を形式検証フレームワークのた

めに定義された C 言語のサブセットに移植することで形式モデルを作成する。また、形式モデルを元の TLS 実装に戻して動作させるための変換器など、必要な周辺ツールを作成する。

課題イ：網羅的テストケース生成による実装のブラックボックス解析技術の開発

イ―１）網羅的汎用テストジェネレータおよび実行環境

1.　網羅的プロトコル検査のためのプロトコル記述方式（産総研）

実験対象となる比較的単純なプロトコルを選定し、プロトコル記述のための中間言語の設計を行う。同言語の詳細設計とインタプリタの実装についてレピダムの作業を支援する。また、そのプロトコルの実装に対しファジングなどの改変した通信を生成するための方法について検討し、簡単なものについて同処理系に実装する。更に課題（イ―２）で今年度および翌年度以降に行う実行トレース・実行状態制御との連携について検討し、レピダム側の実装に反映する。

2.　網羅的プロトコル検査のための処理系の開発（レピダム）

産総研が選定した実験対象プロトコル・中間言語を実行可能にするインタプリタのプロトタイプを開発する。そのうえで、TCP や TLS など、より複雑なプロトコルを記述する際に必要な機能について検討し、可能な範囲で仕様記述に着手する。

イ―２）仮想マシンベースとしたコントロールフロー解析

1. ブラックボックス OS 内の実行トレースおよび解析が行える技術（産総研）

OS 自体を改変することなく、管理者権限がない OS でもその動作の解析が行えるようにする。開発は仮想マシンベースで行い、OS を特定する技術、その OS の構造理解、実行トレースの技術の設計、および一部の試作を行う。

（６）これまで得られた研究開発成果

（全体）　2 件　　（当該年度）2 件

| | | | |
|---|---|---|---|
| 特許出願 | 国内出願 | 0 | 0 |
| | 外国出願 | 0 | 0 |
| 外部発表 | 研究論文 | 0 | 0 |
| | 報道発表 | 0 | 0 |
| | その他研究発表 | 2 | 2 |
| | 展示会 | 0 | 0 |
| | 標準化提案 | 0 | 0 |

具体的な成果

1) プロトコル実装の形式検証フレームワークとケーススタディー（産総研とレピダムの共同成果）

現実的な通信プロトコルの実装を評価するための形式検証フレームワークのプロトタイプを構築し、ケーススタディーを開始した。具体的には、検証のための C 言語の形式モデルを定義し、形式モデルから C プログラムへの変換器を開発した。C 言語の形式モデルは十分な表現力を持ち、検証対象として選定した PolarSSL のパケット処理部分のコードを大きく変更すること無くモデル化できた。また、変換器を利用し、検証モデルを元のプログラムに戻して実行可能であることを確認した。今後はこのケーススタディーを進め、RFC に従うパケット処理コードとその形式証明を得ることを目指す。

2) 網羅的汎用テストジェネレータのためのプロトコル記述方式とリファレンス実装生成系（産総研とレピダムの共同成果）

プロトコル記述のための中間言語のプロトタイプをデザインし、Coq 上に実装した。また、この中間言語を用いて、RFC で定義された TLS プロトコルの一部を対象に記述実験を行った。この技術は、記述されたプロトコルの性質を形式的に検証可能とし、また、プロトコルを実行するリファレンス実装コードを生成可能とする。この中間言語のプロタイプは、TLS プロトコルの一部を記述するのに十分な表現力を持ち、その記述から生成されたリファレンス実装コードは、既存の TLS の実装と正しくやり取り可能なものである。

3) CPU の仮想記憶機能を監視したプロセストレース技術（産総研）

ブラックボックスの OS 内のプロトコル処理の実行トレースおよび解析のために、仮想マシンが提供する CPU の仮想記憶管理機能を監視することでユーザプロセスの切り替えを認識し、物理メモリ上の各プロセスのマッピングを解析できる技術を開発した。これは OS に非依存であり、ブラックボックスの OS に適用できる技術である。

（７）研究開発イメージ図

下記の図で研究課題ア）とイ）の関係を示す。

課題ア）において形式的手法によるプロトコル実装の検証技術の開発を行う。検証結果を元に課題イ）において網羅的テストケース生成技術の作成、および仮想マシンを活用してブラックボックス実装でも解析できる技術の開発を行う。