6Z-2

# サーバの配付コードをクライアントに動的実行させて サーバで動作検証できる動的コードの配付・実行・検証機構

佐々木　啓†　　脇田　知彦†　　福田　洋治‡　　毛利　公美††　白石 善明†　野口　亮司†††

名古屋工業大学†　　愛知教育大学‡　　岐阜大学††　　(株) 豊通シスコム†††

## 1. まえがき

クラウドサービスの高度化により，管理された社員用 PC を使って社内外からクラウドに接続する需要が増えた[1]．また，車載・路上機器による交通情報共有ネットワークが実現されつつある[2]．このような管理された機器により構成されるネットワークにおいては，クライアントソフトウェアのバージョンを統一したいという要望がある[3]（例：図１）．従来のアップデートはユーザがクライアントソフトウェアをアップデートする形式が多いが，この形式では全てのユーザが速やかに更新をするとは限らない．また更新しないユーザに対しアップデートの強制はできない．

本研究では管理された機器によるネットワーク内のクライアントソフトウェアを強制的にアップデートすることを目標としている．このとき，ユーザによるアップデートの妨害とアップデートするデータの偽造が懸念される．本稿では RPC と動的読み込みによる安全なアップデーター配付・実行機構を提案した．

## 2. 要求条件と基本設計

前章の目標を達成する要件を述べる．

【要件 1】配付を妨害されないこと

本研究ではユーザの妨害攻撃に，アップデーターのプログラムファイルを削除すること，アップデーターのプロセスを強制終了することを想定する．これらの妨害攻撃があっても正しく動作することが必要である．

【要件 2】配付コードが即時に実行されること

要件 1 を満たすにはアップデーターを実行時に配付する方法が考えられる．この時配付と実行に時間差があると配付コードがユーザに削除・書き換えられる危険が増す．そのため配付コードは即時に実行される必要がある．

【要件 3】配付コードが実行されたことを検証できること

本研究ではユーザの偽造攻撃に，アップデートするデータの偽造と配付コードの実行結果の偽造を想定する．これらを防ぐためには配付コードの実行をサーバから検証する必要がある．

【要件 4】多数のクライアントに対応できること

本システムは社内情報システムや ITS など数千以上のクライアントを持つネットワークが対象となる．よって多数のクライアントへ対応する必要がある．

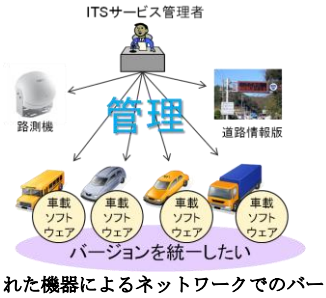


**図 1　管理された機器によるネットワークでのバージョン統一**
**例：車載・路上機器によるネットワーク**

Verifiable Distribution and Execution Mechanism of Server's Dynamic-code

† Kei SASAKI, Tomohiko WAKITA, Yoshiaki SHIRAISHI・Nagoya Institute of Technology

‡ Youji FUKUTA・Aichi University of Education

†† Masami MOHRI・Gifu University

††† Ryoji Noguchi・Toyotsu Syscom Corp.

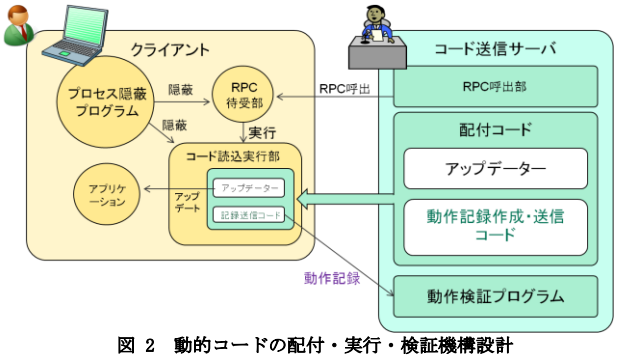


**図 2　動的コードの配付・実行・検証機構設計**

## 3. 提案機構の設計

RPC（Remote Procedure Call）[4]は別のアドレス空間にあるコードを実行する技術で，待受クライアントが起動している端末のライブラリを外部から起動できる．動的読み込み（遅延読み込み）[5]は実行中のプログラムが明示的にライブラリを読み込む技術であり，RPC とは違い呼び出した側の PC でプログラムが実行される．

提案機構の設計を図 2 に示す．また，手順を以下に示す．step1 開始時，クライアント側には RPC 待受部が立ち上がっている．　このとき，RPC 待受部のファイルとプロセス，コード読込実行部のファイルは隠蔽プログラム[6]によって隠蔽されている．

step 1.　サーバの RPC 呼出部がクライアントの RPC 待受部を呼出す
step 2.　RPC 待受部はコード読込実行部を実行する
step 3.　コード読込実行部は SSL 通信路[7]でサーバから配付コードをダウンロードし，電子署名を検証して実行する
step 4.　配付コード内のアップデーターを実行する
step 5.　配付コード内の動作記録作成・送信コードは配付コードの動作結果と，配付コード内に含まれるサーバが生成した乱数から動作記録を作成する
step 6.　動作記録作成・送信コードは動作記録をサーバの動作検証プログラムへ送信する
step 7.　動作検証プログラムは動作記録内の乱数を検証し，動作結果をログへ出力する

## 4. 提案機構の実装

言語は C#[8]，RPC ライブラリは.NET remoting Framework [9]，HTTP サーバは Apache2.2[10]で実装した．配付コードを動的に読み込む際の署名検証は厳密名によるアセンブリ署名[11]を利用する．

## 5. 安全性

提案手法では，以下の 2 通りの攻撃が想定される．

【妨害攻撃】

[手段 1] 配付コードの実行に必要なファイルの削除

[手順]

(1)　攻撃者は RPC 待受部またはコード読込実行部を削除する

[手段 2] 配付コードプロセスの強制終了

[手順]

(1)　アップデート実行時，攻撃者は配付コードのプロセス ID を取得する

(2)　攻撃者は配付コードのプロセスを強制終了する
【偽造攻撃】
[手段] 配付コードの偽造
[手順]
(1)　攻撃者は配付コードからファイル名を得る
(2)　攻撃者は配付コードのファイル名を持つ偽造配付コードを作成する
(3)　攻撃者は偽造した配付コードをコード読込実行部に読み込ませ，実行する
(4)　偽造のコード終了通知をサーバに送信し，配付コードが正常に終了したと偽装する

妨害攻撃は，隠蔽プログラムによりファイルとプロセスをユーザから隠蔽することで防ぐことができる．偽造攻撃は配付コードを SSL 通信路からダウンロードすることで手順(1)の攻撃者による配付コード取得を防ぎ，配付コードの電子署名を検証することで手順(3)の偽造配付コードの読込を防ぐ．また，配付コード内に含まれる乱数を検証することで手順(4)の正常終了の偽装を検知する．

## 6. 計測

提案機構が要件 2 と 4 を満たすことを確認するため，クライアント数の変化による(1)占有メモリの変化，(2)提案機構の動作時間の計測を行った．提案機構の動作時間とは 3 章の step1 から step7 の動作時間から配付コード内のアップデーターの実行時間を除いた時間である．計測用システムを図 3 に示す．計測用システムは提案機構に管理用 GUI・端末状態を保存するデータベース・計測用クライアントプロセスの立ち上げ機構を加えた構成となっている．5000 のクライアントに対し平均 380 ミリ秒で動作する（表 1）ため即時実行を満たす．サーバプログラムが占有するメモリは 1 クライアントにつき約 570KB である（表 1）ため，多くのクライアントに対応できる．よって提案機構は要件 2 と 4 を満たす．

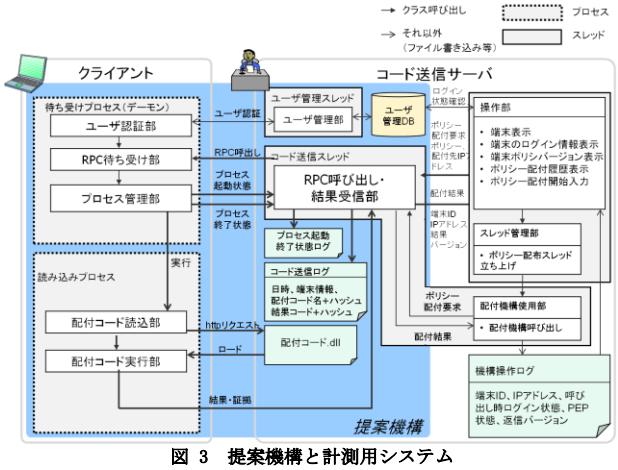


図 3　提案機構と計測用システム

表 1　計測結果

| クライアント数 | 100 | 500 | 1000 | 5000 |
|---|---|---|---|---|
| 動作時間[ms] | 365.28 | 346.29 | 344.77 | 380.25 |
| 占有メモリ[KB] | 429.58 | 463.16 | 486.90 | 566.35 |

表 2　比較評価

| | 運用支援 | アップデート管理 | 提案機構 |
|---|---|---|---|
| 要件 1 | × | × | ○ |
| 要件 2 | ○ | × | ○ |
| 要件 3 | ○ | ○ | ○ |
| 要件 4 | 少 | 多 | 多 |

## 7. 比較評価

PC 運用支援システム[12][13][14]，ソフトウェアアップデート管理サービス[15][16]と比較することで提案機構を評価した．PC 運用支援システムは組織内での PC 資産の統一的な運用管理を行うシステムである．ソフトウェアアップデート管理システムは自動で行われるソフトウェアの更新を組織ごとに管理する拡張サービスである．

これらは管理対象 PC にクライアントソフトウェアを導入し，管理サーバから監視・操作を行う構造である．しかし，クライアントプロセスを終了すると管理の妨害が可能であるため要件 1 を満たさない．また，ソフトウェアアップデート管理システムはクライアント側から定期的にアップデートサーバに更新を確認する仕様であるため，要件 2 の即時実行を満たさない．以上をまとめると表 2 となり，提案機構はこれらに比べ強制的なアップデートに適していると言える．

## 8. おわりに

管理された IT 機器によるネットワークでクライアントソフトウェアを強制的にアップデートする機構を提案した．本システムはクライアント PC 内でサーバの配付したコードを実行できるため，アクセス制御システムでの安全なポリシー配付などに応用できる．

## 参考文献

[1] 株式会社 IDC，“国内プライベートクラウド市場 2010 年の実績と 2011 年～2015 年の予測”，http://www.idcjapan.co.jp/Report/SaaS/j11591001.html（参照 2011 年 12 月 22 日）
[2] 総務省，“「ITS 無線システムの高度化に関する研究会」報告書（案）に対する意見募集の結果及び報告書の公表”，http://www.soumu.go.jp/menu_news/s-news/14422.html（参照 2011 年 12 月 22 日）
[3] IPA，“ハードウェア・ソフトウェア管理”，http://www.ipa.go.jp/security/awareness/administrator/remote/capter8/4.html（参照 2012 年 1 月 12 日）
[4] RFC707, ”AHigh-LevelFrameworkforNetworkasedResourceSharing”, http://tools.ietf.org/html/rfc707（参照 2011 年 12 月 22 日）
[5] Microsoft MSDN, ”How to load an assembly at runtime that is located in a folder that is not the bin folder of the application”, http://support.microsoft.com/kb/837908/en-us?fr=1（参照　2012 年 1 月 12 日）
[6] 佐藤両，福田洋治，毛利公美，白石善明“ポリシー強制をエンドポイントで行うための隠蔽操作によるソフトウェア保護”，情報学ワークショップ 2011（WiNF2011）論文番号 24
[7] 総務省，“国民のための情報セキュリティ SSL/TLS の仕組み”，http://www.soumu.go.jp/main_sosiki/joho_tsusin/security/kiso/k01_ssl.htm（参照 2011 年 12 月 22 日）
[8] MicrosoftMSDN，“.NETFramework デベロッパーセンター”，http://msdn.microsoft.com/ja-jp/netframework/aa496123（参照 2011 年 12 月 22 日）
[9] Microsoft, “Microsoft.NETRemotingFramework の概要”, http://msdn.microsoft.com/ja-jp/library/ms973864.aspx（参照 2011 年 12 月 22 日）
[10]　Apache，“Apache2.2”，http://httpd.apache.org/downloadcgi（参照 2012 年 12 月）
[11]　Microsoft MSDN，“厳密な名前でアセンブリに署名する”，http://msdn.microsoft.com/ja-jp/library/xc31ft41.aspx（参照 2011 年 12 月 22 日）
[12]　富士通四国システムズ，“瞬快”，http://jp.fujitsu.com/group/shikoku/services/packages/shunkai/（参照 2011 年 12 月 22 日）
[13]　PFU，“トータル PC 運用支援システム”，http://www.pfu.co.jp/infra/solution/total.html（参照　2012 年 1 月 12 日）
[14]　NEC ソフト，“PcCreator”，http://www.necsoft.com/press/2008/pdf/081014a.pdf（参照　2012 年 1 月 12 日）
[15]　Microsoft，“Windows Server Update Services”，http://technet.microsoft.com/ja-jp/windowsserver/bb466189.aspx（参照 2011 年 12 月 22 日）
[16]　ORACLE，“Sun Update Connection System 1.0.8 管理ガイド”，http://docs.oracle.com/cd/E19107-01/updconn.sys/819-7283/accessingsolarisupdates/index.html（参照　2012 年 1 月 12 日）