RO-002

センターから端末への動的なコードの配付・実行・検証機構

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

白石 善明† 佐々木 啓† 福田 洋治‡ 毛利 公美††
Yoshiaki SHIRAISHI† Kei SASAKI† Youji FUKUTA‡ Masami MOHRI††

1. まえがき

国税庁の調査では 20 歳から 64 歳までの人口と 65 歳以上の人口の比は，2000 年では 1:36 だったが，2050 年には 1:1.2 となる[1]．また，厚生労働省の調べでは高齢者の単身世帯もしくは夫婦のみの世帯は 2005 年には 850 万世帯だったが，2030 年には 1280 万世帯に増えるという見込みである[2]．厚労省では，高齢者世帯が受動的に医療を受けることができる在宅医療を推進していると同時に，増え続ける老老介護世帯に対する訪問介護を推進している[3]．

医療・看護・介護の間で情報を共有できれば，医師は看護・介護記録をもとに高齢者に最適な医療を提供，あるいは看護・介護の指示を行うことができる．看護師や介護福祉士は投薬などの医療行為を医師に依頼することができるなど，高齢者はよりよい医療・看護・介護サービスを受けることができる．

このような通院医療・在宅医療・訪問看護・訪問介護の連携のための医療機関で作成されたカルテなどの診療記録や介護福祉士が測る血圧などの介護記録を一元管理する医療クラウドの開発が進められている．在宅介護の訪問先でデータセンタに接続できる携帯端末を使用することを考えると，医療クラウドのセキュリティを高いレベルで実現するためには，その端末に導入されたソフトウェアや設置されるファイルなどを医療クラウドの管理者によって管理される必要がある．

このとき，アンチウィルスソフトの導入や医療クラウドへアクセスするソフトウェアのバージョンアップやバージョンの統一などをするには端末を直接操作しなければならない．長時間にわたり端末を医療従事者が使用，あるいは在宅サービスに携行すると，管理者が直接操作できる時間や機会は少なくなる．管理者としては，迅速かつ確実に端末に管理のための操作をしたい．

管理者が医療従事者の端末に対し直接操作を行う方法として，ネットワークを介した PC 管理ツールが挙げられる．PC 管理ツールは予め端末にクライアントソフトウェアを導入し，操作時はクライアントに命令を送り操作を行わせるもので，市販のソフトウェア製品がある．しかし，PC 管理ツールでも，管理のための任意の操作をできるわけではない．そこで，管理者が端末に対して迅速かつ確実に管理のための操作をすることを支援するために，本論文ではサーバ側から任意のプログラムコードをクライアントに配付・実行し，サーバ側でプログラムが実行されたことを確認できる機構を提案する．

† 名古屋工業大学 Nagoya Institute of Technology
‡ 愛知教育大学 Aichi University of Education
†† 岐阜大学 Gifu University

2. 任意のプログラムをクライアントに配付し実行するモデル

2.1 端末管理モデル

図 1 に示した，組織で所有する端末の管理者が存在し，管理者がすべての端末の管理を行うモデルを考える．端末は多くの時間を管理者以外の利用者に業務で利用されているとする．

ここでの管理とは，組織のセキュリティポリシーに基づく端末が守るべき状態を維持することである．また，管理に必要な操作とは，管理を行うために管理対象もしくはそれ以外に対して必要な操作のことである．例えば，セキュリティポリシーの中に“バージョン X のブラウザ A がインストールされていること”とある場合は，バージョン X の A がリリースされると，管理対象となる端末にバージョン X のブラウザ A をインストールするという，管理者による操作が求められる．

また，ここでのモデルには次のような仮定をおいている．

[管理者] 全ての端末の管理を業務とし，管理に必要な操作を確実に行う．

[利用者] 端末を使用し業務を行う．利用者のリテラシは高くなく，例えば仮想マシンを用いたプロセスダンプなどによる操作の妨害はしないものとする．

[端末] 管理者が管理し，利用者が業務に使用する端末である．利用者により携行されて管理者が操作を直接行うことができない場合がある．

図 1 端末管理モデル

2.2 プログラムを実行するコードを配付するモデル

管理者が遠隔地にある端末で操作を実行する方法としてセンターサーバ側にあるプログラム（実行対象プログラム）をクライアントに配付し実行することを考える．実行対象プログラムとは任意の外部プログラムであり，これをソフトウェアのインストーラーやアップデーターなどのプログラムにすることで任意の操作をクライアントで行う．端末には予めクライアントソフトウェア（実行クライアント）がインストールされており，これがサーバから実行対象プログラムを受け取り実行する．これを実現するモデルを図

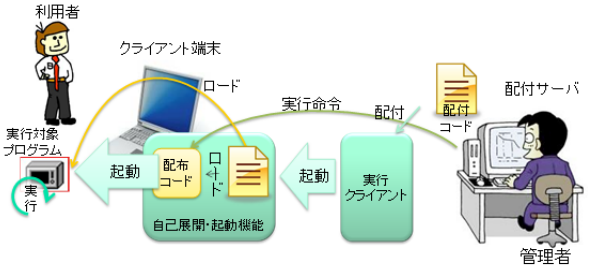

図 2　“ プログラムを実行するコード ” を配付するモデル

2 に示す．本モデルを構成する要素は次の通りである．管理対象のソフトウェアにはコマンドプロンプトで実行するものもあり，引数や実行オプションの入力が必要なものもある．

実行クライアントが任意の実行対象プログラムを起動するときに柔軟に実行方法・引数を変化させる方法として，“ 実行対象プログラムおよびそれを起動するプログラム ” を配付コードとして配付し実行する．構成するエンティティは次のとおりである．

[配付サーバ]　管理者が使用するサーバである．実行対象プログラムを保持し，端末に導入された実行クライアントにこのプログラムを配付・実行命令を送ることで端末を管理する．

[実行対象プログラム]　端末で管理に必要な操作を行うプログラムの実行ファイルである．実行されると動作する端末に対して操作を行う．例えば，セキュリティポリシーの中に “ バージョン X のソフトウェア A がインストールされていること ” でバージョン X のソフトウェア A がリリースされた場合，本プログラムは A のアップデーターとなる．このように任意のプログラムに置き換わり，端末に対し管理に必要な操作を行う．

[実行クライアント]　端末にインストールされるソフトウェアで，端末起動時に立ち上がり常駐する．配付サーバから実行対象プログラムを内包する配付コードを受け取り，配付コードを実行するプロセスを起動する．起動後はプログラム配付の待ち受けに戻る．

[配付コード]　実行対象プログラムとそれを実行するプログラムからなり，実行対象プログラムを外部に出力し，その実行対象プログラムを起動する機能を持つ．

[配付コード（中間言語形式）]　配付コードを配付するための中間コード形式のファイルである．

[配付コード（実行形式）]　中間コード形式の配付コードから作成した実行コードである．

3. クライアントへの動的なコードの配付・実行・検証機構の提案

ある実行対象プログラムを起動・実行し，起動時の引数を変えて再実行を行うような操作を反映させるためには，図 3 のように配付コードロード・起動に関わるプロセスを一度終了して再起動しなければならない．そこで，RPC サーバを端末内に常時起動しておき，配付管理サーバから端末のリソースを利用して実行することで，配付・ロード後に配付管理サーバ側から端末内の実行対象プログラムの実行を制御する命令を受け付けるようにする．

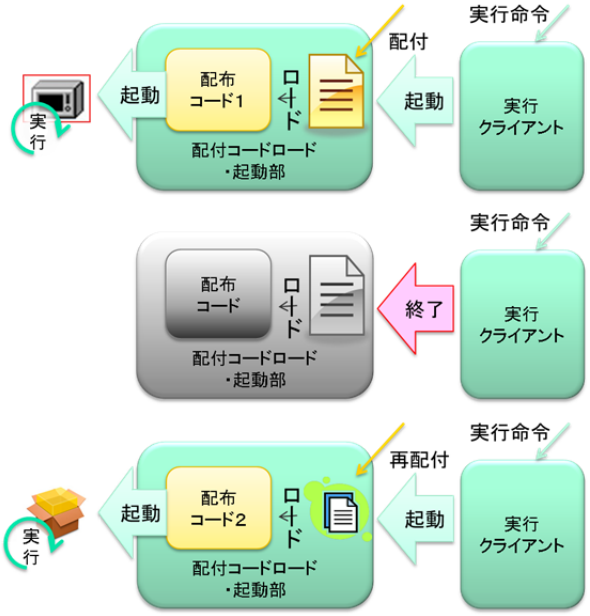

図 3　同一の実行対象プログラムを再操作するための動作

端末上で配付コードの実行ファイルが変更されないように，配付コードはメモリのみに展開し実行する．そのメモリに展開されて起動しているプロセスは自己防衛機能により保護されるものとする．配付コードに署名を付加し端末で実行前に検証することで実行対象プログラムが異なる場合には検知することができる．さらに，配付コードごとに異なる値（証拠情報）を埋め込み，これを実行結果に含ませることで正しく実行されたかを確認する．以上のような同一実行対象プログラムの再操作を確実にできる動的なコードの配付・実行・検証機構を提案する．

3.1 提案機構の構成

図 4 に提案機構の全体構成を示す．

3.1.1 配付管理サーバ

実行対象プログラムと配付コードの端末に対する配付・実行を管理する物理的なサーバホストである．配付時にアプリケーションであるサーバ（.exe）を起動し配付・実行の管理を行う．

[配付コード管理部]　配付コードの配付・実行を管理する．機構全体が動作を開始する動作の起点である．次の動作を順に行う．配付・実行検証部を呼び出して証拠情報をつくり，配付コードに埋め込む．RPC 呼び出し部を呼び出し端末と RPC 接続し，リモートで端末上の起動プロセス管理部を呼び出す．起動プロセス管理部を利用して配付コードロード・起動部を別プロセスとして起動する．このプロセスの実行結果を受け取り，配付・実行検証部を呼び出して第三者による実行結果の改変がないことを検証する．

[RPC 呼び出し部]　端末の RPC 待ち受け部とソケット通信し，サーバから端末のリソースに対し RPC により呼び出せるようにする．

[起動プロセス管理部（インターフェース）]　端末の起動プロセス管理部のインターフェースであり，RPC 接続

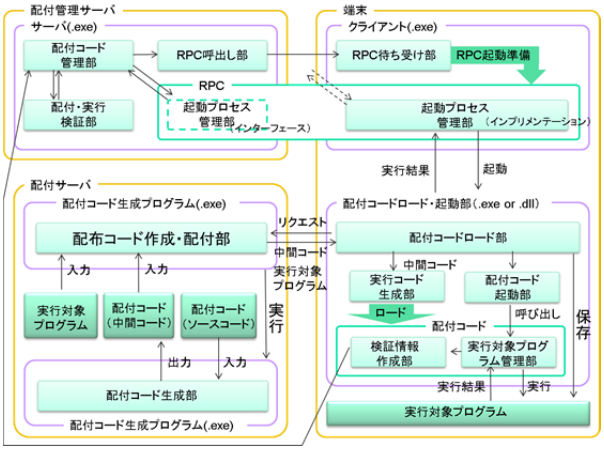

図 4　提案機構の構成

中にこれを呼び出すことで端末側のインプリメンテーションに対し呼び出しを行うことができる．

[配付・実行検証部]　証拠情報を生成し，保存する．また，配付コードのソースコードに証拠情報を埋め込む．動作結果内の証拠情報と自身の保持する証拠情報を突き合わせて動作結果が改変されていないか検証する．

3.1.2 配付サーバ

配付コードと実行対象プログラムを保持し，リクエストに対しこの二つを返すことを主な役割とする物理的なサーバホストである．アプリケーションサーバである配付コード生成プログラム（.exe）が常時起動しており，サーバに対するリクエストにレスポンスする．レスポンスに含まれる中間言語形式の配付コードを生成するのが配付コード生成プログラム（.exe）である．

[配付コード作成・配付部]　リクエストを受けると配付コードの中間言語ファイルと実行形式の実行対象プログラムをまとめてレスポンスとして返す．実装によって，レスポンスを受けた後に配付コード生成部を実行する場合がある．

[配付コード生成部]　配付コードのソースコードから中間言語形式のファイルを生成する．

[配付コード（ソースコード）]　端末に配付されるコードで，実行対象プログラム管理部と検証情報作成部を持つ．そのソースコードである．　証拠情報はここに埋め込まれる．

[配付コード（中間コード）]　端末に配付されるコードで，実行対象プログラム管理部と検証情報作成部を持つ．その中間言語形式のファイルである．配付サーバから端末へ移動する．

[実行対象プログラム]　2.2 節と同じである．任意のプログラムに置き換わる．

3.1.3 端末

一つのプロセスが常駐し，配付時は二つのプロセスが配付実行を行う．常駐するプロセスはクライアント（.exe）で，これは RPC 待ち受け部・起動プロセス管理部（インプリメンテーション）を持つ．起動プロセス管理部が配付時にもう一つのプロセスを起動する．二つ目のプロセスは配付コードロード・起動部.exe で配付コードロード部・実行コード生成部・配付コード起動部を持つ．このプロセスは配付コードと実行対象プログラムをサーバから受け取り，配付コードを自身のプロセスにロードする．配付コードをロードすることで増える機能は検証情報作成部と実行対象プログラム管理部である．配付コードが実行対象プログラムを起動・管理する．

3.1.3.1 クライアント（.exe）

[RPC 待ち受け部]　配付管理サーバの配付命令を待ち受け，命令がくると RPC 接続を開始する．RPC 接続している間はサーバから端末の起動プロセス管理部をリモート呼び出し可能となる．

[起動プロセス管理部（インプリメンテーション）]　配付コードロード・起動部.exe を新たなプロセスとして起動し，実行結果を返す．再配付時，配付コードロード・起動部が終了していない場合これを強制終了する．

3.1.3.2 配付コードロード・起動部（.exe）

[配付コードロード部]　配付サーバにリクエストを送信し，中間言語形式の配付コードと実行対象プログラムを取得する．これらに付加された署名を検証し，配付コードは実行コード生成部を呼び出すことで自プロセスにロードする．実行対象プログラムはファイルとして保存する．この二つが終わると配付コード起動部を呼び出し，ロードされた配付コードを実行する．

[実行コード生成部]　中間言語形式の配付コードを受け取り，実行形式にコンパイルした後，自プロセスにロードする．既存の言語体系でサポートされており，Java のクラスローダーや C#の CLR がこれにあたる．

[配付コード起動部]　配付コード内の実行退場プログラム管理部を呼び出す．

[実行対象プログラム管理部]　実行形式のファイルとして保存されている実行対象プログラムを別プロセスとして起動する．また，検証情報作成部を呼び出す．その際，実行対象プログラムが正常に起動したかどうかを起動ステータスとして検証情報作成部に渡す．

[検証情報作成部]　実行対象プログラムの起動ステータスと証拠情報から検証情報を生成し端末へ送信する．証拠情報は配付前のソースコードだった時に埋め込まれる．

[実行対象プログラム]　2.2 節と同じである．任意のプログラムに置き換わる．

3.2 提案機構の動作

本機構の動作は 3 段階に分けられる．1 つ目は図 5 に示す配付の準備である．ここでは証拠情報を生成し，それを埋め込んだ配付コードの中間言語ファイルの生成を行う．2 つ目は図 6 に示す RPC でのクライアント呼び出しである．リモート接続のコネクションを張りサーバから端末上のリソースを呼び出す．3 つ目は図 7 に示す配付コード・実行対象プログラムのロードと実行である．

3.2.1 配付の準備

1. 配付コード管理部は配付・実行検証部を呼び出す
2. 配付・実行検証部は証拠情報を生成し配付コードのソースコードに埋め込む
3. 配付コード生成部はソースコードから中間言語形式の配付コードファイルを生成する

3.2.2 RPC によるクライアント呼び出し

1. 配付コード管理部は RPC 呼び出し部を呼び出す

2. RPC 呼び出し部は端末の RPC 待ち受け部に RPC 接続のリクエストを行う
3. RPC 呼び出し部と RPC 待ち受け部は RPC 接続のコネクションを張る
4. 配付管理サーバの配付コード管理部は起動プロセス管理部のインターフェースを呼び出す
5. RPC により端末の起動プロセス管理部がリモート呼び出しされる
6. 端末の起動プロセス管理部は配付コードロード・起動部.exe を新たなプロセスとして起動する

3.2.3 配付コードロード・起動部（.exe）の動作

1. 配付コードロード・起動部内の配付コードロード部は配付サーバにリクエストを送信する
2. 配付サーバの配付コード作成・配付部は中間言語形式の配付コードファイルと実行形式の実行対象プログラムを合わせてレスポンスとして端末へ返す
3. 配付コードロード部は取得した二つのファイルに付加された署名を検証した後，実行対象プログラムをファイルとして保存し，実行コード生成部を呼び出す
4. 実行コード生成部は中間言語形式の配付コードを実行形式にコンパイルした後，自プロセスにロードする
5. 配付コードロード部は配付コード起動部を呼び出す
6. 配付コード起動部は配付コード内の実行対象プログラム管理部を呼び出す
7. 実行対象プログラム管理部は実行形式のファイルとして保存されている実行対象プログラムを別プロセスとして起動し，検証情報作成部を呼び出す
8. 検証情報作成部は検証情報を生成し端末へ送信し，配付コードロード・起動部（.exe）は終了する
9. 配付・実行検証部が検証情報から配付コードと実行対象プログラムが正しく実行されたことを検証する

3.3 動的読み込み

提案機構では配付コードを配付サーバから受け取った後，配付コードロード・起動部（.exe）のプロセスに動的に読み込まれる．動的読み込み[4]は実行時にリンクされていないライブラリを実行中のプロセスがロードすることである．コンパイル言語では事前コンパイルもしくは実行時コンパイルでのコンパイル時にリンカによってライブラリのオブジェクトファイルをロードするが，明示的に指定することで実行中にライブラリを読み込むことができる．C，C#，Java，Ruby，Flex などの言語でサポートされている．読み込むとき，ライブラリの改変を防ぐために署名をサポートする言語もある．今回は配付コードの改変を防ぐために言語のサポートする署名機能を使用する．

3.4 RPC

提案機構では端末の起動プロセス管理部を呼び出す際に RPC を使用している．RPC は，ソケット通信などをラップすることで他の PC の計算資源に直接アクセスしているようなインターフェースを提供する機能である．C，C#，

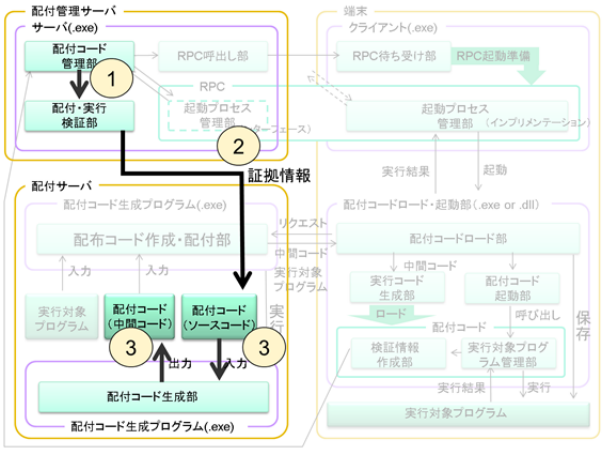

図 5　動作の流れ　～1. 配付の準備～

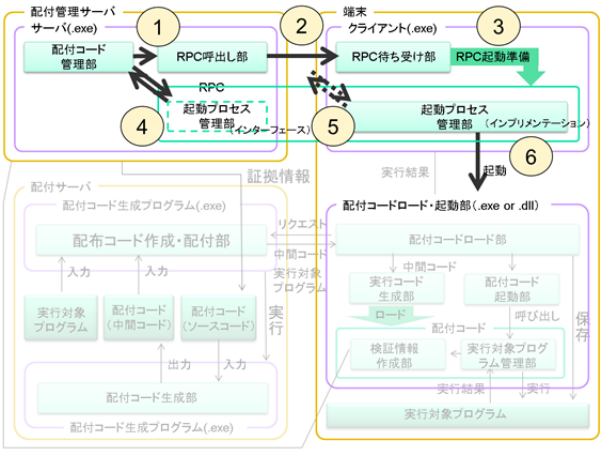

図 6　動作の流れ　～2. RPC によるクライアント呼び出し～

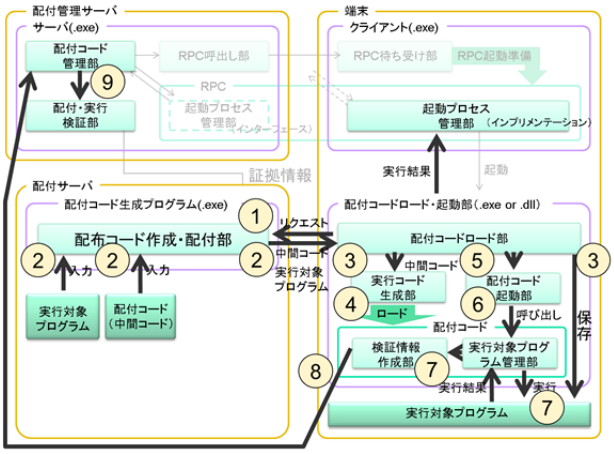

図 7　動作の流れ　～3. 配付コード・実行対象プログラムのロードと実行～

Java など多くの言語でサポートされている．提案機構ではこれを使用することで配付コードロード・起動部の動作をサーバから制御し，ロードや実行の方法・タイミングをサーバから制御する．

3.5 配付コードが正しく動作したことの確認

証拠情報は配付ごと・端末ごとに違う値を生成する乱数である．制御部が生成し，検証部が保持する．これをサーバ側の検証情報とすると，クライアント側の証拠情報は配付前に配付コードの報告部に書き込む．置換部が実行されると，報告部は置換部の実行結果と証拠情報を報告にまとめ，制御部に返す．検証部でサーバ側の検証情報とクライアント側の証拠情報を比較し，同じであれば正常に動作したとみなす．

動的読み込みにより利用者は配付コードの実行ファイルを取得できないため配付コード内の値を取得できない．証拠情報は配付コード内に値として書き込まれるため，利用者は取得できない．つまり，サーバ側の検証情報と同じ物を提出できるのは報告部のみである．したがって，証拠情報が正しければ報告部は正常に動作したといえる．このとき，報告部は置換部の前に動作するため，置換部も正常に動作したといえる．

4. 性能評価

提案機構の性能評価をするために計測用システムを C# で実装した．

100，500，1000，5000 クライアントに対して同時に配付を行ったときの，サーバが端末のクライアントに読み込み命令を送ってから動作結果を受け取るまでのサーバでの 1 クライアントあたりの平均動作時間とサーバでの占有メモリを計測した．表 1 の環境で動作させて計測した．実験機はすべて有線 LAN で接続しており．端末での実行時間が無視できる程度のコードを配付し実行した．

測定結果を図 10 に示す．1 クライアントの応答時間はクライアント数が変わっても約 380 ミリ秒で一定の値となった．クライアント数が増加すると共にサーバの占有メモリも増加したが，5000 クライアントでも 600KB を下回ることを確認した．

5. 既存技術との比較

5.1 PC 運用支援システム

PC 運用支援システムは組織内の PC に対し利用者の介入しない形でサーバからファイル等を読み込ませるシステムである．

例えば，“瞬快”[5] は，多数の PC を維持・管理していくうえで発生する様々な作業を効率化できる．主な用途は利用者が使用して変更した PC 環境を，再起動すると元の環境に復元することである．リモート操作機能がついており，管理ソフトウェアから複数のクライアント PC を遠隔操作が可能である．任意の指定したファイルやフォルダを配付でき，さらに配付前後に任意のプログラムやバッチファイルを起動できるという機能もある．

“トータル PC 運用支援システム”[6]は管理対象であるハード・ソフトの把握と最適化，セキュリティの保全に関する継続的な維持・管理を行う．ハード・ソフトの把握は

表 1　計測環境

実験機名	OS
	CPU
	メモリ
配付管理サーバ／配付サーバ	Windows7 SP1 x64
	Core2Quad Q9550 @2.83GHz
	4GB
端末 A	Windows7 SP1 x64
	Core i7 2620M @2.70GHz
	16GB
端末 B	Windows7 SP1 x64
	Core2Quad Q9550 @2.83GHz
	4GB
端末 C	Windows7 SP1 x64
	Core2Quad Q9550 @2.83GHz
	2GB
端末 D	Windows7 SP1 x64
	Core2Quad Q9550 @2.83GHz
	3GB

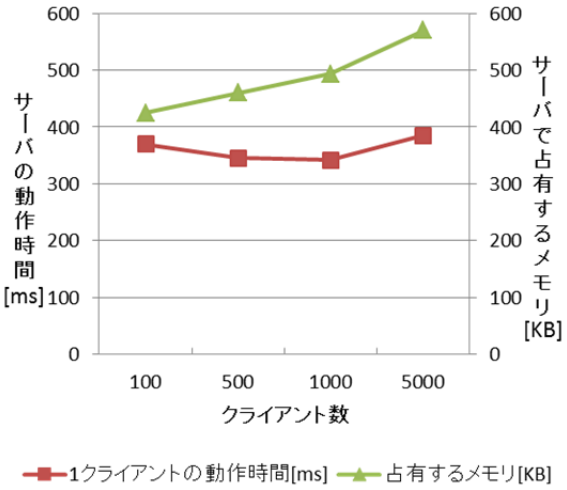

図 8　計測結果

ネットワーク接続された PC の自動検知や PC 情報の自動収集によって行い，セキュリティの維持・管理はセキュリティパッチの配付，適用とセキュリティ適用状態の監査を行う．セキュリティパッチの配付，適用にはポリシーの配付と適用も含まれる．

“PcDoctor”[7]は PC 導入から障害復旧，資産情報収集などの機能を持つ運用管理支援ソフトウェアである．管理対象 PC をネットワークブートし，コンソールを管理端末に表示するリモートコンソール機能を持ち，サーバからプログラム配付を行うことができる．

以上のような PC 運用支援システムは多くの端末に対し同時に実行できる操作は既定の操作に限られる．例えば瞬快の場合，ファイルの送信は全ての端末に同時に行うことができるが，一般的な操作はリモート接続により一台ずつ人手で行う必要がある．それに対して提案機構は任意のプログラムをサーバから配付し，実行する操作ができるようになっている．

5.2 ソフトウェアアップデート管理システム

ソフトウェアアップデート管理システムでも同様にサーバからファイルを読みこませることができる．例えば Microsoft の提供する“ Windows Server Update Services ”（WSUS）[8]は Microsoft Update からリリースされた更新プログラムをネットワーク内のコンピュータに配付する作業の管理を行うサービスである．管理コンソール上で更新プログラムをどのように配付するか制御することができる．また，ORACLE の提供する“ Sun Update Connection System ”[9]は最新の修正及び機能にアクセスし Solaris システムを常に最新の状態に保つシステムで，複数のリモートシステム更新管理を行うことができる．

ソフトウェアアップデート管理システムは端末にインストールされたアップデーターによってアップデートを行う．このとき，WSUS など多くはアップデーターが定期的にサーバへ接続し更新の有無を確認する構造である．これはアップデーターから更新の確認がくるまでサーバは更新ファイルをクライアントに送信することができないということである．さらに，アップデーターは利用者が設定を変更できる場合がある．それに対して提案機構はサーバから強制的にプログラムを配付し，実行できるようになっている．

6. おわりに

本論文ではサーバ側で用意した任意のプログラムコード（配付コードに含まれる実行対象プログラム）をクライアントに配付し実行する機構を提案した．

端末の台数が多くてもサーバの負荷とメモリ使用量は問題にならないことを計測用システムの実装により確認した．提案機構は既存技術に比べて，任意のプログラムをサーバ側から実行でき，サーバ側が主体でプログラムを実行し，正しく実行されたことをサーバ側で確認できる機能を持つ．

提案機構は管理者が端末に対し，迅速かつ確実に管理のための操作を強制でき，多数の携帯端末をクライアントとして用いるクラウドサービスに使われることが期待される．

提案機構のセキュリティについて詳細に分析すること，多数の実端末による評価が今後の課題としてあげられる．

参考文献

[1] 国税庁，“ 少子・高齢化ってなに？ ”，http://www.nta.go.jp/shiraberu/ippanjoho/gakushu/nyumon/page11.html?non
[2] 国立社会保障・人口問題研究所，“ 日本の世帯数の将来推計（全国推計）2008（平成 20）年 3 月推計，http://www.ipss.go.jp/pp-ajsetai/j/HPRJ2008/gaiyo20080314.pdf
[3] 国立保健医療科学院 健康危機管理支援ライブラリー，“ 在宅医療の推進について ”，http://h-crisis.niph.go.jp/node/53284
[4] Microsoft，リンカによる DLL の遅延読み込み，http://msdn.microsoft.com/ja-jp/library/151kt790%28v=vs.80% 29.aspx
[5] 富士通四国システムズ，“ パソコン運用支援パッケージ 瞬快マニュアル ”，http://jp.fujitsu.com/group/shikoku/downloads/services/packages/shunkai/simplemanual.pdf
[6] PFU，“ トータル PC 運用支援システム ”，http://www.pfu.co.jp/infra/solution/total.html
[7] NEC ソフト，“ PcDoctor ”，http://www.necsoft.com/press/2008/081014a.html
[8] Microsoft, “Microsoft Windows Server Update Services(WSUS)”, http://technet.microsoft.com/ja-jp/wsus/bb332157
[9] Oracle，“ Sun Update Connection ”，http://docs.oracle.com/cd/E19253-01/819-0359/gasua/index.html