IC カード認証と連携した非ドメイン型移動ユーザプロファイルの共有端末への実装

葛生和人† 平野靖† 間瀬健二† 渡邉豊英†

† 名古屋大学情報連携基盤センター
〒464-8601 名古屋市千種区不老町
E-mail: † {kuzuu,hirano,mase,watanabe}@itc.nagoya-u.ac.jp

概要 本研究において筆者らは，ドメインコントロールサーバによるドメイン構築を必要としない非ドメイン型移動ユーザプロファイルの考え方を導入し，さらにその概念を IC カード認証プロセスに組み込むことにより，クライアント PC 上でカードユーザに対応した個別の作業環境を再構築できるようなログオン認証システムを実現した．本システムは，特に，不特定多数のゲストユーザを対象とするような共有端末環境に適用することにより，その効果が期待される．すなわち，使用者の作業環境が端末ごとに再構築されるというユーザの利便性を保ちつつ，PKI と連携させたスマートカードログオンの採用により端末認証へのセキュリティを向上させ，さらには，ドメインコントロールサーバの導入を必要としないという点で管理運用面と経済性の両面において有利なソリューションである．

Implementation of IC Card Authentication System combined with Roaming User Profile not belonging to Domain into a Shared Terminal

Kazuto KUZUU†, Yasushi HIRANO†, Kenji MASE†, and Toyohide WATANABE†

† Information Technology Center, Nagoya University

Furo-cho, Chikusa-ku, Nagoya-shi, Aichi 464-8601, JAPAN

E-mail: †{kuzuu,hirano,mase,watanabe}@itc.nagoya-u.ac.jp

Abstract In this study, we developed newly the logon authentication system which can rebuild the environment of the individual corresponding to each card user on client PC terminal. This system is realized by introducing the concept of a roaming user profile which need not prepare any domain environments, and combining that concept with IC card authentication. This system is especially useful for a shared terminal where both high security and user's convenience are required. Furthermore, since a domain control server is not required, this system is regarded as advantageous solution from the points of both user management and economical efficiency.

1 はじめに

近年，IC カードの普及とセキュリティ向上に対する要求から PC へのログオン認証においてもスマートカードログオンの考え方が急速に普及しつつある．しかしながら，その認証方法は，個別のカード ID や PIN コードを用いた方法からカードに格納した証明書を利用して PKI と連携させる方法にいたるまでさまざまなものが存在しており，それらシステムの標準化はいまだ不十分であるというのが現状である．すなわち，実際のシステムの多くはカードベンダーが独自に提供するアプリケーションであるため，ユーザが新たな利用法を探ったり，異なるベンダー間でのカード連携を視野に入れたアプリケーションを開発することが困難であるという問題が生じる．筆者らはそのような状況に対処すべく，これまで Java Card™ Technology [1]を利用した PKI 連携用の IC カードアプリを Java Card VM 上に構築し，スマートカードログオン実現のための独自のミドルウェアを Windows 上に実装してきた[2][3].

実際，これらのシステムは，ベンダーの異なるカードであっても，そこに実装された Java Card VM 上の Java Card アプリであれば，共通のミドルウェアを用いてスマートカードログオンが機能することを確認している．この点で，これまで開発してきたスマートカードログオンシステムは，汎用性の面で有用であるといえる．しかしながら，汎用性と共に利便性の面でも他のアプリケーションに対して優位であることは，独自のアプリケーション開発において重要な課題であろう．そこで筆者らはさらに独自の新たな利用法を探り，アプリケーションに利便性を追求した付加価値を加えるべく，共有端末へのニーズとその利用方法に着目した．

2 共有端末のニーズ

一般の利用者が対象となる共有端末の例としては，図書館等の公共施設に設置されている端末が挙げられる．そのような共有端末では，通常はセキュリティの関係からゲストユーザログオンの状態で画面が開かれ，その上でログオフできない状態となっており，さらに，そこでは利用可能なアプリケーションが蔵書検索用のソフトに制限されていたり，マウスや画面の操作制限が端末ごとに厳重に管理されたものとなっている．このようなシステム管理は，部外者も含めて広範囲な利用者を想定した公共施設の共有端末の場合は，セキュリティ上必要不可欠なものであるが，関係者の入退出がある程度制限された中小規模の施設では，これまでは管理運用面での人的経済的限界から共有端末といえども必然的にセキュリティ面で不十分な対応とならざるを得なかった．その一方で，そのような共有端末においては，ある程度の自由度を持った使用方法が要求されるのも事実である．たとえば，図書館側で用意された検索システムだけではなく，一般のインターネット検索エンジンや，文献のダウンロードシステムを利用したいなど，ある程度の利便性も必要となってくる．当然その場合のセキュリティの確保はログオン時の認証をはじめとして重要な課題となるが，そこで注目されるのが，IC カードを用いたログオン認証である．つまり，IC カードのセキュリティ機能を生かすことにより，端末利用者にある程度の自由度のある作業環境が提供できるような共有端末利用形態があり得るのではないかと考えられる．

そのような観点から，共有端末の利用形態に求められる仕様を考えると以下のようなものがあげられる．

1 IC カードの所有者であれば利用可能である．
2 利用者はゲストユーザ権限の範囲内で端末を利用できる．
3 いずれの共有端末でもユーザの作業環境は保持される．
4 ユーザ個人情報は端末には残されないが，管理者はユーザの作業ログ，作業環境などいつでもチェックできる．

上に掲げた共有端末としての利用形態を実現しようとしたとき，まず考えられるのは Windows 系サーバからクライアント PC システムを構築して，ドメインを構成することであろう．そこでは，移動ユーザプロファイルという概念[4]を利用することにより，個別ユーザの作業環境はドメインコントロールサーバ側に保存され，登録したユーザアカウントに対応した環境がどの共有端末においても再現されるようになる．特に，スマートカードログオン機能と組み合わせて，セキュリティポリシを設定すれば，上で述べた共有端末の利用形態は実現可能である．また，同様のアプローチは，Windows 系サーバ以外でも Samba のドメインコントローラ機能[5]を代替サーバとして活用すれば，Linux/UNIX 系マシンを通しての利用も可能である．図 1 は，共有端末用の移動ユーザプロファイルがドメインコントロールサーバで管理されている状態を示したものである．

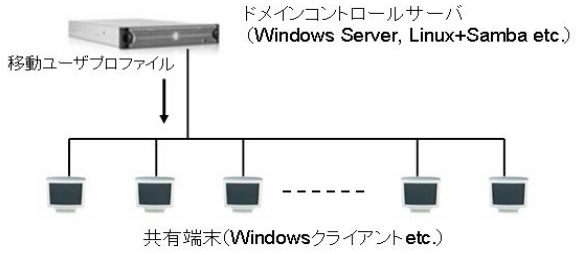

図 1 ドメイン構成による共有端末

しかし，上に示したようなドメイン構成によるユーザ管理ではなく，LDAP などのディレクトリサーバを用いてユーザ情報を独自に管理運用しているようなシステムが既に構築されている環境では，必ずしも少数ではないクライアント PC，少人数ではない IC カード所有者のために，新たにドメインコントロールサーバを導入し，クライアントアクセスライセンスの提供や個別利用アカウントの設定などをしなければならないということは，経済性の面でも運用面でも多大な労力を要することとなる．

3 非ドメイン型移動ユーザプロファイル

Windows 系システムのユーザ作業環境はユーザプロファイルにより管理されるが，これは，その利用形態によって，ローカルユーザプロファイル，移動ユーザプロファイル，固定ユーザプロファイルに分けられる[4]．これらプロファイルのうち，ローカルユーザプロファイルはスタンドアロンのマシンに登録ユーザの作業環境が保存される．一方，移動ユーザプロファイルや固定ユーザプロファイルはドメイン管理サーバによ

って管理保存されるため，クライアントマシンが異なってもユーザごとに作業環境が再構築されることとなる．なお，図 2 に示すように，移動ユーザプロファイルが登録ユーザごとにプロファイルの書き換えが可能で個々の作業環境が再構築されるのに対し，固定ユーザプロファイルはサーバ管理者が1つのプロファイルを管理し，複数のユーザが同じ作業環境を共有するものである．

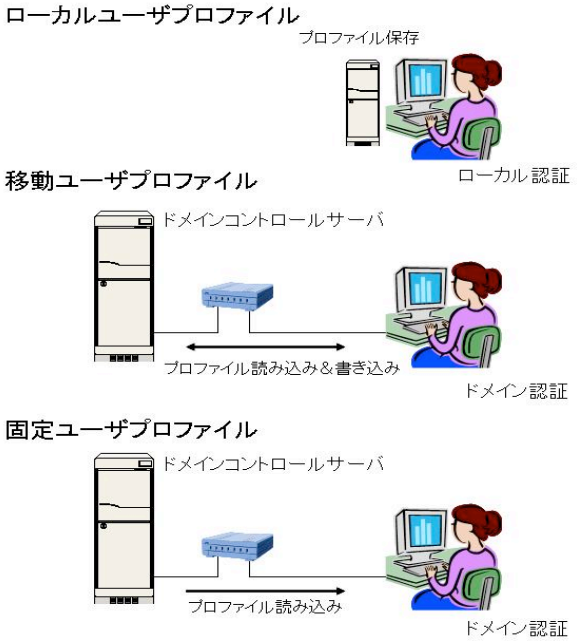

図 2 ユーザプロファイルの種類

いずれにしても，サーバ側で管理されるユーザプロファイルを利用すれば，共有端末の利用形態を満足させるシステムを構築することは可能であろう．しかし，LDAP などのディレクトリサーバを用いてユーザ情報を管理運用しているようなシステムが既に構築されていた場合，新たに別のドメイン管理サーバを導入し，利用者のための移動ユーザプロファイルに対応した新たなユーザアカウントの設定も必要となり，経済面，管理運用面での負担が大きくなってくる．

そこで，前項で述べたような共有端末の利用形態を満足するために，筆者らはドメインを構成する必要のない，もう少し簡便なソリューションを提案する．

Windows システムにおけるユーザプロファイル構築プロセスでは，ログオン認証時にユーザプロファイル構成ファイルの1つである NTUSER.DAT に含まれるプロファイルデータ（作業環境パラメータ）をレジストリハイブである HKEY_CURRERNT_USER にロードすることにより，個別ユーザの作業環境を再構成している．このプロセスは，筆者らが既に GINA（Graphical Identification aNd Authentication）[6]を拡張する形で開発してきたスマートカードログオンシステム[2][3]のミドルウェアを利用すれば，ログオン認証プロセスの前に挿入することが可能であり，システムが行うレジストリのロードプロセスへの引き渡しが実現する．したがって，ログオン認証前に作業環境の構築に必要なユーザプロファイルのみをネットワーク上の別のストレージから取得しローカルファイルシステム上に展開できるようにすれば，ログオン認証そのものは，ドメイン認証である必要はなく，ローカルマシンのゲストユーザとして行われるだけで十分な形となる．ただし，この場合，ゲストユーザとは言っても IC カードによって PKI と連携した証明書検証を行っており，決して匿名ユーザではない点がセキュリティ上重要なポイントとなる．なお，展開されるべきユーザプロファイルは，各ユーザの IC カードに格納されたユーザ ID を通して，識別可能な個別ファイル名でストレージサーバ上に保存される．図 3 は，ゲストユーザ用端末と PKI 認証で利用される LDAP サーバ，ユーザプロファイル格納用のストレージサーバ，それぞれの間でのデータ通信状況を示したものである．

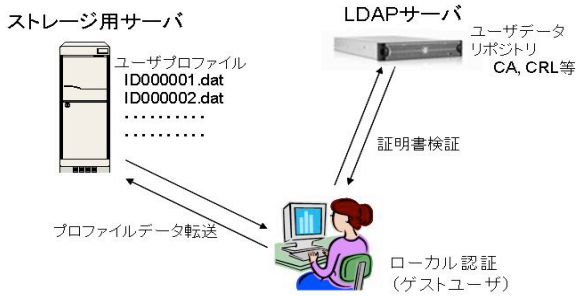

図 3 ID で識別可能なユーザプロファイルの格納

前述のとおり，本システムでは，ドメイン構成を通したログオン認証の必要はなくなる．ただし，そのためにドメイン構成で管理されるセキュリティポリシも存在せず，この場合のユーザプロファイルに対するセキュリティポリシはあくまでローカルマシンで設定されるゲストユーザとしてのものとなる．そして，共有端末の利用者はローカルマシン上では常に同一のゲストアカウントとしてログオンしているため，ローカルマシンでのセキュリティポリシの設定がユーザプロファイルに反映されることとなる．以上のようなことから本システムで導入されるユーザプロファイルは非ドメイン型移

動ユーザプロファイルと呼ぶことができる.

なお, 既に LDAP などのディレクトリサーバを用いてディレクトリ情報を管理, 運用しているような環境では, 本システムはそれらのサーバや情報資源を流用することができる. すなわち, Windows 系ドメイン構成の新たな採用によって生ずる別系統のディレクトリ情報管理, およびそれに伴うシステム運用の煩雑さを避けることができるという点で大きなメリットとなる.

4 IC カード認証と移動ユーザプロファイル連携

4.1 IC カード認証ミドルウェア

前項で述べたように本システムでは移動ユーザプロファイルの概念を導入するに当たり, ログオン認証前に該当プロファイルをシステムにセットアップする手順を組み込む必要がある. このプロセスは, 既に筆者らが開発したスマートカードログオン用ミドルウェア[2][3]の拡張により実現できる.

本システムに使われる IC カード認証ミドルウェアの実行プロセスは, Windows のログオンプロセスで使われる GINA を拡張したものであり, システム起動時の winlogon.exe から呼ばれるサブルーチン群のダイナミックリンクライブラリ(DLL)で構成される. 実際の GINA の実行シーケンスを図 4 に, スマートカードログオンプロセスに関わる IC カードアクセスルーチンや LDAP サーバ通信ルーチンのプロトコルを図 5 に示す.

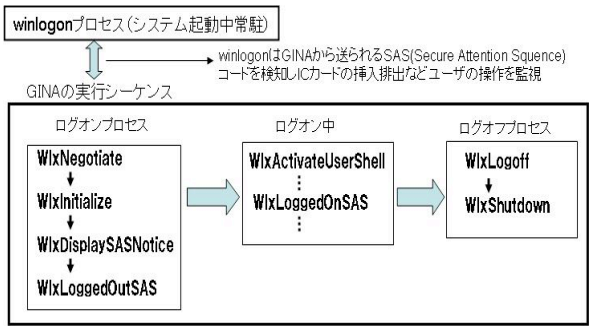

図 4 GINA の実行シーケンス

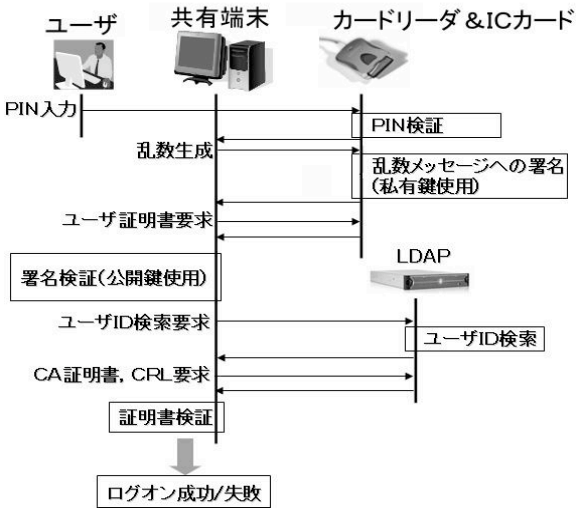

図 5 スマートカードログオンプロトコル

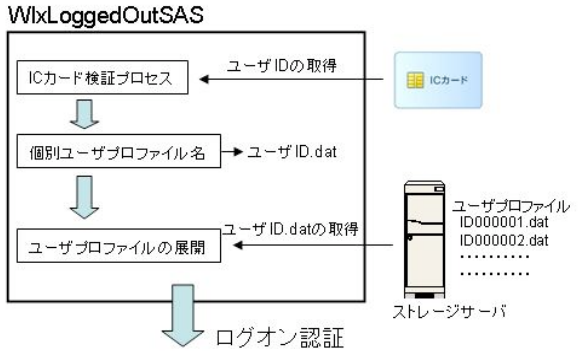

図 6 ID 番号から個別プロファイル名指定

スマートカードログオンプロトコルに示される手続きは, 実際には GINA 実行シーケンス(図 4)の中では, WlxLoggedOutSAS で処理される. なお, この時点でストレージサーバに格納された個別ユーザプロファイルは IC カード内のユーザ ID とファイル名を通して紐付けされている. したがって, ログオン時のユーザプロファイル取得と, IC カード認証を通したユーザ ID の取得はここで連携されることとなる. なお, 図 6 は, 上で述べた各情報の取得タイミングを示したものである.

4.2 移動ユーザプロファイルのストレージへの格納

作業環境の再構築に必要なユーザプロファイルは, どこの共有端末からも参照可能なストレージに格納しておく必要がある. この場合のデータストレージ用サーバは, Windows, Linux いずれのシステムでも問題ないが, 個人情報がネットワーク上を流れるため ssh のような暗号化プロトコルの利用できるシステムで

あることが望ましい．本システムではLinux 上の ssh サーバを利用することとした．なお，ssh サーバへのアクセス権限は，通常ユーザプロファイルの管理者であるストレージサーバ管理者のものであるが，本システムではミドルウェアからシステム権限で直接アクセスされる．すなわち，一般ユーザ権限でストレージサーバにアクセスされることは無く，また，パスワード情報が開示されることもないためセキュリティが確保される．

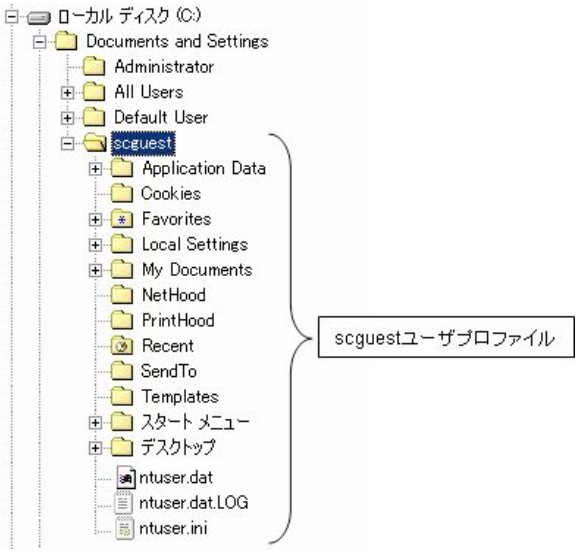

図 7 ユーザプロファイル構成

一方，実際に格納されるユーザプロファイル情報は Windows ではシステムドライブの Documents and Settings 中のユーザ名で表されるフォルダに納められている．これらは，具体的には図7のようなフォルダ群およびファイルで構成されているが，格納時の形態についての制限は無いため，本システムにおいてはデータ格納前にzip形式でパッケージングし転送する形式をとった．

4.3 アクセス権限とユーザプロファイル

Windows のファイルシステムである NTFS は，それぞれのファイル，フォルダに対してアクセス権限が細かく設定される．共有端末のゲストユーザとして登録されるユーザプロファイルに対しても同様であり，アクセス権限は，通常のそのユーザ自身によるプロファイルの読み出し，変更が許可されている．しかしながら，共有端末のように異なる端末からログオンした場合，Windows では，同じゲストユーザであってもシステム上異なる SID（セキュリティ識別子）[7]が割り当てられる．したがって，図 8 に示すように共有端末上に同じユーザアカウントでユーザプロファイルを展開しながら，その変更権限が拒否されてしまう可能性がある．

ドメイン構成下での移動ユーザプロファイルに対しては，ユーザのアクセス権限はドメインにより管理されるが，本システムにおけるユーザプロファイルに対しては，上の状況を回避するためにあらかじめすべてのユーザに対する読み出し，変更権限を有効にしておく（Everyone:フルコントロール）必要がある．これは，NTFS を利用したシステムでは，Windows のファイルプロパティ属性メニュー（図9）により設定可能である．

なお，上記のようなアクセス権限の設定は，ローカルマシン上にプロファイルが残されるため，ゲストユーザや管理者以外のユーザが共有するような通常の端末では，セキュリティや個人情報の保護の観点から好ましいものではないが，本システムにおいては，ログオフ時にはプロファイルはストレージサーバ上に移され，ローカルマシン上に残ることはない．また，プロファイルを含むフォルダに関してネットワーク共有の設定を行わず，Windows XP ではリモートデスクトップの接続を許可しないことにより，他のマシンからのアクセスも不可能となる．すなわち，ストレージサーバ上のプロファイルデータはそのサーバ管理者または ID 番号と紐付けされたゲストユーザが端末へログオンするときのみアクセス可能となり，一般ユーザからは保護されることになる．なお，図 10 はログオンプロセス時にプロファイルがレジストリに登録され，ログオフプロセス時にプロファイルデータが端末から削除されることを示したものである．

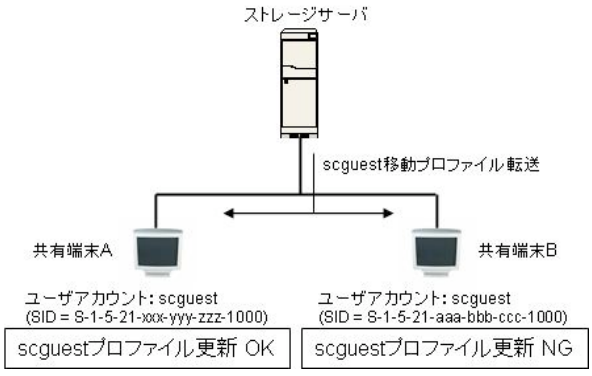

図 8 異なる SID でのプロファイルへのアクセス

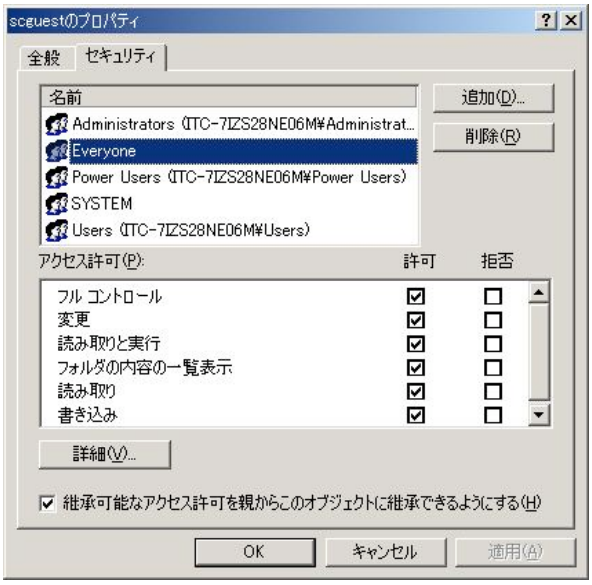

図 9 プロファイルへのアクセス権限設定

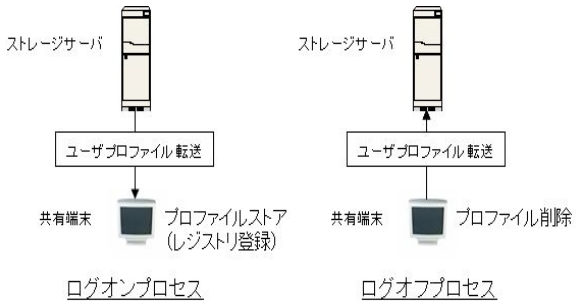

図 10 ストレージ上プロファイルの保護

4.4 移動ユーザプロファイル通信プロセス

前項で示したように，ユーザ ID で識別されるゲストユーザプロファイルは，ログオフ時に zip コマンドによりまとめられ，ssh 通信プロトコルを通してストレージサーバに送信される．そして，ユーザのログオン時には，逆にストレージサーバよりユーザ ID に相当するプロファイルデータが取得され，ローカルシステムに展開される．これらのプロセスは，具体的には GINA の実行シーケンスの中で外部プロセスとして実行される．

これらプロファイルの取得，格納のための外部プロセスはユーザのログオン，ログオフの直前にそれぞれ呼ばれる．（図 11）

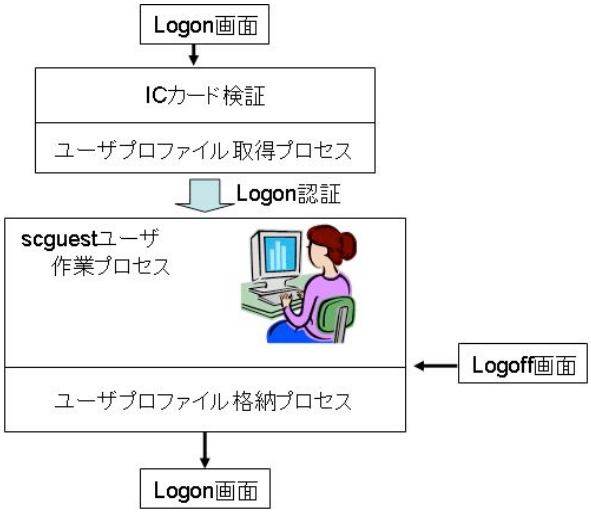

図 11 プロファイルの取得，格納プロセス

ログオン直前のプロファイル取得に関しては，取得以前に IC カード認証による ID 番号の取得が行われている必要があるため，IC カード検証ルーチンの後に呼ばれ，該当するプロファイルデータを獲得，システム上に外部プロセスを通じて展開される．なお，実際の外部プロセス呼び出しは WIN32 API に含まれる CreateProcess を用いており，そのルーチンをリスト 1 に示す．また，ログオフ直前のプロファイル格納に関しても同様で，リスト 2 のような外部プロセス呼び出しを通じて，ストレージサーバ上に格納される．

一方，本システムではプロファイルデータの通信プロセスとして ssh サーバを利用しているが，通常，Windows システムにおいてクライアント側に格納される ssh 通信用の公開鍵情報は，カレントユーザのレジストリにロードされている．したがって，本システムのように，ログオン前にシステムユーザが外部プロセスとして ssh 通信を行う場合には，システムユーザがデフォルトとして使用するレジストリにあらかじめ公開鍵情報を格納しておく必要がある．実際には，Windows システムユーザがデフォルトとして参照するレジストリキー HKEY_USERS¥DEFAULT¥Software に対して図 12 のように公開鍵情報を設定する．

```
ZeroMemory(&si_prof, sizeof(si_prof));
ZeroMemory(&pi_prof, sizeof(pi_prof));
si_prof.cb = sizeof(si_prof);

CreateProcessA(NULL, "pscp.exe
                        ストレージ上プロファイルデータ
                        "c:¥Documents and  Settings¥scguest.zip"",
                        NULL, NULL, FALSE, 0, NULL, NULL, &si_prof, &pi_prof);

WaitForSingleObject(pi_prof.hProcess, INFINITE);
CloseHandle(pi_prof.hThread);
CloseHandle(pi_prof.hProcess);

ZeroMemory(&si_prof, sizeof(si_prof));
ZeroMemory(&pi_prof, sizeof(pi_prof));
si_prof.cb = sizeof(si_prof);

CreateProcessA(NULL, " unzip.exe
                        "c:¥Documents and  Settings¥scguest.zip"  -d c:¥"
                        NULL, NULL, FALSE, 0, NULL, NULL, &si_prof, &pi_prof);

WaitForSingleObject(pi_prof.hProcess, INFINITE);
CloseHandle(pi_prof.hThread);
CloseHandle(pi_prof.hProcess);
```

リスト 1 プロファイル取得の外部プロセス呼び出し

```
ZeroMemory(&si_prof, sizeof(si_prof));
ZeroMemory(&pi_prof, sizeof(pi_prof));
si_prof.cb = sizeof(si_prof);

CreateProcessA(NULL, " zip.exe
                        "c:¥Documents and Settings¥scguest.zip"
                        "c:¥Documents and Settings¥scguest"",
                        NULL, NULL, FALSE, 0, NULL, NULL, &si_prof, &pi_prof);

WaitForSingleObject(pi_prof.hProcess, INFINITE);
CloseHandle(pi_prof.hThread);
CloseHandle(pi_prof.hProcess);

ZeroMemory(&si_prof, sizeof(si_prof));
ZeroMemory(&pi_prof, sizeof(pi_prof));
si_prof.cb = sizeof(si_prof);

CreateProcessA(NULL, "pscp.exe
                        "c:¥Documents and  Settings¥scguest.zip"
                        ストレージ上プロファイルデータ",
                        NULL, NULL, FALSE, 0, NULL, NULL, &si_prof, &pi_prof);

WaitForSingleObject(pi_prof.hProcess, INFINITE);
CloseHandle(pi_prof.hThread);
CloseHandle(pi_prof.hProcess);
```

リスト 2 プロファイル格納の外部プロセス呼び出し

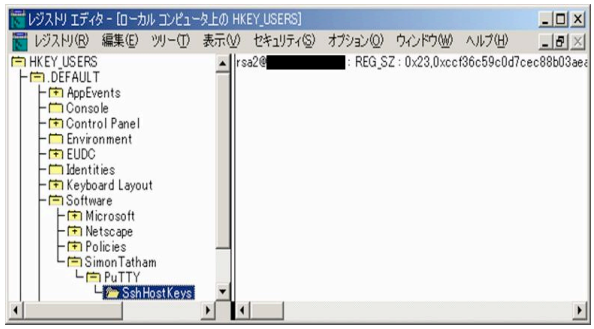

図 12 公開鍵情報のレジストリへの格納

5 システム仕様と共有端末への実装

本システムを検証するに当たって利用した各ソフトウェア，ハードウェアの仕様とシステムの実装手順について述べる．

5.1 IC カード，カードリーダおよびカードアプリ

IC カードは，接触，非接触のいずれにも対応したデュアル・インタフェース型 1MB メモリを有するものでネイティブプラットフォーム上に Java Card VM を組み込んでいる．カード内でのセキュリティ API は RSA, DES, T-DES の複数の暗号処理が可能である．詳細の仕様は，表 1 に示すとおりである．また，カードリーダは，想定した 2 台の共有端末のそれぞれに対して接触型，非接触型のものを用意した．（表 2）

表 1 IC カード仕様

タイプ	Java Card VM	
	接触型	非接触型
準拠規格	ISO/IEC7816	ISO/IEC 14442 Type B
通信プロトコル	T = 0, 1	ISO/IEC14443-4
通信速度(kbps) MAX	19.2	424.0
メモリ	1M バイト（フラッシュメモリ）	
CPU	16 ビット	
セキュリティ	RSA, DES, T-DES 演算対応	

表 2 カードリーダ仕様

タイプ	接触型	非接触型
品名（品番）	GemPC TWIN	PD2992P
準拠規格	ISO/IEC7816	ISO/IEC 14442 Type B
インタフェース	USB 2.0	USB 1.1

Java Card VM 上に搭載する認証用アプリケーションは，Java Card Technology を使って既に開発したもので[2][3]，カード内には PIN コード，ユーザ証明書，ユーザの私有鍵が格納されている．なお，ユーザ証明書は，X.509 標準規格に従い，ASN.1, DER フォーマットでエンコードされたもの，私有鍵に関しては 1024 ビット長の RSA 暗号鍵を格納している．

なお，本実装は Java Card VM 上の Java Card アプリとそれに対応するミドルウェアで構成されているが，IC カード機能として必要な要件は，証明書の格納と PKI 認証に必要な暗号処理機能を満たすことで

あり，Java Card™ 仕様に限定されるものではない．

5.2 ミドルウェア構成および共有端末への実装

開発されたスマートカードログオン用ミドルウェアは，認証，暗号化に関わる API として Windows の CriptoAPI を使用，その他 IC カードとの通信，LDAP クライアントとしての通信プロセスに関しても WIN32 API を利用している．開発環境は，VC++ ver.6 および Platform SDK を使用した．

一方，本検証では仮想共有端末として2台のデスクトップマシン（ThinkCentreA52T および DELL Precision 650 で OS は Windows 2000 Professional SP4）を利用している．それぞれのマシンに，今回開発したスマートカードログオン用ミドルウェアをセットアップし，winlogon プロセスから参照されるようにリポジトリ情報をレジストリ HKEY_LOCAL_MACHINE ¥SOFTWARE¥MyGina に登録している．（図 13）

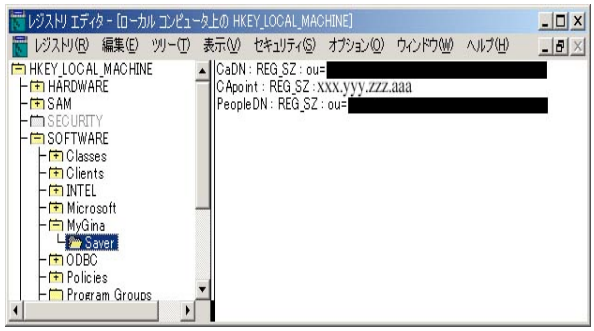

図 13 GINA レジストリ登録パラメータ

5.3 認証局，証明書

本システムを検証するに当たり，プライベート認証局，および CA 証明書や失効リスト（CRL）を格納するためのリポジトリサーバを1台準備した．サーバ用ハードおよび OS の仕様は，DELL Power Edge 2850 3.8GHz Xeon プロセッサ，Cent OS5，認証局は NAREGI-CA[8]をセットアップして，CA 証明書，ユーザ証明書，CRL の発行を行っている．

また，各証明書，CRL のリポジトリとして同サーバに OpenLDAP2.3[9]を実装し，CA 証明書，CRL とともに仮想ユーザ情報も格納している．なお，図 14 はリポジトリに格納された仮想ユーザ情報をLDAP 用ブラウザにより表示したものである．

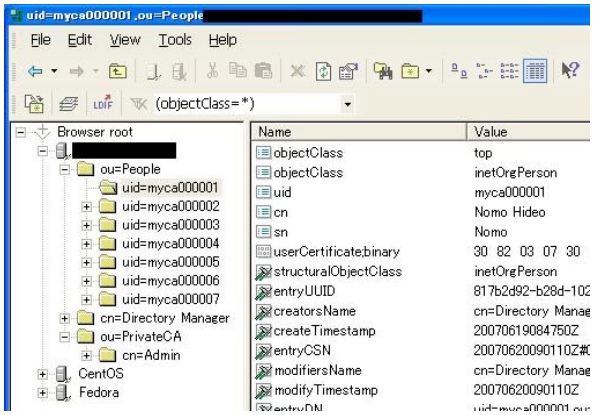

図 14 LDAP ブラウザによるリポジトリ情報の確認

5.4 プロファイル格納用ストレージ

ユーザプロファイル格納用ストレージは，本システムにおいては前述の認証局，リポジトリと共用のものとして設定した．したがって，実際には同サーバ上にプロファイル格納用にディレクトリーを確保し ssh プロトコルを通してデータ通信を行うものとした．なお，これらの設定は，システム検証のために簡易的に行っているものである．

6 実証実験

以上のように構築したシステムに対して，実際のネットワーク環境を通した実証実験を試み，その使い勝手や機能について検証した．実験は，異なる 2 ユーザ（2 枚の IC カードに対して個別のユーザ証明書，私有鍵を格納），異なる仮想共有端末 2 台を準備して，

1 異なるユーザが同一の共有端末からログオンした場合のそれぞれの作業環境の構築状況

2 同一ユーザが異なる共有端末からログオンした場合の作業環境の保持状況

の 2 点に着目し検証を行った．

また，本システムではローカルマシンへの通常ログオンと比較して，プロファイルの取得，格納プロセス時間分だけログオン時間がかかることが予想される．そのことに対するアクセス時間の比較も同時に行い使用上の改善点の抽出も試みた．

6.1 異なるユーザの同一共有端末作業環境

図 15 は共有端末起動後の IC カードの挿入を促す画面から PIN コード入力，さらに 2 枚の異なる IC

カード挿入に対応したそれぞれのデスクトップの画面を示している．作業環境の差異を明らかにするためにデスクトップの背景イメージとして異なる画像を用いており，ユーザに対応した環境が再現されていることが視認されるが，スタートアップメニュー，ブラウザのパラメータ，履歴，ブックマーク等各種作業環境がユーザごとに保持され，さらに変更分が格納されていることも確認することができた．

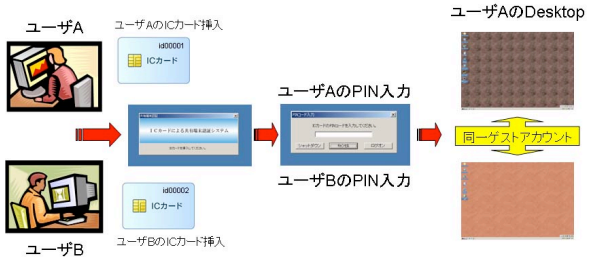

図　15 別ユーザ同一共有端末のデスクトップ比較

6.2 同一ユーザの遠隔地共有端末作業環境

図 16 は異なる 2 台の共有端末に同じ IC カードを挿入した場合の作業環境構築状況を確認したものである．この場合，2 台の共有端末はそれぞれ 1600×1200，1024×768 の異なる解像度を持つディスプレーを備えているが，いずれかの異なる解像度で設定された作業環境であってもシステム側で自動的に適応することが確認できた．その他の作業環境パラメータに関してもそれぞれの変更がストレージデータとして格納されていることも確認できた．

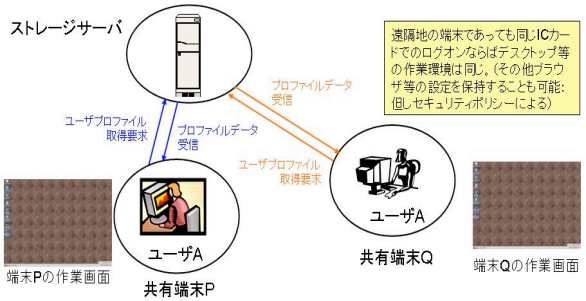

図　16 同一ユーザ別共有端末のデスクトップ比較

6.3 ログオン，ログオフ時間の比較

先にも述べたように，本システムではストレージサーバからユーザプロファイルデータを取得，格納するプロセスがログオン認証時間の遅れを引き起こすことが予想される．通常の利用状態で，作業環境そのものを表すプロファイル情報の容量は数百 KB から数 MB であるが，それらのデータ通信がログオン，ログオフ時間にどの程度影響するかを調べた．

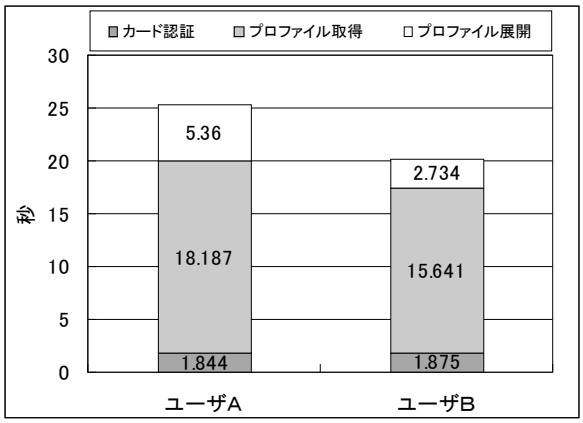

図　16 ログオンプロセスにおけるアクセス時間

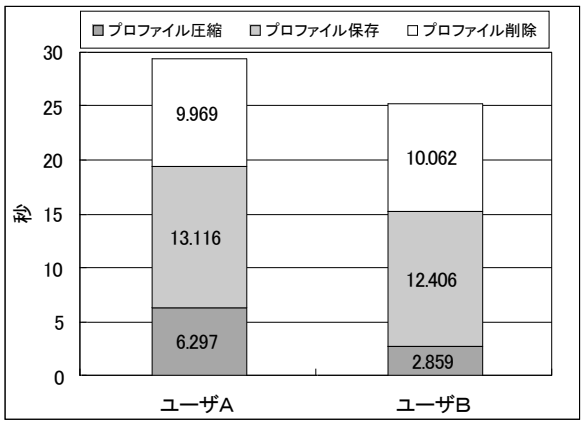

図　17 ログオフプロセスにおけるアクセス時間

図 16，17 は，それぞれログオン時とログオフ時のプロセス時間データをグラフに示したものである．それぞれは，プロファイル容量の異なる 2 枚のカードユーザとしてユーザ A（プロファイル容量≒9.3MB）とユーザ B（プロファイル容量≒7.6MB）のプロセス時間を比較している．ここで，図 16 において，グレー（濃）は IC カード認証プロセス時間，グレー（淡）はプロファイル取得プロセス時間，白はプロファイル展開（解凍）プロセス時間を示し，図 17 において，グレー（濃）はプロファイル圧縮プロセス時間，グレー（淡）はプロファイルのストレージ保存時間，白はプロファイル削除時間を示している．本システムにおけるネットワークスピードは共有端末－ストレージサーバ間で約 11.5MB/秒（ダミーファイルの ftp 転送測定結果より）である．

各プロセス時間の測定結果から，プロファイルのローカルシステムへの展開やデータ通信，データ削除などの各プロセスが作業時間へ与える影響は，ログオン，ログオフいずれにおいても無視できないことが確認された．なお，本実験で用いたプロファイル情報は，図 7 で示したユーザプロファイル構成の全てを対象としているが，共有端末の利用方法をふまえて，最終的には，ストレージデータとして保存すべき情報を選択する必要があると考える．

7 まとめ

共有端末利用時の利便性と経済性を考慮した利用形態として，IC カード認証と連携した非ドメイン型移動ユーザプロファイルの考え方を導入し，ドメイン構築を行うことなく，ユーザごとにどの共有端末からでも自身の作業環境が再構築できるシステムを開発した．実証実験を通した結果から，本システムを使ってストレージサーバへのユーザプロファイルの格納と作業環境の再構築が実用的に機能していることを確認した．しかしながら，ユーザプロファイルの容量が増えた場合や大規模マルチユーザ環境下でのログオン，ログオフ時アクセス時間の問題，個別ユーザが所有するユーザプロファイル以外のデータの保存場所，個別ユーザのアクセスログ管理の問題などが今後の課題として残されている．また，本システムでは，ユーザプロファイルとストレージ格納データの紐付けを IC カードのユーザ ID 情報にもとづいて行っているが，紐付けの方法によっては生体認証との連携の可能性もあり，これも将来的な検討課題となるであろう．

謝辞

本研究は，国立情報学研究所の最先端学術情報基盤（CSI）事業の一環として行われたものである．ここに記して謝意をあらわす．

参考文献

[1] Zhiqun Chen, “Java Card™ Technology for Smart Cards”, Addison Wesley.

[2] 葛生和人，平野晴，間瀬健二，渡邊豊英，*IC カードによる共有端末認証システムの構築*，第 35 回 コンピュータセキュリティ（CSEC）研究発表会研究報告，No.2006-CSEC-035, pp.45-50.

[3] 葛生和人，*PKI と連携したスマートカードログオンについて－共有端末における個人認証システムへの適用－*，名古屋大学情報連携基盤センターニュース，Vol.6, No.1, 2007, pp.27-40.

[4] ユーザプロファイルの概念，http://www.microsoft.com/technet/prodtechnol/windowsserver2003/ja/library/ServerHelp/20f61c10-0b87-41c9-a343-b4342c5562e8.mspx

[5] 武田保真，「徹底解説 Samba LDAP サーバ構築」，技術評論社

[6] GINA, http://msdn.microsoft.com/msdnmag/issues/05/05/SecurityBriefs/

[7] SID, http://support.microsoft.com/kb/243330/ja

[8] T.Okuno,”New open source CA development as Grid research platform”, http://www.naregi.org/papers/data/ggf12-caops_pki.pdf, Global Grid Forum, 2004.

[9] OpenLDAP, http://www.openldap.org/