スマートホンをセキュアデバイスとして用いるリモート接続システムの開発と評価

梅澤　克之†　　手塚　悟††

Development and Evaluation of a Remote Access System Using Smart Phone as a Secure Device

Katsuyuki UMEZAWA† and Satoru TEZUKA††

あらまし　近年，企業において多発している情報漏えい事故に対して，シンクライアントソリューションのニーズが高まっている．記憶領域をもたないノート PC を用いて，社内システムにリモート接続し，情報そのものは社外に出さないという運用方法である．このようなシンクライアントシステムにおいては，社外から社内に接続する際の認証機能が重要となる．従来はノート PC などに IC カードリーダをつなげ，個人用の IC カード（社員証等）やフラッシュメモリ型のセキュアデバイスを用いて個人認証を行い社内への接続を許可することが行われていた．本論文では，新規に開発した microSD 型のセキュアデバイスをスマートホンに装着し，スマートホンとノート PC 間で無線 (Bluetooth) を用いて IC カードコマンドの送受信を行い VPN 接続を行う Bluetooth 連携型リモート接続システムの提案を行う．具体的にはスマートホンを PC と連携させる際に，CSP (Cryptographic Service Provider) と，PC/SC (Personal Computer/Smart Card) という二つのインタフェースを Bluetooth で送受信できるデータ列に変換して連携を行う方法を提案し，開発したシステムの性能を計測し，実測値を評価する．ノート PC に直接 IC カードリーダをつなげる従来の方法と比較して，CSP 連携方式はわずか 0.85 秒の増加で，ユーザの使い勝手の大幅な向上を実現した．

キーワード　携帯電話，スマートホン，リモート接続，認証

1. まえがき

近年，企業における情報漏えい事故が多発しており，情報管理の徹底が求められている．情報漏えい事故の事例としては，自宅に空き巣が入り，個人情報が保存された個人所有 PC が盗難されたり，ノート PC や USB メモリの入った鞄を帰宅中の電車内で盗難されたりする事故が報告されている．これらに共通していることは，情報を社外へ持ち出しているということである．このような情報漏えい事故への対策として，シンクライアントソリューションのニーズが高まっている [1]〜[3]．記憶領域をもたないノート PC を用いて，社内システムに VPN (Virtual Private Network) プロトコル [4] で接続し，情報そのものは社外に出さないという運用方法である．このようなシンクライアントシステムにおいては，社外から社内に接続する際の認証機能が重要となる．従来はノート PC などに IC カードリーダをつなげ，個人用の IC カード（社員証等）やフラッシュメモリ型のセキュアデバイスを用いて個人認証を行い社内への接続を許可することが行われていた．このようなリモート接続を行う場合，セキュアデバイスやそのリーダを鞄の中から探し出し，PC に接続し準備をするのに手間が掛かっていた．その手間を省くために，セキュアデバイスを PC のスロットに常時接続しておくことも考えられるが，上述のように，盗難や紛失の際に PC とともにセキュアデバイスも同時になくなることになり，安全性の観点で推奨されない．本論文では，新規に開発した microSD 型 KeyMobile を装着したスマートホン(注1)を用いて，

† （株）日立製作所システム開発研究所，横浜市
Hitachi Ltd., Systems Development Laboratory, 292 Yoshida-cho, Totsuka-ku, Yokohama-shi, 244–0817 Japan
†† 東京工科大学，八王子市
School of Computer Science, Tokyo University of Technology, 1404–1 Katakuramachi, Hachioji-shi, 192–0982 Japan

(注1)：本提案はスマートホンだけに限らず一般的な携帯電話に対しても適用できる技術である．

スマートホンとノート PC 間で無線 (Bluetooth) を用いて IC カードコマンドの送受信を行い VPN 接続を行う Bluetooth 連携型リモート接続システムの提案を行う．具体的にはスマートホンを PC と連携させる際に，CSP (Cryptographic Service Provider) と，PC/SC (Personal Computer/Smart Card) という二つのインタフェースを Bluetooth で送受信できるデータ列に変換して連携を行う方法を提案し，開発したシステムの性能の計測及び評価を行う．以下では，まず，**2.** で従来技術として KeyMobile の概要，及び CSP と PC/SC の概要，更に従来型のリモート接続システムの概要について記述する．**3.** でスマートホンを用いたリモート接続の提案を行う．**4.** で性能を評価し，**5.** で性能に関する考察を行う．そして最後に **6.** でまとめと今後の課題を示す．

2. 従来技術

従来技術として，IC カード機能とフラッシュメモリ機能を併せ持つ KeyMobile の概要，暗号及び IC カード関連の標準的な機能としての CSP と PC/SC，USB リーダを介して KeyMobile を用いてリモート接続を行う従来型リモート接続の概要を示す．

2.1 KeyMobile の概要

KeyMoble は，IC カード機能とフラッシュメモリ機能を併せ持つセキュアデバイスである．KeyMoble の内部構造を図 1 に示す．KeyMobile の IC カードマイコン内には電子証明書 [5]〜[8] を格納し，PKI 技術に基づいた認証や署名などの機能を利用することができる．KeyMobile がもつ主な特徴は下記のとおりである．

- クレジットカードと同等の IC チップとフラッシュメモリを搭載
- IC チップ内に電子証明書をあらかじめ格納済み
- VPN アクセス時のクライアント認証が可能

このほかに，電子証明書の利用には PIN（暗証番号）入力を必要とし，PIN を複数回誤ると KeyMobile が自動的にロックする機能を有する．また，管理者のポリシーにより，PIN がロックするまでの回数を「5〜255 回」あるいは「無制限」へ変更可能である．

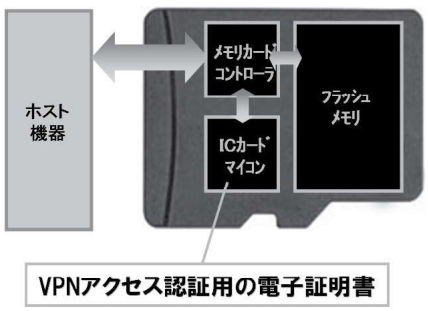

図 1　KeyMobile の内部構造
Fig. 1　Internal structure of KeyMobile.

2.2 CSP と PC/SC

暗号及び IC カード関連の標準的な機能として CSP と PC/SC について示す．

2.2.1 CSP (Cryptographic Service Provider)

CSP とは，CryptoAPI [9] というインタフェースで呼び出される暗号エンジンで，アプリケーションやシステムの要求に応じて暗号化やディジタル署名などの機能を提供する．CSP 関数の一覧（抜粋）を表 1 に示す．

2.2.2 PC/SC (Personal Computer/Smart Card)

PC/SC とは，異なるベンダの IC カードやリーダライタを Windows プラットホームで相互運用できるように策定された統一仕様である．PC/SC の仕様は以下の八つのパートから構成されている．

Part 1:　PC/SC の導入とアーキテクチャ概要
Part 2:　PC/SC 準拠 IC カードとインタフェースデバイス (IFD) へのインタフェース要件
Part 3:　PC に接続される IFD 要件
Part 4:　IFD 設計に関する考慮点と参照設計情報

表 1　CSP 関数一覧（抜粋）
Table 1　List of CSP functions (excerpt).

CSP 関数	意味
CPAcquireContext	CSP 内の特定の鍵コンテナのハンドルを取得する．
CPReleaseContext	CPAcquireContext で取得したハンドルを開放する．
CPDestroyKey	鍵を破棄する．
CPExportKey	アプリケーションのメモリスペースで鍵を CSP から転送する．
CPGetKeyParam	鍵のパラメータを検索する．
CPGetUserKey	鍵交換後，あるいは署名鍵のハンドルを取得する．
CPCreateHash	ハッシュオブジェクトを生成して，そのハンドルを返す．
CPDestroyHash	ハッシュオブジェクトを破壊する．
CPSetHashParam	ハッシュオブジェクトのパラメータをセットする．
CPSignHash	指定したハッシュオブジェクトに署名する．

表 2　PC/SC Part5 (WinSCard) 関数一覧（抜粋）
Table 2　List of PC/SC Part 5 (WinSCard) functions (excerpt).

PC/SC 関数	意味
SCardBeginTransaction	カードに対して論理トランザクションを開始する.
SCardConnect	リーダに入っているカードとの接続を開く.
SCardDisconnect	カードからの接続を断つ.
SCardEndTransaction	論理トランザクションを終了させる.
SCardEstablishContext	IC カードリソースマネージャとの通信で使用されるコンテクストを作成する.
SCardReleaseContext	SCardEstablishContext で作成されたコンテクストを解放する.
SCardTransmit	カードにデータを送り，カードから帰ってくるデータを受け取る.

Part 5:　IC カードリソースマネージャの定義
Part 6:　IC カードサービスプロバイダインタフェースの定義
Part 7:　アプリケーション分野及び開発者が設計する上での考慮点
Part 8:　セキュリティ及びプライバシーのための IC カードデバイス実装に関する推奨

Windows 環境では，Part 5 [10] で規定されるインタフェースが，Windows スマートカード (WinSCard) クライアント API として利用可能となっている．表 2 に PC/SC Part5 (WinSCard) 関数の一覧（抜粋）を示す.

2.3　従来型リモート接続

2.3.1　従来型リモート接続の概要

本項では，従来型のリモート接続の概要を説明する．現状では，シンクライアント PC に RS-MMC (Reduced Size MultiMedia Card) 型の KeyMobile 用のリーダを接続し，リーダに KeyMobile を挿すことによって社外から社内へリモート接続を行っている [1], [2]．従来型の接続例を図 2 に示す.

2.3.2　従来型リモート接続のフロー

従来型のリモート接続の関数呼出しを含むフローの概略を図 3 に示す．シンクライアント PC 内の上位アプリケーションが CSP を呼び，CSP が，PC/SC で規定されている SCardBeginTransaction や SCardTransmit などの関数を呼び出すことで，PC/SC ライブラリを経由して IC カードとのコマンドの送受信が行われる.

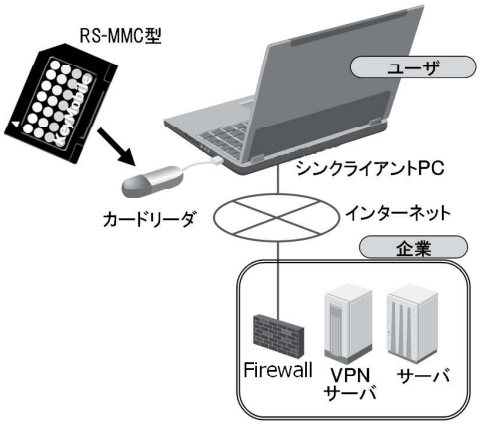

図 2　従来型のリモート接続システム
Fig. 2　Conventional remote access system.

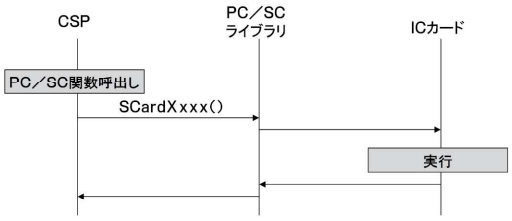

図 3　従来のリモート接続のフロー
Fig. 3　Flow of conventional remote access system.

3. 提 案 方 式

前章に示したように，従来は，シンクライアント PC にリーダを接続し，IC カードアクセスを行っていた．このような使い方の場合，リーダや IC カードの準備に手間が掛かるため使い勝手が悪かった．これを解決するために，セキュアデバイスを PC のスロットに常時接続しておくことも考えられるが，盗難や紛失時に，PC と同時にセキュアデバイスもなくなることになり，安全性の観点で推奨されない．そこで，筆者らはスマートホンや携帯電話機に直接挿せる microSD 型の KeyMobile を開発した．これにより，スマートホンを胸ポケットや鞄の中に入れておくだけで，IC カードリーダを準備する必要もなく，リモート接続を行うことを可能とした.

3.1　提案方式の全体アーキテクチャ

図 4 に提案システムの概要を示す．本提案システムでは，シンクライアント PC と KeyMobile を装着したスマートホンを Bluetooth で連携し，インターネットを経由して社内システムに接続する.

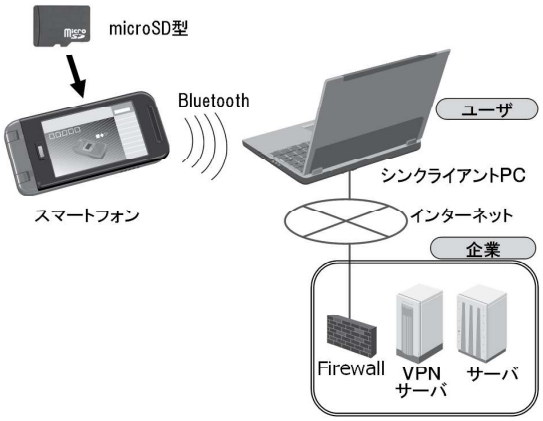

図 4　Bluetooth 連携型リモート接続システム
Fig. 4　Remote access system of Bluetooth federation type.

スマートホンとシンクライアント PC の間の通信は，Bluetooth 以外にも Wi-Fi や ZigBee などの近距離無線通信が考えられる．Bluetooth は多くの機器に対応しているという点や，通信距離が 10 m と比較的短いために今後の拡張において離席時の自動ロックなどにも対応できる可能性があるという点等を考慮して，今回の提案では Bluetooth を採用した．

また，そのときの近距離無線通信路上の安全性に関しては，Bluetooth 仕様で規定されている暗号化機能を用いるものとする．具体的には，本提案システムを稼動させる前段階でスマートホンとシンクライアント間で事前に PassKey を共有し，その鍵で暗号化通信を行うことで近距離無線通信路の安全性を確保することとする．

提案方式の全体アーキテクチャを図 5 に示す(注2)．CSP は暗号関連に特化した機能を提供するのに対して，PC/SC は暗号関連以外のコマンド送受信を含めて IC カードへの一般的なアクセス手段を提供する．図 5 に示すように，CSP は IC カードに関連した処理を実行する場合には PC/SC を呼び出す．このように IC カード関連の機能に関しては PC/SC の方が一般的で利用範囲が広い．一方 CSP は暗号関連に特化しているので機能をまとめて転送することで効率的に PC とスマートホンで連携できる可能性がある．このような考察に基づき，本論文では，PC とスマートホンを連携させる方式として CSP の関数を Bluetooth 通信で連携させる方式（CSP 連携方式）と，PC/SC の関数を Bluetooth 通信で連携させる方式（PC/SC 連携方式）の 2 通りの方式を提案し，実装・評価を行う(注3)．

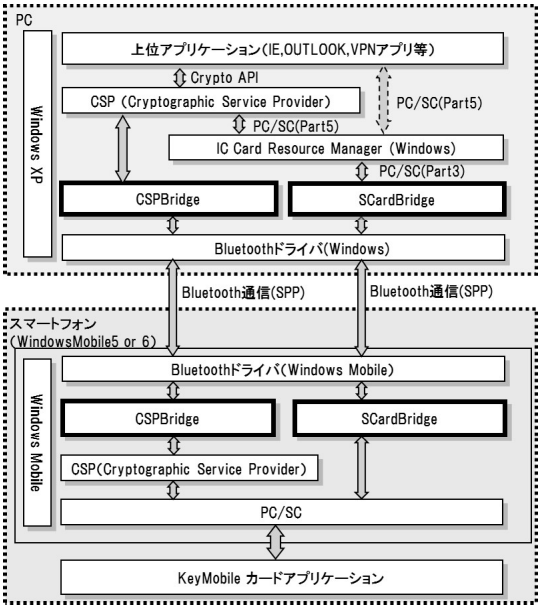

図 5　提案方式の全体アーキテクチャ
Fig. 5　Total architecture of proposal method.

3.2　CSP 連携方式

CSP 連携方式は，PC 側の上位アプリケーションが CryptoAPI を呼び出した後，CSP が直接 Bluetooth 連携機能（図 5 の CSPBridge）を呼び出し，CSP-Bridge は，CSP 関数を Bluetooth で送受信可能なようにシリアル化（バイト列化）して，Windows の Bluetooth ドライバ経由でスマートホンへコマンドを送信する．Windows Mobile の Bluetooth ドライバ経由でコマンドを受け取ったスマートホン側 CSPBridge が，シリアル化されたコマンドを元に戻して，スマートホン内の CSP インタフェースで KeyMobile にコマンドを伝達する．CSP 連携方式のフローを図 6 に示す．

次に，Bluetooth で送受信するデータの TLV (Tag-Length-Value) 構造化について示す．図 7 は CPAcquireContext に対応する CryptAcquireCon-

(注2)：図 5 中の上位アプリケーションと IC Card Resource Manager を結ぶ PC/SC (Part5) インタフェース（破線の矢印で表示）は，上位アプリケーションが暗号関連以外の一般的な IC カードコマンドを送受信する場合に呼び出すインタフェースである．本論文では，VPN 接続（暗号関連）を対象にしているため直接上位アプリケーションからこの PC/SC (Part5) インタフェースが呼ばれることはない．
(注3)：暗号エンジンのインタフェースとしては CryptoAPI 以外にも Cryptography Next Generation (CNG) [13] や，Public Key Cryptography Standards (PKCS #11) [14] などもある．今回の提案は，これら CryptoAPI 以外のインタフェースにも適用可能である．

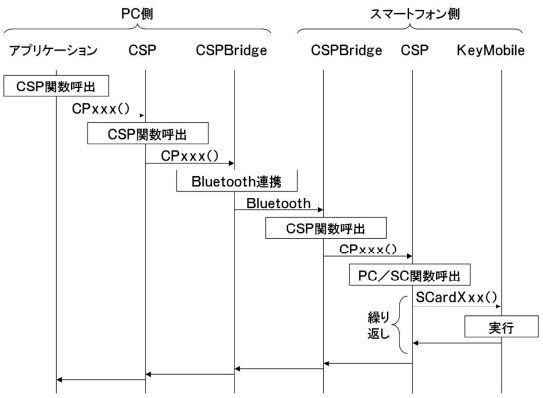

図 6　CSP 連携方式の処理フロー
Fig. 6　Flow of CSP federation method.

```
BOOL CRYPTFUNC CryptAcquireContext(
  HCRYPTPROV* phProv,          OUT
  LPCTSTR pszContainer,        IN
  LPCTSTR pszProvider,         IN
  DWORD dwProvType,            IN
  DWORD dwFlags                IN
);
```

図 7　CryptAcquireContext の関数定義
Fig. 7　Function definition of CryptAcquireContext.

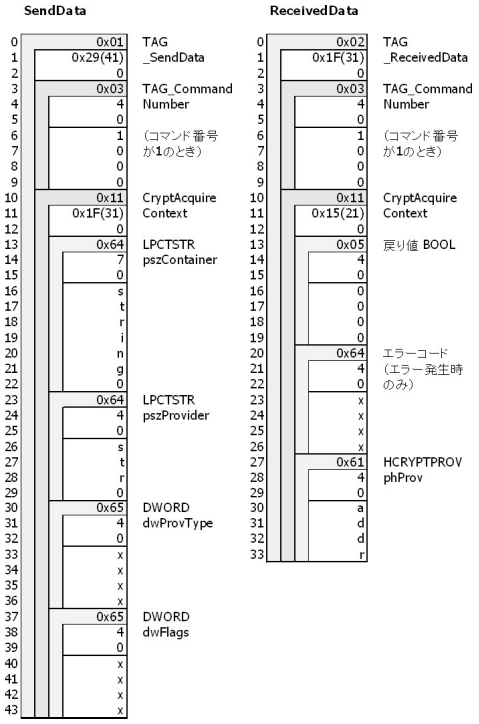

図 8　CryptAcquireContext の TLV 構造化
Fig. 8　TLV structure of CryptAcquireContext.

text (WindowsAPI) の関数定義である．図 8 は図 7 の関数定義を TLV 構造としてシリアル化した例である．SendData には，関数の種類を表す情報とともに，入力データとしての引数が TLV 構造化される．ReceiveData には，関数の種類，戻り値とともに出力データとしての引数が TLV 化されて返される．

3.3　PC/SC 連携方式

PC/SC 連携方式は，PC 側の上位アプリケーションが CryptoAPI を呼び出した後，CSP が PC/SC Part5 のインタフェースで IC カードリソースマネージャを呼び出す．その後，IC カードリソースマネージャが Bluetooth 連携機能（図 5 の SCardBridge）を呼び出す．SCardBridge は，PC/SC 関数を Bluetooth で送受信可能なようにシリアル化（バイト列化）して，Windows の Bluetooth ドライバ経由でスマートホンへコマンドを送信する．Windows Mobile の Bluetooth ドライバ経由でコマンドを受け取ったスマートホン側 SCardBridge が，シリアル化されたコマンドを元に戻して，スマートホン内の PC/SC インタフェースで KeyMobile にコマンドを伝達する．PC/SC 連携方式の処理フローを図 9 に示す．

Bluetooth で送受信するデータの TLV 構造化に関

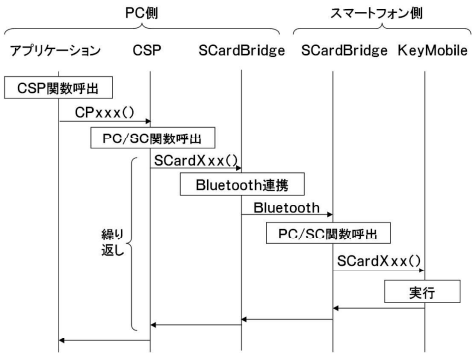

図 9　PC/SC 連携方式の処理フロー
Fig. 9　Flow of PC/SC federation method.

しては，CSP 連携方式で述べた方法と同様の方法で，それぞれの PC/SC 関数 (SCardXxxx) が TLV 化される．

3.4　ユーザインタフェース

本節では開発した Windows Mobile 側の CSP-Bridge 及び SCardBridge について記述する．これらのアプリケーションは，起動時にメモリ上に常駐し，タスクトレーにアイコン化され表示される．タスクト

図 10　ログ表示画面
Fig. 10　Screen of log display.

図 11　通信パラメータ設定画面
Fig. 11　Screen of communication parameter setting.

レーのアイコンをタップすることで最大化した状態で最前面に表示され，ログ表示及び通信パラメータ設定を行うことができる．図 10 にログ表示画面を，図 11 に通信パラメータ設定画面を示す．

4. 性能評価

本章では，CSP 連携方式，PC/SC 連携方式，及び，従来方式を用いて VPN リモートアクセスに要する時間を計測した．性能測定対象システムの構成は，図 2 及び図 4 に示したとおりである．性能評価で用いたスマートホン及びノート PC の機器構成を表 3 に示す．なお，測定には Bluetooth Ver.2.1 対応のアダプタを用いた．

表 4 に測定結果を示す．表 4 に示した CSP 関数の種類と順序は，PC 上の VPN クライアントアプリケーションが，VPN 接続を完了するまでに呼び出す CSP 関数の典型的な例であり(注4)，各方式における CSP 関数の処理に要する時間を示す．なお従来とは，他方式と同じ KeyMobile を装着したリーダを直接 PC に接続した構成での測定結果である．どの方式においても同一の KeyMobile カードを用いた．また，表 4 の結果は，各測定項目において 12 回測定し，全体の時間が最小と最大のデータを除外した 10 回分の測定値から平均値を算出した．

表 3　性能測定対象システム構成機器
Table 3　Specs of equipment used for evaluation.

	スマートホン	ノート PC
CPU	Qualcomm QSD8250 (1 GHz)	Intel Core 2 Duo (2.4 GHz)
OS	Windows Mobile 6.5 Professional	Windows XP SP3
Memory	512 MB (ROM) 256 MB (RAM)	2 GB

表 4　測定結果
Table 4　Evaluation result.

CSP 関数	従来	PC/SC	CSP
CPAcquireContext	1.379	3.908	1.873
CPGetUserKey	0.086	0.461	0.169
CPExportKey	0.024	0.034	0.045
CPDestroyKey	0.024	0.033	0.045
CPReleaseContext	0.217	0.805	0.295
CPAcquireContext	1.331	3.861	1.871
CPGetUserKey	0.081	0.464	0.172
CPGetKeyParam	0.024	0.031	0.045
CPDestroyKey	0.030	0.030	0.041
CPCreateHash	0.024	0.025	0.036
CPSetHashParam	0.017	0.020	0.045
CPSignHash	0.047	0.047	0.039
CPSignHash	1.241	2.625	0.637
CPDestroyHash	0.070	0.034	0.040
CPReleaseContext	0.208	1.081	0.298
合計時間	4.802	13.460	5.652

表 4 に示したように，従来方式としてリーダを直接 PC に接続する方式に比べて CSP 連携方式で 0.85 秒，PC/SC 連携方式で約 8.66 秒遅くなるという結果となった．しかし従来方式の場合には，KeyMobile カード及びリーダを鞄の中などの他の携行品の中から探し出し PC に接続するという煩雑な作業をユーザに強いているのに対して，提案方式の場合には，その手間を省くことができるためユーザの利便性は大幅に向上しているということができる．

(注4)：IC カードに搭載されている電子証明書の数や PIN の要求の有無などの環境の違いにより呼び出される回数や関数の種類に違いが生じる場合がある．

5. 考　　察

本章では前章の測定結果のうち PC/SC 連携方式と CSP 連携方式の動作速度の違いについて考察する．研究開発当初，PC/SC 連携方式では，一つの CSP 関数に対して複数の PC/SC 関数が Bluetooth で送受信されるため，CSP 連携方式より効率が悪いであろうと考えた．実際に予備実験において，SCardBeginTransaction を 1 回，SCardTransmit を 100 回，SCardEndTransaction を 1 回行うような処理の場合には，CSP 連携方式の方が 32%高速であるという結論を得ていた [11]．しかし，表 4 より，CPExportKey，CPDestroyKey，CPGetKeyParam，CPCreateHash，CPSetHashParam，CPDestroyHash の関数は PC/SC 連携方式の方が CSP 連携方式より速い結果となった．本結果を考察するにあたり，実際に VPN 接続を行う際の認証処理において PC 上の VPN クライアントから呼び出される CSP 関数と，その CSP 関数がどんな PC/SC 関数を何回呼び出しているかの調査を行った．表 5 に結果を示す．

表 4，表 5，及び文献 [12] の結果から，CSP 関数の振舞いについて以下の 3 分類が可能である．

- IC カードでの処理が不要で PC/SC 関数を呼び出さない関数（ケース 1）
- 複数の PC/SC 関数から構成され，一括してスマートホン側で処理した方が高速な関数（ケース 2）
- 複数の PC/SC 関数から構成されるが，一括してスマートホン側で処理すると遅くなる関数（ケース 3）(注5)

まず，「IC カードでの処理が不要で PC/SC 関数を呼び出さない関数（ケース 1）」（表 5 の対応する PC/SC 関数欄に「なし」と表記した関数）に関しては，図 12 と図 13 に示すように，PC/SC 連携方式では PC 側だけで処理される関数が，CSP 連携方式では，Bluetooth で送受信されているために多くの時間がかかっていることが確認できる．これらの関数に関する解決策は比較的容易であり，PC 側の CSPBridge に当該関数をスマートホン側に転送しない機能を付加することで解決可能と考えられる(注6)．

次に，「複数の PC/SC 関数から構成され，一括してスマートホン側で処理した方が高速な関数（ケース 2）」に関しては，CPAcquireContext，CPGetUserKey，CPReleaseContext，CPSignHash 関数が該当する．本結果は，Bluetooth 通信を複数回繰り返し行うより，その上位関数をまとめて処理した方が高速になるはずであるという当初の想定及び文献 [11] に示した予備実験の結果に当てはまるものである(注7)．

表 5　CSP 関数と PC/SC 関数の対応関係
Table 5　Relation between CSP function and PC/SC function.

CSP 関数	PC/SC 関数	回数
CPAcquireContext	SCardEstablishContext	1
	SCardConnect	1
	SCardBeginTransaction	5
	SCardTransmit	19
	SCardEndTransaction	5
CPGetUserKey	SCardBeginTransaction	1
	SCardTransmit	2
	SCardEndTransaction	1
CPExportKey	なし	0
CPDestroyKey	なし	0
CPReleaseContext	SCardBeginTransaction	1
	SCardTransmit	1
	SCardEndTransaction	1
	SCardDisconnect	1
	SCardReleaseContext	1
CPAcquireContext	SCardEstablishContext	1
	SCardConnect	1
	SCardBeginTransaction	5
	SCardTransmit	19
	SCardEndTransaction	5
CPGetUserKey	SCardBeginTransaction	1
	SCardTransmit	2
	SCardEndTransaction	1
CPGetKeyParam	なし	0
CPDestroyKey	なし	0
CPCreateHash	なし	0
CPSetHashParam	なし	0
CPSignHash	なし	0
CPSignHash	SCardBeginTransaction	4
	SCardTransmit	15
	SCardEndTransaction	4
CPDestroyHash	なし	0
CPReleaseContext	SCardBeginTransaction	1
	SCardTransmit	1
	SCardEndTransaction	1
	SCardDisconnect	1
	SCardReleaseContext	1

(注5)：今回の実測ではこの分類に該当する結果はないが，文献 [12] の実測では多くの関数がこの分類に該当している．
(注6)：ただし，表 4 からも分かるようにこれらの関数の処理時間の差はわずかであるため実装上は現状のままで問題ないと考えられる．
(注7)：表 4 の下から三つ目の CPSignHash 関数に関して，従来方式が 1.241 秒に対して CSP 連携方式では 0.637 秒というように CSP 連携方式の方が速くなっている．これは，複数リーダに対応した PC 上の CSP と，単一リーダを前提としたスマートホン上の CSP の実装方法（PC 上の CSP の方が複雑な作りになっている）が原因と考えられる．また，表 4 の下から四つ目の CPSignHash 関数に関して PC/SC 連携方式が 0.047 秒に対して CSP 連携方式では 0.039 秒というように CSP 連携方式の方が速くなっている．この理由は定かではないが，前述の理由のように CSP の実装方法の違いによる誤差ではないかと考えられる．つまり，CSP 連携方式の場合は，何も判断せずに無条件に Bluetooth を呼び出しているのに対し，PC/SC 連携の場合は，複数リーダに対応した処理に時間がかかっていると考えることができる．

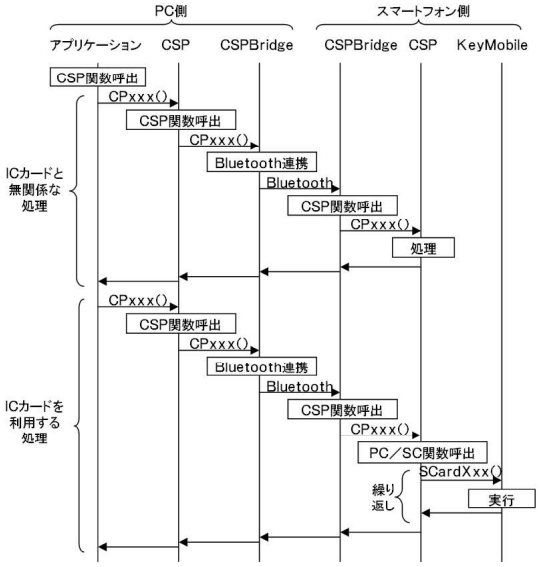

図 12　CSP 連携方式の実際の処理フロー
Fig. 12　Actual flow of CSP federation method.

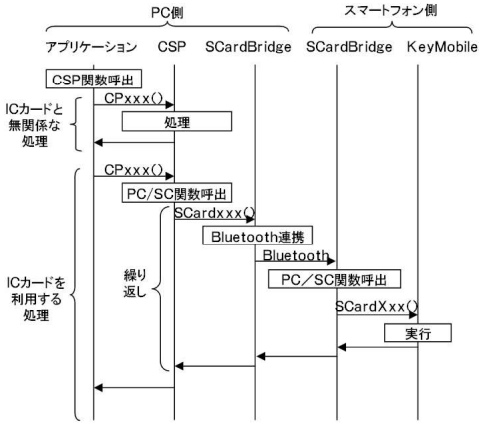

図 13　PC/SC 連携方式の実際の処理フロー
Fig. 13　Actual flow of PC/SC federation method.

最後に，「複数の PC/SC 関数から構成されるが，一括してスマートホン側で処理すると遅くなる関数（ケース 3）」に関して考察する．今回の実測環境では，複数の PC/SC 関数から構成されるが一括してスマートホン側で処理すると遅くなるような結果は得られなかった．文献 [12] では，多くの関数がこの分類に該当した．文献 [12] で評価に用いたスマートホンの性能が低く，IC カード内での処理以外の処理を PC 上で実行するよりもスマートホン上で実行する方が多くの時間を要している結果であるということができる．

つまり，今回の実験環境では，

- ケース 1：CSP 方式と PC/SC 方式の差はほとんどない．
- ケース 2：CSP 方式の方が性能が良い．
- ケース 3：該当なし．

という結果を得た．よって今回の実験環境のようにスマートホンの性能が十分である場合は，CSP 連携方式の方が性能が良いといえる．

6. むすび

本論文では，microSD 型の KeyMobile を装着したスマートホンを用いて，スマートホンとノート PC 間で無線 (Bluetooth) を用いて IC カードコマンド送受信を行い VPN 接続を行う Bluetooth 連携型リモート接続システムの提案を行った．具体的には，スマートホンを PC と連携させる際に，CSP と PC/SC の二つのインタフェースを Bluetooth で送受信できるデータ列に変換して連携を行う方法の提案を行った．更に開発したシステムを実測し，実測値に対する評価を行った．評価の結果，従来方式としてリーダを直接 PC に接続する方式に比べて，CSP 連携方式はわずか 0.85 秒の増加で，ユーザの使い勝手の大幅な向上を実現した．また，CSP 関数と PC/SC 関数の関係（CSP 関数がどんな PC/SC 関数を何回呼び出しているか）を調査することで，CSP 連携方式と PC/SC 連携方式の速度の違いを考察した．

今後は，スマートホンと PC の実行速度の差と，スマートホンと PC 間の通信速度の差に着目し，全体としての処理時間の分岐点を明らかにするための理論的な解析を行う．また，CryptoAPI 以外の Cryptography Next Generation (CNG) [13] や，Public Key Cryptography Standards (PKCS #11) [14] への適用も推進する．

謝辞　本研究は，（独）情報通信研究機構 (NICT) の委託研究「端末プラットフォーム技術に関する研究開発」の成果の一部である．

［商標に関する表示］

- Windows，Windows Mobile は米国 Microsoft Corporation の米国及びその他の国における登録商標である．
- Intel，Intel Core™ は，米国及びその他の国における，Intel Corporation またはその子会社の商標または登録商標である．
- Qualcomm は QUALCOMM Incorporated の登録商標である．

文　献

[1] 新井利明，田中輝雄，野田文雄，“情報漏えいを阻止する「セキュアクライアントソリューション」，” 日立評論，vol.88, no.5, pp.20–25, May 2006.

[2] 中西　潤，牧野一郎，小高　浩，杉山卓也，石原　修，“「セキュアクライアントソリューション」を支える「セキュリティPC」と周辺装置，” 日立評論，vol.88, no.5, pp.26–29, May 2006.

[3] 宮本久仁男，田中英彦，“シンクライアントアーキテクチャをベースにしたセキュアクライアントの検討，” IPSJ SIG Notes 2007(71), pp.305–310, July 2007.

[4] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Maris, RFC 2764: A Framework for IP Based Virtual Private Networks, IETF, Feb. 2000.

[5] ITU-T Recommendation X.509 ISO/IEC 9594-8: Information technology — Open Systems Interconnection — The Directory: Authentication Framework, 1997.

[6] R. Housley, W. Ford, W. Polk, and D. Solo, RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF, Jan. 1999.

[7] R. Housley, W. Polk, W. Ford, and D. Solo, RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF, April 2002.

[8] D. Cooper, S. Santesson, S. Farrell, R. Housley, and W. Polk, RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF, May 2008.

[9] Microsoft Corporation, Microsoft Cryptographic API, URL: http://msdn.microsoft.com/en-us/library/aa380252(v=VS.85).aspx

[10] Interoperability Specification for ICCs and Personal Computer Systems, Part 5. ICC Resource Manager Definition, Revision 2.01.01, PC/SC Workgroup, Sept. 2005.

[11] 梅澤克之，洲崎誠一，“スマートフォンを用いたリモート接続システムの開発，” 第 31 回情報理論とその応用シンポジウム予稿集，pp.971–974, Oct. 2008.

[12] 梅澤克之，加藤崇利，手塚　悟，“スマートフォンを用いたリモート接続システムの開発と評価，” 第 8 回情報科学技術フォーラム (FIT2009) 予稿集，第 4 分冊，pp.67–73, Sept. 2009.

[13] Microsoft Corporation, Cryptography API: Next Generation (CNG) URL: http://go.microsoft.com/fwlink/?LinkID=74141

[14] PKCS #11 v2.30: Cryptographic Token Interface Standard, RSA Laboratories, April 2009.

（平成 22 年 7 月 20 日受付，10 月 8 日再受付）

梅澤　克之　（正員）

1996 早稲田大学大学院理工学研究科機械工学専攻修士課程了．同年（株）日立製作所システム開発研究所入所．以来，分散オブジェクトシステム，モバイルセキュリティ技術，スマートカードセキュリティ技術などの研究・開発に従事．情報処理学会，電気学会各会員．博士（工学）．

手塚　　悟　（正員）

2009 より，東京工科大学コンピュータサイエンス学部教授，現在に至る．1984 より，（株）日立製作所入社．マイクロエレクトロニクス機器開発研究所に勤務し，パーソナルコンピュータのオペレーティングシステム，デバイスドライバ，LAN システム等の研究開発に従事．その後，システム開発研究所に勤務し，パーソナルコンピュータを中心とした LAN システムの構築・運用管理の研究開発，更に電子政府，電子自治体等を主に情報セキュリティシステムの研究開発に従事．特に，PKI 技術を用いた電子署名，電子認証等の研究．慶應義塾大学理工学部特別講師，大阪大学非常勤講師等歴任．2004 年度情報処理学会論文賞，2008 年度情報処理学会論文賞，IEEE-IIHMSP2006 Best Paper Award．工博．著書に「Inside CORBA」アスキー出版（共訳）(1998),「インターネットコマース　新動向と技術」共立出版（共著）(2000),「インターネット時代の情報セキュリティー　暗号と電子透かし」共立出版（共著）(2000)．