# 非接触型 IC カードを用いた認証プロトコル SPAIC の研究

宮 崎　 雄 介

クライアント/サーバ間通信において重要な情報を交換する場合，確実な認証と暗号化が必要となる．また，ユーザが自由に移動する環境においても認証と暗号化による情報配送を行いたいという要求がある．このような環境では，ユーザ固有の情報を格納した IC カードを利用する方式が注目されている．これまでは，接触型 IC カードを利用する場合がほとんどであり，IC カード/クライアント間通信のセキュリティはそれほど重要ではなかった．しかし，今後は非接触型 IC カードの普及が見込まれ，IC カード/クライアント間でも暗号通信を行うことが必須になると考えられる．これを実現するために，すべての IC カードとクライアントに同じ共通鍵を所持させるという方法があるが，クライアントから情報が流出するという懸念があった．本論文では，非接触型 IC カードを利用し，初期情報を一切持たないクライアントに重要情報を配送することを可能とするプロトコル SPAIC（Secure Protocol for Authentication with IC card）を提案する．

# Proposal of an Authentication Method “SPAIC” using a Non-contact Type IC Card

YUSUKE MIYAZAKI

When important information is exchanged in a client-server system, it is quite essential to perform reliable authentication and encryption. In addition, there is a demand to want to perform the certification and information delivery by the coding in the environment that a user moves freely. In such a situation, a method using an IC card that stores unique user information has been widely used. Until recently, the security of communication between an IC card and a client has not been a critical concern because contact type IC cards have been used in most cases. However, now that non-contact type IC cards are expected to be spreading in the near future, secure communication between an IC card and a client is considered to become a serious concern. Although there exists a method whereby all IC cards and clients use a common key to realize secure communication, there is a concern that secret information is easily leaked from the client terminal. To solve this problem, we propose in this study a protocol called “SPAIC”. Presuming that a non-contact type IC card is used, SPAIC can deliver the important information from a server to a client that has no initial information so that there exists no risk of information leakage.

## 1. はじめに

インターネットの発展に伴い，ユーザがクライアント端末を利用して遠隔地のサーバと情報交換したいという要求が高まっている．クライアント/サーバ間通信において重要な情報を交換する場合，強固な認証と暗号化が要求される．認証と暗号化による情報配送は，従来から様々な方式が検討されている[1]-[10]．近年では，ユーザが自宅あるいは会社など異なるクライアントからでもサーバにアクセスしたいというニーズが増えている．このような環境においても同様に認証と暗号化による情報配送を行えることが望ましい．このような要求を満たす方式の一つとして，ユーザが IC カードを所持する方式が注目されている[11]，[12]．IC カードは CPU やメモリを搭載し，内部で演算ができる．また，IC カードは耐タンパ性を有しており，認証に必要な情報を安全に格納することが可能である．従って，クライアント端末にユーザの情報を保持する必要がない[13]．すなわち，ユーザが端末を選べるという利便性を得ると同時に，端末からユーザの情報が盗まれるのを防止できるという利点もある．近年では非接触型 IC カードの発展により，IC カードの利便性が一層向上することが期待されている[14]．

IC カードを利用した認証方式では，クライアント/サーバ間で行われる認証に加えて，IC カードの持ち主を確認するためのユーザ認証も併せて行う必要がある．ユーザ認証は，IC カード内にパスワードなどのユーザ情報を格納し，クライアントから入力されたユーザ認証情報を IC カードの内部で検証する方法が主流である[15]．接触型 IC カードでは，IC カードとクライアントが一体であるため，両者の間の通信に係るセキュリティは大きな問題にはならなかった．しかし，非接触型 IC カード

を用いる場合，これらの認証処理に必要な情報を無線でやりとりするため，IC カード/クライアント間の暗号通信が必須である．

IC カード/クライアント間の通信を暗号化する方法として，事前共有鍵を利用する方式が JICSAP によって定義されている[16]．しかし，この方式では，すべての IC カードおよびクライアントに事前共有鍵を所持させるため，クライアント側から共有鍵が漏洩する危険性がある．さらに，漏洩した場合，影響がシステム全体に波及する可能性がある．クライアントは IC カードのような耐タンパ性を有していないのが一般的であるため，クライアントに秘密情報を所持させない方式が望ましい．

そこで，本論文では非接触型 IC カードを利用し，初期情報を一切持たないクライアントに対し，サーバから重要情報を配送することを可能とするプロトコル SPAIC（Secure Protocol for Authentication with IC card）を提案する．

SPAIC では，IC カード公開鍵を利用して，クライアントから IC カードへの通信の暗号化を行う．また，サーバ公開鍵を利用して IC カードからクライアントを経由し，サーバまで通信の暗号化を行う．更に，クライアント/サーバ間では Diffie-Hellman 鍵交換[17]-[19]により，動的に暗号鍵を生成し，サーバから安全に重要情報を配送することを実現する．

以降，2 章で既存技術とその課題，3 章で提案方式，4 章で SPAIC の実装，5 章で評価，6 章でまとめを述べる．

## 2. 既存技術とその課題

これまでは，接触型 IC カードを IC カードリーダに挿入して利用するような場合がほとんどであったため，IC カードとクライアントが一体のものであるとみなし，両者間の暗号化を行っていないものが殆どであった．しかし，非接触型 IC カードを利用する場合，IC カード/クライアント間が無線通信になるため，暗号化が必須となる．これを実現するための方式として，暗号通信の種となる共有鍵をすべての IC カード，クライアント端末に所持させる事前共有鍵方式が定義されている．事前共有鍵方式の例を図 1 に示す．この方式では，全ての端末，IC カードが共通の事前共有鍵を保持する．この鍵を用いて IC カード/クライアント間で利用する暗号鍵を動的に生成する．

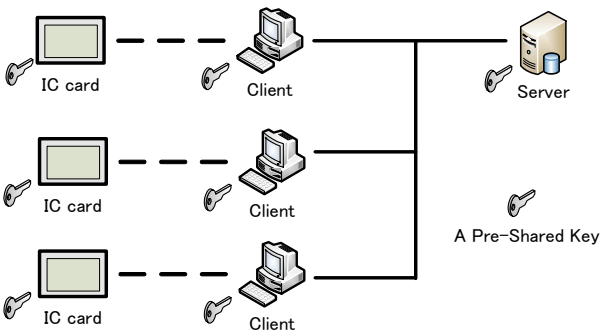


図 1 事前共有鍵方式の認証方式例

しかし，事前共有鍵方式では，クライアントに共通鍵を所持させる必要があるため，クライアントからの情報漏洩の危険性がある．更に，システム全体で同じ事前共有鍵を所持しているため，この共有鍵が漏洩した場合，その影響がシステム全体に波及する可能性がある．このため，システムの安全性を確保するためにはすべての IC カード，クライアントの事前共有鍵を定期的に変更する作業が必要となる．このような事情から，事前共有鍵は管理が煩雑で大規模システムへの適用が困難という課題がある．

## 3. SPAIC の提案

本章では，事前共有鍵方式の課題を解決するために，SPAIC（Secure Protocol for Authentication with IC card）を提案する．提案方式では，クライアントに秘密情報を一切所持させないモデルを定義する．この条件のもとで，サーバからクライアントへ暗号鍵など第三者に秘匿すべき重要情報を安全かつ確実に配送することを目的とする．

### 3.1 システムモデルと前提条件

本提案で想定するシステムモデルを図 2 に示す．矢印は認証の方向を示している．ユーザは個人情報を格納した IC カードを所持している．各クライアントには IC カードリーダが搭載されており，各ユーザに発行された IC カードを用いてユーザ認証を行う．ユーザ認証後，IC カードとサーバの間で相互認証を行い，クライアントへ重要情報を配送する．また，クライアントには認証動作と情報配送に必要となるプログラムだけを格納し，認証に必要となる秘密情報は一切所持させない．このためクライアントからの情報漏洩の心配がない．

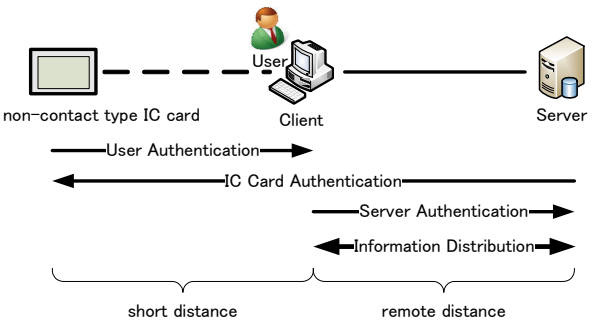


図 2 想定するシステムモデル

想定システムでは，以下のような前提をおく．

(1) 今後の普及を考え，非接触型 IC カードを利用する．
(2) IC カードは耐タンパ性を有し，IC カード内部情報が漏洩することはない．
(3) IC カード/クライアント間はユーザが確認できる程の近距離であり，中間者攻撃(Man-in-the-middle Attack)はできないものとする．

### 3.2 記号の定義

提案方式の説明に使用する記号を以下のように定義する．

uID：ユーザ ID
PW：パスワード
T：生体情報テンプレート
PSK：事前共有鍵（従来方式）
PrI, PuI：IC カード秘密鍵，公開鍵
PrS, PuS：サーバ秘密鍵，公開鍵
Ni, Nr：乱数
DH1, DH2：Diffie-Hellman 交換値
K：Diffie-Hellman 共通鍵
Ci, Cr：クッキー
$E_Y[X]$：データ X を鍵 Y で暗号化
$S_I(X)$：IC カードによるデータ X へのディジタル署名
$S_S(X)$：サーバによるデータ X へのディジタル署名
Key_REQ：鍵配送要求パケット
Key_RES：鍵配送応答パケット
Cookie_REQ：クッキー配送要求パケット
Cookie_RES：クッキー配送応答パケット
CertUser_DIST：ユーザ認証情報配送パケット
SignIC_DIST：IC カード署名情報配送パケット
Info_DIST：情報配送パケット
SignMS_DIST：サーバ署名情報配送パケット

### 3.3 各端末の初期情報

事前共有鍵方式と SPAIC が所持する初期情報を表 1 に示す．ユーザ認証にはパスワードと生体認証を用いるものとする．

事前共有鍵方式では，各ユーザが所持する IC カードには，IC カード固有の ID（uID），IC カード秘密鍵 PrI，サーバ公開鍵 PuS，パスワード PW，生体情報テンプレート T が格納されている．サーバには，サーバ秘密鍵 PrS，各 IC カードの ID（uID）と公開鍵 PuI が格納されている．また，IC カード/クライアント間の通信を暗号化するため，事前共有鍵 PSK をすべての IC カード，クライアント端末に所持させる．

**表 1. 事前共有鍵方式と SPAIC の初期情報**

| | PSK Method | SPAIC |
|---|---|---|
| IC crad | uID<br>PrI<br>PuS<br>PW<br>T<br>**PSK** | uID<br>PrI<br>PuS<br>PW<br>T<br>**PuI** |
| client | **PSK** | - |
| Server | uID<br>PrS<br>PuI | uID<br>PrS<br>PuI |

SPAIC の場合は，IC カードには，共有鍵 PSK に代わり，IC カード公開鍵 PuI を格納する．クライアントには初期情報を一切所持しない．その他の初期情報は事前共有鍵方式と同様である．表 1 に示す初期情報はサーバ側で一括して作成し，IC カードの発行はあらかじめオフラインで実施しておく．IC カード公開鍵 PuI は IC カード秘密鍵 PrI と同時に生成するものであり，この情報を IC カードに格納することによって管理負荷が増えることはない．

### 3.4 SPAIC の認証動作概要

SPAIC の認証動作概要を図 3 に示す．SPAIC では IC カード/クライアント/サーバを独立したものとして環状の認証を行うため，認証動作は三段階よりなる．ユーザはクライアントを操作しているため，両者は一体のものとみなす．また，予めクライアントの電源を投入し，SPAIC のプログラムを起動しておく．

第一段階として，IC カードがユーザ認証を行う．ユーザがサーバへアクセスするために，IC カードを IC カードリーダにかざすと，クライアントとの間にコネクションが確立される．IC カード公開鍵 PuI，サーバ公開鍵 PuS がクライアントに送信される．クライアントにはパスワード入力画面が表示される．ユーザは，ユーザ認証情報となるパスワード PW や生体情報 T をクライアントに入力する．クライアントではユーザ認証情報を IC カード公開鍵 PuI で暗号化し，更に Diffie-Hellman 鍵交換の交換値（DH1）を生成する．これらの情報を IC カードへ送信する．IC カードでは IC カード秘密鍵 PrI を用いてユーザ認証情報（PW や T）を取り出し，内部に保持している秘密情報と照合することによりユーザ認証を行う．上記手順により，間接的にユーザが使用しているクライアントを認証したことになる．

次に第二段階として，サーバが IC カード認証を行う．IC カードは IC カード秘密鍵 PrI を用いて，DH1 にディジタル署名を付加し，ユ

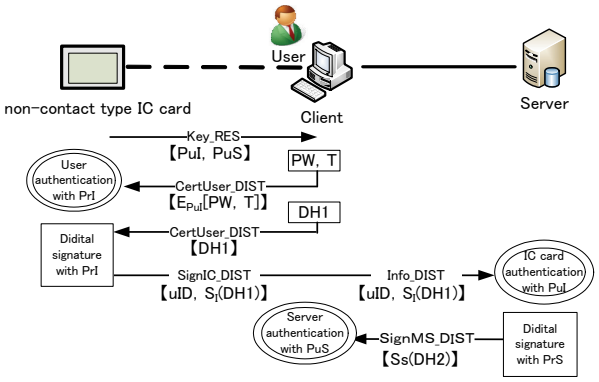


図 3 SPAIC の認証動作概要図

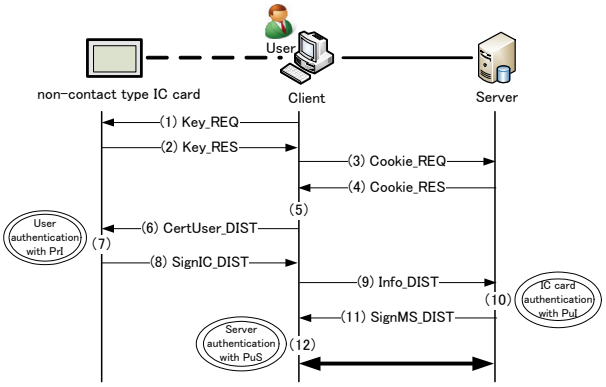


図 4 SPAIC の詳細シーケンス図

ーザ ID（uID）とともにクライアント経由でサーバへ送信する．サーバでは受信した uID から対応する IC カードの公開鍵 PuI を読み出し，ディジタル署名の検証を行い，IC カードを認証する[20]．IC カードはユーザを認証済みなので，間接的にユーザが使用しているクライアントを認証したことになる．サーバは同時に DH1 を取得する．

最後に第三段階として，クライアントがサーバ認証を行う．サーバは DH 交換値（DH2）を生成し，サーバ秘密鍵 PrS を用いてディジタル署名を行いクライアントへ送信する．クライアントでは，IC カードから受信したサーバ公開鍵 PuS を利用してディジタル署名の検証を行い，サーバを認証する． 以上の 3 段階の認証により，クライアント/サーバ間の認証が完了する．上記手順の中で DH1，DH2 の共有が行われているため，クライアント，サーバは共通暗号鍵 K を生成できる．以降のクライアント/サーバ間の通信はこの暗号鍵 K を用いて行う．

### 3.5 SPAIC の詳細なシーケンス

IC カード/クライアント/サーバ間の認証は 3.4 節の手順で達成される．しかし，認証の過程において攻撃者の存在を考慮しなければならない．本節では，Dos 攻撃（Denial of Service attack）やリプレイ攻撃（replay attack），中間者攻撃（man-in-the-middle attack）などの対策を含めた SPAIC の詳細な認証処理を述べる．SPAIC の詳細なシーケンスを図 4 に示す．

(1) Key_REQ の送信
　クライアントはユーザ認証情報などの暗号化を行うために，IC カードへ公開鍵等の情報配送を要求する．

(2) Key_RES の送信
　IC カードは乱数 Ni を生成し，ユーザ ID(uID)，IC カード公開鍵 PuI，サーバ公開鍵 PuS と共にクライアントへ送信する．
　Key_RES = uID|PuI|PuS|Ni

(3) Cookie_REQ の送信
　クライアントは DoS 攻撃を防止するためのクッキーCi を生成して，サーバへ送信する．
　Cookie_REQ = Ci

(4) Cookie_RES の送信
　サーバは乱数 Nr とクッキーCr を生成する．また，クッキーCi と共にクライアントへ送信する．
　Cookie_RES = Ci|Cr|Nr

(5) ユーザ認証情報の暗号化
　クライアントではログイン画面が表示され，ユーザが認証情報を入力する．その後，ユーザ認証情報(PW，T)を PuI で暗号化する．また，Diffie-Hellman 交換値 DH1 を生成する．
　ユーザ認証情報⇒$E_{PuI}$[PW|T]|DH1

(6) CertUser_DIST の送信
　(5)で作成したユーザ認証情報と IC カードから受け取った乱数 Ni，サーバから受け取った乱数 Nr を IC カードへ送信する．
　CertUser_DIST = $E_{PuI}$[PW|T]|Ni|Nr|DH1

(7) ユーザ認証，IC カード認証情報の生成
　IC カードでは IC カード秘密鍵 PrI を利用して PW，T を取り出しユーザ認証を行う．更に，（2）で生成した乱数 Ni と受け取った乱数 Ni を比較する．ユーザ認証後，乱数 Nr に DH1 を付加して，これらの情報に IC カード秘密鍵 PrI でディジタル署名を作成する．
　IC カード認証情報⇒$S_I$(DH1|Nr)

(8) SignIC_DIST の送信
　(7)で作成した IC カード認証情報を uID と共にクライアントへ送信する．
　SignIC_DIST = uID|$S_I$(DH1|Nr).

(9) Info_DIST の送信
　クライアントは(8)で受信した IC カード認証情報を(4)で受信したクッキーCi, Cr と共にサーバへ送信する．
　Info_DIST = uID|$S_I$(DH1|Nr)|Ci|Cr

(10) IC カード認証，サーバ認証情報の生成
　サーバではクライアントが送ってきたクッキーの正当性を確認する．また，uID から該当する IC カード公開鍵 PuI を読み出し，ディジタル署名の検証を行い，IC カードを

認証する．同時に，（4）で生成した乱数 Nr と受け取った乱数 Nr を比較する．その後，Diffie-Hellman 交換値 DH2 を生成する．DH2 に DH1 を付加して，これらの情報にサーバ秘密鍵 PrS を用いてサーバ認証を行うためのディジタル署名を作成する．更に，取得した DH1 と DH2 を利用して共通暗号鍵 K を生成する．

サーバ認証情報⇒$S_S$(DH1|DH2)

(11) SignMS_DIST の送信

(10)で作成した署名情報とクッキーCi, Cr をクライアントへ送信する．

SignMS_DIST = $S_S$(DH1|DH2)|Ci|Cr

(12) サーバ認証

クライアントではサーバが送ってきたクッキーの正当性を確認する．また，あらかじめ受信した PuS を利用しディジタル署名の検証を行い，サーバを認証する．その後 DH1 と DH2 を取得する．DH1 が（6）で送信した値と等しいか検証する．更に，DH1, DH2 を利用して共通暗号鍵 K を生成する．以降のクライアント/サーバ間の暗号通信はこの暗号鍵 K を用いて行う．

### 3.6 セキュリティの考察

3.5 節で施した対策についての考察を述べる．

(1) DoS 攻撃

クライアント/サーバ間の認証処理に先立ち，お互いがクッキー交換することにより対応する[21]．クッキーの値は，送信元 IP アドレスと送信先 IP アドレス，乱数を基に生成される．サーバはクライアントの IP アドレスと生成したクッキーの対応をサーバ認証が終了するまで保持する．クッキーは通信ごとに異なる値となるため，IC カード認証時にクライアントからサーバへのパケットに含むことにより，無関係な端末からの Dos 攻撃を防止することができる．なぜなら，IP アドレスを偽造して攻撃を行う攻撃者の IP アドレスは，事前に作成したクッキーの対応表に該当しないからである．

(2) リプレイ攻撃

IC カードが生成する乱数 Ni とクッキー交換時に送信される乱数 Nr を利用することで対応する．乱数は通信毎に生成されるため，攻撃者が認証で成功したパケットを用いてリプレイを試みたとしても，その乱数を含むパケットはすでに認証済みであるため拒否する．全ての認証処理が終了した際に受信パケットは破棄される．また，過去の認証済みパケットの場合は，送られてくるクッキーのタイムスタンプが古いため防ぐことができる．乱数 Nr は，IC カード認証情報が認証時に作成されたものであるかどうかを確認するためにも利用する．

(3) 中間者攻撃

IC カード/クライアント間は中間者攻撃が出来ないという前提であるため，クライアント/サーバ間の対策について述べる．

中間者攻撃を防ぐにはエンドエンドの認証が必要である．そこで，ディジタル署名を用いて対策する．クライアントは秘密鍵を持たないため，サーバに対して完全性を保証したいデータである DH を IC カードへ送信している．IC カードはクライアントの代わりとなって，IC カードの秘密鍵で DH にディジタル署名を行いクライアント経由でサーバへ送信する．サーバは，受信したディジタル署名を検証することでクライアントを間接的に認証し，DH を得ることができている．サーバからクライアントへ送信する DH も同様にディジタル署名を行うことでデータの完全性と作成者の認証を行っている．さらに，クライアントは送信した DH1 と受信した DH1 を比較することで正しく交換が行えていることを確認することでより強固な対策としている．

## 4. SPAIC の実装

### 4.1 USB トークンによる試作

SPAIC では非接触型 IC カードを利用することを前提としているが，公開鍵演算処理が可能な非接触型 IC カードの入手は現段階では困難である．そこで，試作システムの実装には，IC カードの代わりに USB トークンである PUPPY[23]を利用することとした．USB トークンにはプロセッサとメモリが搭載されており，内部で演算することができる．また，RSA キーペア生成，ディジタル署名生成と認証機能を持つため，試作システムの実装に利用可能である．ユーザ認証には，USB トークンの内部に保持する秘密鍵を用いて，公開鍵により暗号化されたパスワードを復号し，クライアント上で照合することができる．USB トークンの認証には，サーバ側で USB トークンの秘密鍵により作成されたディジタル署名を，対応する公開鍵を用いて検証する．

### 4.2 モジュール構成

各端末における試作システムのモジュール構成を図 5 に示す．また，試作システムでは，IC カード内で実現すべきプログラムの一部がパソコン（PC）上での開発となるため，該当箇所を図 5 に網掛けで示している．詳細については後述する．

各端末には共通するモジュールと固有のモジュールで構成される．共通モジュールには，メインモジュールと初期処理，暗号化処理モジュールがある．メインモジュールは一連の処理状態を管理して，その状態に対応した処理を行うサブモジュールを呼び出す．暗号化

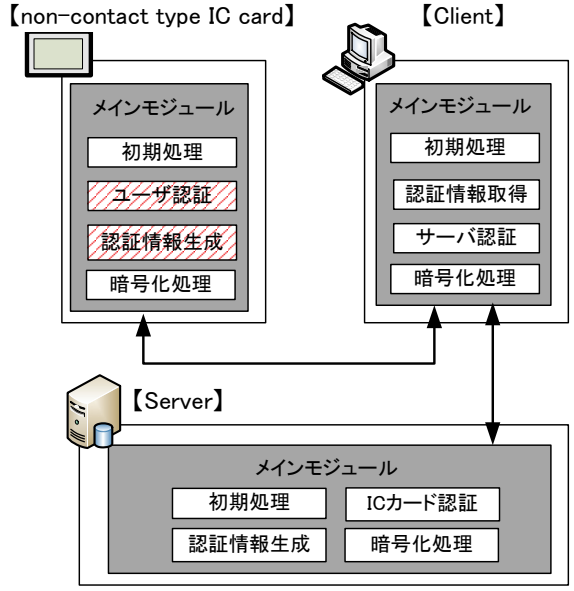


図 5 モジュール構成

処理モジュールは通信パケットの暗号化/復号化や，ディジタル署名の検証などを行う．各端末の固有のモジュールは以下の処理を行う．

クライアント固有のモジュールは認証情報取得とサーバ認証がある．認証情報取得モジュールはパスワードの取得を行う．サーバ認証モジュールはサーバ署名情報を検証して認証を行う．

サーバ固有のモジュールは IC カード認証と認証情報生成がある．IC カード認証モジュールは IC カードを認証するための処理を行う．認証情報生成モジュールはサーバ認証に必要な情報を生成する．

IC カード固有のモジュールはユーザ認証と認証情報生成がある．ユーザ認証モジュールはクライアントから受信したパスワードを照合することよりユーザ認証処理を行う．認証情報生成モジュールは IC カード認証に必要な情報を生成する．また，ユーザ認証に含まれるパスワードの検証とパスワード検証，認証情報に含まれる署名のメッセージダイジェストの生成などは PC 上での処理となる．

### 4.3 実験の概要

SPAIC における処理の中では，公開鍵演算にかかる時間が大部分を占めるため，IC カードが行う合計処理時間，クライアントが行う合計処理時間，サーバが行う合計処理時間を求め，そのうち公開鍵演算に掛かる処理を別途測定した．また，公開鍵暗号アルゴリズムは RSA（PKCS モード）とし，RSA の鍵長は 1024bit，Diffie-Hellman の鍵長は 768bit としている．

実験に用いた PC と USB トークンの装置仕様を表 2 に示し，実験装置のネットワーク構成を図 6 に示す．

表 2. 装置仕様

| 装置名 | 項目 | スペック |
|---|---|---|
| PC1（ICcard） | CPU | Pentium M (1.7GHz) |
| | Memory | 504MB |
| | OS | XP Professional SP3 |
| PC2（Client） | CPU | Core 2 Duo U7600(1.2GHz) |
| | Memory | 2GB |
| | OS | 7 Ultimate |
| PC3（Server） | CPU | Core2 Duo E6600(2.4GHz) |
| | Memory | 4GB |
| | OS | Vista Business SP2 |
| USB トークン | Model CPU | Sony FIU-810-N03 ARM7 |
| | Interface | USB |
| | Power | From USB |

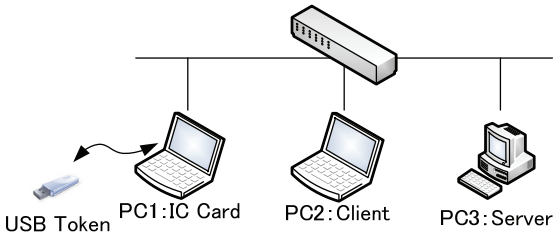


図 6 ネットワーク構成

実験には，ラップトップ PC を 2 台とデスクトップ PC を 1 台の合計 3 台の PC を用いて行った．各 PC はスイッチングハブで接続せれている．ラップトップ PC1 には，USB トークンを接続しておき，PC1 と USB トークンを 1 組として IC カードの処理プログラムを実行させた．ラップトップ PC2 には，クライアントの処理プログラムを，デスクトップ PC3 にはサーバの処理プログラムを実行させた．また，各端末における送信時間と受信時間の計測にはクライアント上でパケットキャプチャソフトである Wireshark[23]を用いて行った．受信時間と送信時間の差を求めることにより，シーケンス処理時間を求めている．

## 5. 評価

### 5.1 性能評価

各端末におけるシーケンス毎の測定結果を図 7 に，その内の公開鍵演算処理に掛かる時間を表 4 に示す．表 4 中の番号は図 7 において，どのタイミングで処理項目が行われているかを示している．図 7 と表 4 より，IC カードの復号処理時間は 308.8ms，署名の処理時

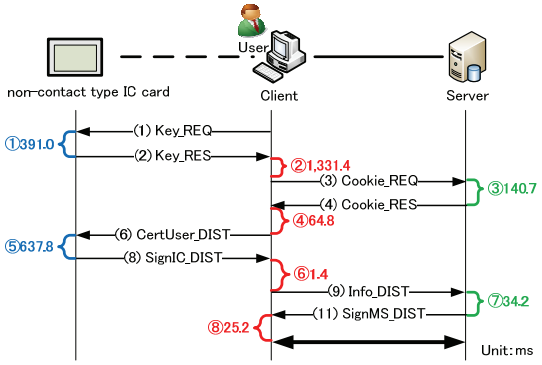


図 7 各端末におけるシーケンス処理時間

表 4 公開鍵暗号の処理時間

| Terminal | Number | Item | Times |
|---|---|---|---|
| USB Token | ⑤ | Decrypt | 308.8ms |
| | | Sign | 317.7ms |
| Client | ② | Encrypt | 1,263.5μs |
| | | DH | 7,456.0μs |
| | ⑧ | Verify | 673.2μs |
| Server | ⑦ | Sign | 13.0ms |
| | | DH | 5,418.0μs |
| | | Verify | 427.7μs |

間は 317.7ms となり，全ての処理に掛かる時間は 1.029s であった．IC カードの合計処理時間にはクライアント上で代わりに行っている処理が 12ms 程度あるが，特別な処理ではないため，IC カードで処理を行っても大きく違うことはないと考える．クライアントでは，暗号化の処理時間が 1,263.5μs，署名の確認処理が 673.2μs，DH の生成時間が 7,456.0μs となり，合計処理時間が 1,436s であった．サーバでは署名の処理時間が 13.0ms，署名の確認処理が 427.7μs，DH の生成処理時間が 5,418.0μs となり，全ての処理に掛かる時間は 0.175s であった．従って，SPAIC の認証動作処理が終了するまでに要する時間は 2,64s であった．

しかし，IC カード処理①とクライアント処理②で予想よりも処理時間が長くなっている．IC カード処理①の原因は，USB トークンからの鍵の読み出しに 347.4ms 要しているからである．クライアント処理②の原因は，OPENSSL[24]の関数を利用する際に，DLL（Dynamic Link Library）の読み出しにおよそ 1.2s 要しているからである．クライアント処理②を改善することで，SPAIC の認証動作は 1.5s 程度となり，立ち上げ時に掛かる処理としては，許容範囲になると考える．

### 5.2 PSK 方式との比較

事前共有鍵方式と SPAIC の比較を表 5 に示す．SPAIC ではクライアント端末に格納する情報が動作プログラムのみであるため，クライアントからの情報漏洩の心配がない．事前

表 5. 事前共有鍵方式と SPAIC の比較

| | 事前共有鍵方式 | SPAIC |
|---|---|---|
| クライアントに格納する情報 | 動作プログラム，事前共有鍵 (×) | 動作プログラムのみ (○) |
| 管理負荷 | 共有鍵の変更が必要 (×) | ユーザの追加・削除程度 (○) |
| IC カード/クライアント間の暗号化 | 事前共有鍵を利用 (○) | 公開鍵方式を利用 (○) |
| IC カードへの負荷 | 中程度 (○) | 高い (△) |

共有鍵方式では，システムの安全上共有鍵を頻繁に更新する必要があるため，運用時の管理が煩雑になる．一方 SPAIC ではユーザの追加，削除程度の作業で済むため，管理負荷の低減が見込まれる．SPAIC では，IC カードで公開鍵演算を行うため，IC カードへの処理負荷は事前共有鍵方式より高い．しかし，SPAIC が動作するのはクライアントの立ち上げ時のみであるため，5．1 節の結果から許容範囲と考えられる．

## 6. まとめ

本論文では，事前共有鍵方式においてクライアント端末からの情報漏洩の問題を解決するために，クライアント端末が動作プログラム以外の初期情報を一切所持しないというモデルを定義し，非接触型 IC カードを用いてサーバからクライアントに重要情報を配送することを可能とするプロトコル SPAIC の提案を行った．IC カード公開鍵を新たに IC カードに所持させることにより，クライアントが初期情報を持たなくとも IC カード/クライアント間の暗号通信を行い，IC カード/クライアント/サーバ間での確実な認証を可能にした．更に，クライアント/サーバ間で Diffie-Hellman 鍵交換で作成した暗号鍵を利用することにより，安全に重要情報を配送するための通信経路を確立した．

性能評価においては，IC カードの代替として USB トークンを用いて，SPAIC の認証動作に掛かる時間を推測した．さらに，各端末に掛かる処理時間を求め，ボトルネックとなる公開鍵演算処理がどの程度の割合を占めるかを推測した．本方式では，システム立ち上げ時の認証において十分に利用できると考えられる．

## 参考文献


[1] J. Kohl, Digital Equipment Corporation, C. Neuman, ISI., “The Kerberos Network Authentication Service (V5),” RFC 1510, IETF, Sep. 1993.

[2] S. Kent and R. Atkinson, “IP Encapsulating Security Payload (ESP),” RFC 2406, IETF, Nov. 1998.

[3] D. Harkins and D. Carrel, “The Internet Key Exchange (IKE),” RFC 2409, IETF, Nov. 1998.

[4] T. Dierks, Certicom, C. Allen, and Certicom, “The TLS Protocol Version 1.0,” RFC 2246, Jan. 1999.

[5] Richard E. Smith (著)，稲村雄 (訳), “認証技術－パスワードから公開鍵までー”, オーム社，2003.

[6] 渡邊晃，厚井裕司，井手口哲夫，横山幸夫，妹尾尚一郎，“暗号技術を用いたセキュア通信グループの構築方式とその実現”，情報処理学会論文誌，Vol.38，No.4，pp.904-914，Apr. 1997.

[7] 渡邊晃，岡崎直宣，朴美娘，井手口哲夫，笹瀬巌，“イントラネット閉域通信グループの構築に適した安全な鍵配送方式とその運用管理方式”，電気学会論文誌 C，Vol.121-C ， No.9 ， pp.1429-1438 ，Sep.2001.

[8] 妹尾尚一郎，厚井裕司，貞包哲男，中谷直司，馬場義昌，鹿間敏弘，“生体認証によるネットワーク個人認証システム”，情報処理学会論文誌，Vol.44，No.4，pp.1111-1120，Apr. 2003.

[9] 瀬戸洋一，“ユビキタス時代のバイオメトリクスセキュリティ”，日本工業出版，2003.

[10] 今本健二，櫻井幸一，“認証付き鍵交換における動的情報管理に関する考察”，電子情報通信学会技術研究報告，Vol.102，No.741，pp.91-96，Mar. 2003.

[11] 磯部義明，三村昌弘，瀬戸洋一，菊池良知，“本人認証 IC カードによる高セキュリティシステムの構築”，情報処理学会コンピュータセキュリティ研究報告，99-CSEC-4，Vol.99，No.24，pp.55-60，Mar. 1999.

[12] 吉田壱，平田真一，“IC カード技術の現状と課題”，情報処理学会会誌，Vol.43，No.3，pp.296-303，Mar. 2002.

[13] 影井良貴，“IC カードの動向”，情報処理学会会誌，Vol.39，No.5，pp.429-433，May. 1998.

[14] 伊藤雅彦，“非接触 IC カード技術とその応用”，情報処理学会会誌，Vol.43，No.3，pp.304-307，Mar. 2002.

[15] 佐藤良夫，“IC カードによる個人認証”，Unisys Technology Review ， No.73 ，pp.131-139，May 2002.

[16] IC カードシステム利用促進協議会，“JICSAP IC カード仕様書 V2.0”，July 2001.

[17] Diffie, W. and Hellman, M.: New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp.644-654, 1976.

[18] E. Rescorla, RTFM Inc., “Diffie-Hellman Key Agreement Method”, RFC2631, IETF, June. 1999.

[19] J. Schiller, Massachusetts Institute of Technology, “Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)”，RFC4307, IETF, Dec. 2005.

[20] W. Polk, R. Housley, and L. Bassham, “Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 3279, IETF, Apr. 2002.

[21] D. Maughan，National Security Agency，M. Schertler，Securify, Inc.，M. Schneider，National Security Agency ，J. Turner ，RABA Technologies, Inc. ，“Internet Security Association and Key Management Protocol (ISAKMP)” ，RFC2408，IETF，Nov. 1998.

[22] Sony Japan | 指紋認証トークン PUPPY http://www.sony.co.jp/Products/Media/puppy/

[23] Wireshark Go deep. http://www.wireshark.org/

[24] OpenSSL Project. The Open Source toolkit for SSL/TLS. http://www.openssl.org/.