目　次

概要

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．

第 1 章　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，確実な認証と暗号化が必須である．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，異なるクライアントからサーバにアクセスしたいというニーズが増えており, 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 チップに CPU やメモリを搭載し，カード内部で演算することができる．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, クライアント端末内にユーザの情報を保持することなく認証と暗号通信を行うことが可能である[13]．これは，ユーザが端末を選べるという利便性だけでなく，端末からユーザの情報が盗まれるのを防止するという利点もある．

近年では非接触型 IC カードの発展に大きな期待が寄せられている．非接触型 IC カードでは IC カードを IC カードリーダに近づけるだけでデータの交換が行えるため, 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 カード内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安全にやりとりするためには，IC カードとクライアント間の暗号通信が必須である.

IC カードと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情報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が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, IC カード, クライアント, サーバがあらかじめどのような初期情報を所持し, どのような手順で認証や暗号化を行うべきかについて検討を行った.

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．

第 2 章　既存技術とその課題

従来システムでは，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るため，IC カードとクライアントが一体のものであるとみなし，IC カード/クライアント間の暗号通信を行っていないものが殆どである．

しかし，非接触 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（図 1)．この方式では，事前共有鍵を用いて IC カード/クライアント間で暗号通信を行うための暗号鍵をダイナミックに生成する．

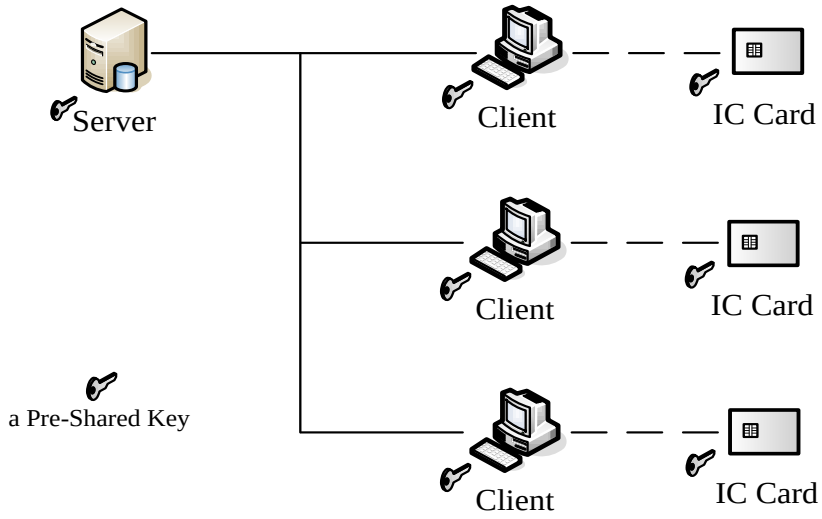


図 1　従来システムの認証方式

Figure 1　Conventional Authentication Method

しかし, 事前共有鍵方式では, クライアントに秘密情報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であり，管理が煩雑である．

第 3 章　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）というプロトコルを提案する．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．

第 3.1 節　システムモデルと前提条件

本研究で想定するシステムモデルを図 2 に示す．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．

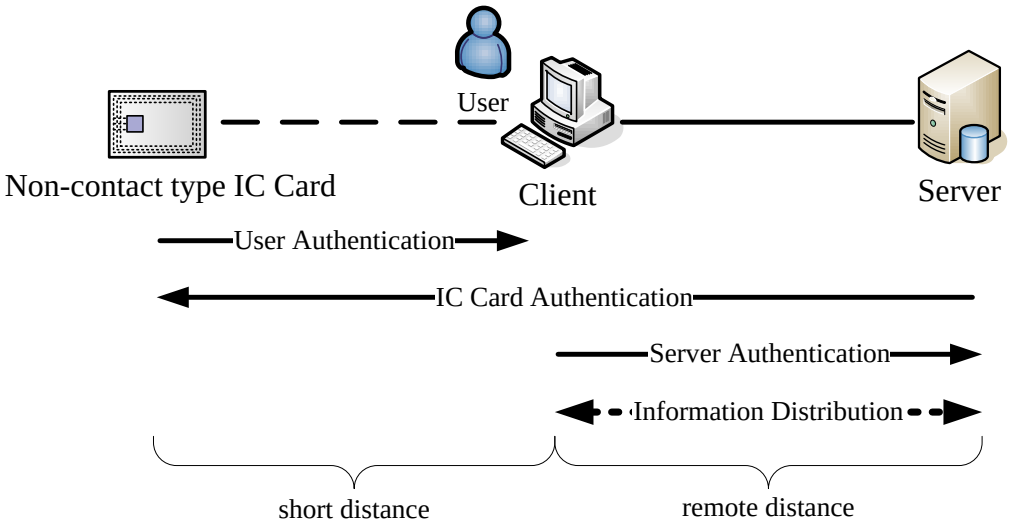


図 2　想定するシステムモデル

Figure 2　System Model Assumed in Our Proposal

想定システムでは，以下の条件を前提とする．

(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．

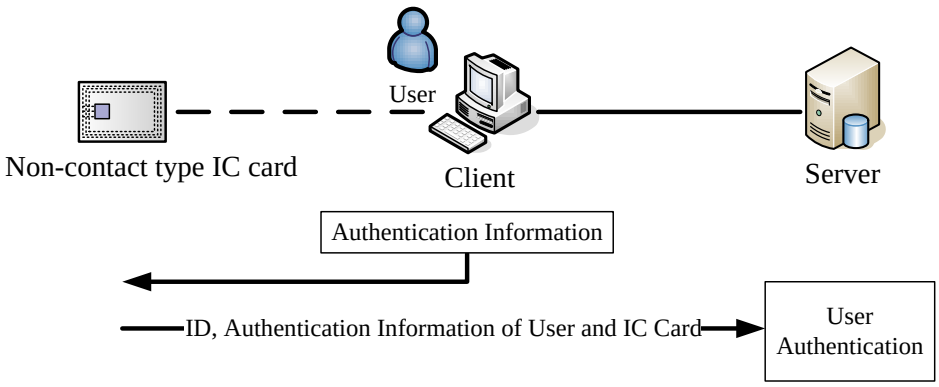
IC カードの正当性は「IC カード認証」で確認されるが，IC カードの正当な持ち主を確認するためには「ユーザ認証」が必要となる．

ユーザ認証の方法として，一般的にはパスワードが用いられる．より高い安全性を必要とする場合には，生体認証などと組み合わせる．このとき，ユーザ認証情報の格納場所の違いにより，サーバに情報を格納して認証を行うサーバ型認証と IC カード内に情報を格納して認証を行うクライアント型認証に分けられる（図 3)．

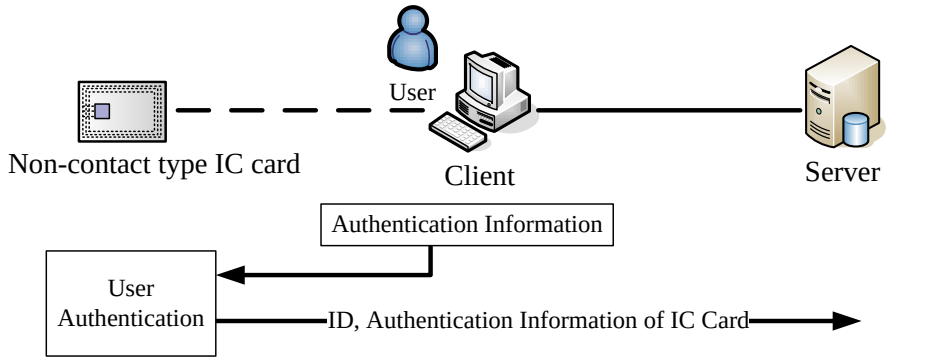
サーバ型認証は，ユーザとサーバ間で直接認証を行うエンドエンドのユーザ認証である．クライアントで取得した認証情報を，IC カードを経由してサーバへ送信して認証を行う．この認証方式ではサーバ側でユーザ認証と IC カード認証を一括して行うため，IC カードの処理負荷を軽減できるというメリットがある．しかし，ユーザ全員の情報をサーバ側で一括して管理するため，サーバの管理体制が重要となる．このため，大規模な耐タンパハードウェアを用いたり，厳重な設備を準備するといった対策が必要となる．

クライアント型認証は，ユーザ/IC カード，IC カード/サーバ間でそれぞれ認証を行うリンクバイリンク認証である．クライアントで取得した認証情報を IC カードへ送信して IC カード内でユーザ認証を行い，その後 IC カード/サーバ間で IC カード認証を行う．この認証方式ではサーバにおける IC カード認証がユーザ認証を兼ねることになる．IC カードは耐タンパ性を有しているため，パスワードや生体情報などのユーザ認証情報を安全に格納することができるというメリットがある．しかし，IC カードに掛かる処理負荷が大きくなる．

どちらの認証方式においても，安全に個人認証を行うことが可能である．本論文ではユーザ認証と IC カード認証を独立して扱うことができ，簡単に安全性が達成できるクライアント型認証を採用する．



(a) SERVER-type Authentication



(b) CLIENT-type Authentication

図 3　ユーザ認証方式

Figure 3　User Authentication Methods

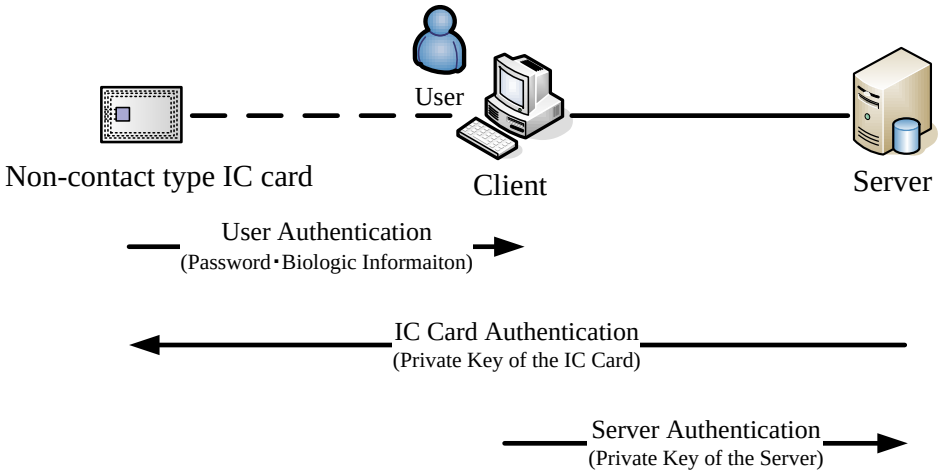


図 4　認証の関係

Figure 4　Relationships of Authentications

SPAIC では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．

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．SPAIC で行う認証の関係を図 4 に示す．矢印は認証の方向を示している．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．IC カードはパスワードや生体情報を用いてユーザ認証を行うことによりクライアントを認証する．サーバは IC カード秘密鍵から作成されたディジタル署名を検証することにより IC カードを認証する[20]．クライアントはサーバ秘密鍵から作成されたディジタル署名を検証することによりサーバを認証する．

以上の 3 つの経路の認証を実現することにより，クライアント/サーバ間の認証が行われる．

第 3.2 節　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

第 3.3 節　各端末の情報

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．

SPAIC の場合は，IC カードには，事前共有鍵方式における共有鍵 PSK に代わり，IC カード公開鍵 PuI を格納する，その他の初期情報は同様である．クライアントは初期情報を一切所持しない．また，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 を所持する．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．

表 1　事前共有鍵方式と SPAIC の初期情報

Table 1　Comparison of the Initial Information Possessed by PSK Method and SPAIC

	PSK Method	SPAIC
IC card	uID PrI PuS PW T ***PSK***	uID PrI PuS PW T ***PuI***
Client	***PSK***	−
Server	uID PrS PuI	uID PrS PuI

第 3.4 節　SPAIC の動作

SPAIC の動作概要を図 5 に示す．SPAIC の認証動作は三段階ある．まず，IC カードは以下の手順によりユーザ認証を行う．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，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，サーバは以下の手順により IC カードを認証する．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．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，以下の手順によりクライアントはサーバを認証する．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 つの経路の認証により，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．

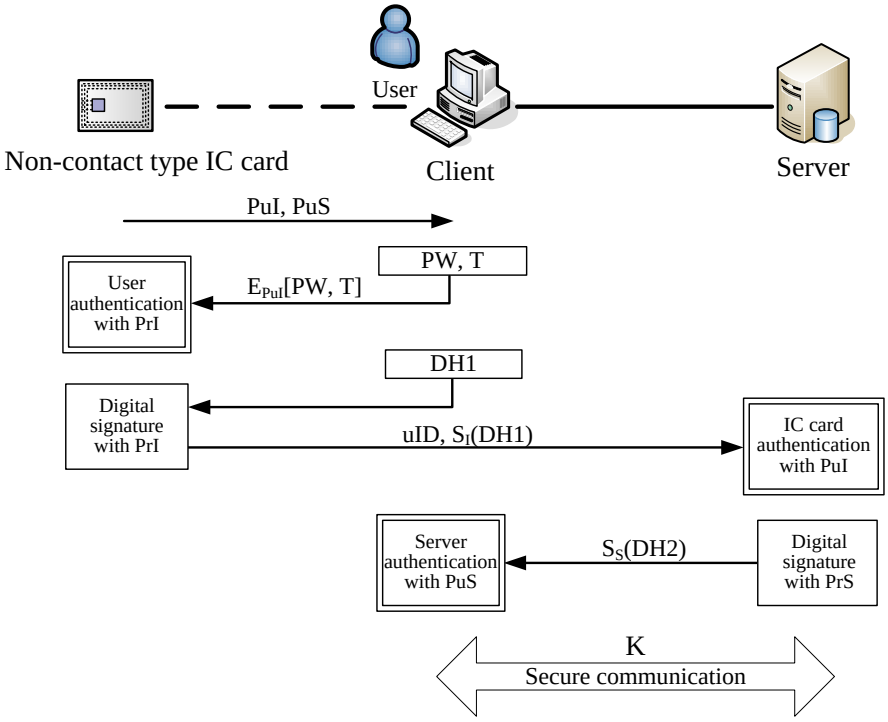


図 5　SPAIC の動作概要

Figure 5　Outline of SPAIC Operation

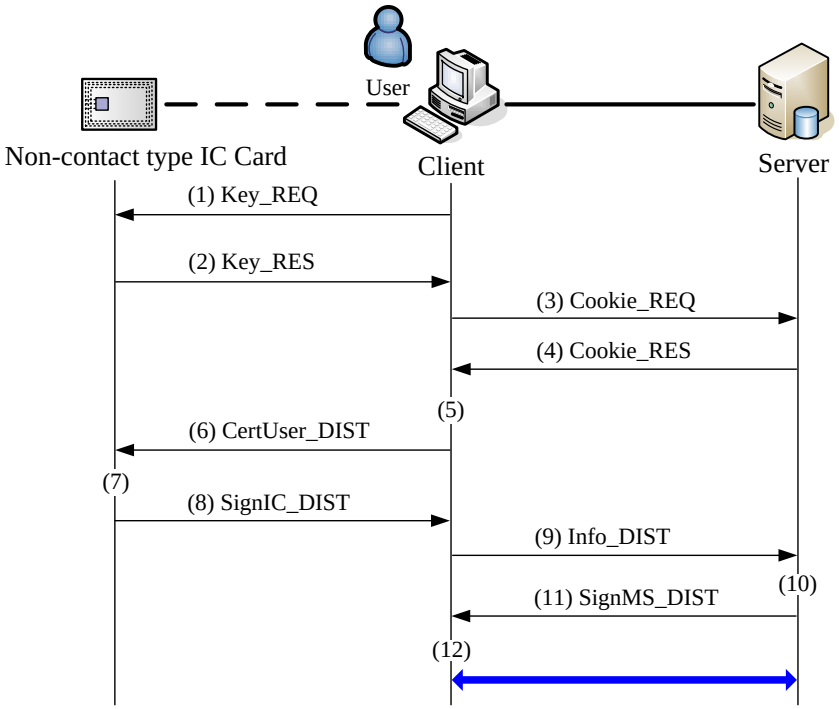


図 6　SPAIC の詳細シーケンス

Figure 6　Detailed Sequence of SPAIC

実際の認証システムにおいては，サービス不能攻撃（DoS 攻撃）やリプレイ攻撃への対応が重要となる．

DoS 攻撃に対しては，クライアント/サーバ間で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，通信相手の ID，IP アドレスと乱数をもとに生成される．クッキーは通信ごとに異なる値が生成されるため，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．

リプレイ攻撃に対しては, 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 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．

SPAIC の詳細シーケンスを図 6 に示す．クライアントの電源を ON して，プログラムを起動しておく．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， 認証処理を開始する．

(1) ICカードへKey_REQを送信
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を送信
ICカードはユーザID(uID)，ICカード公開鍵PuI，サーバ公開鍵PuS，乱数Niを送信する．
uID, PuI, PuS, Ni

(3) Cookie_REQを送信
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．
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．
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, ICカードから受け取った乱数Niとユーザ認証情報(PW, T)をPuIで暗号化する．同時にサーバから受け取った乱数NrをPuSで暗号化する．また，Diffie-Hellman交換値DH1を生成する．
$E_{PuI}[PW, T, Ni], E_{PuS}[Nr], DH1$

(6) CertUser_DISTの送信
(5)で作成した認証情報をICカードへ送信する．
$E_{PuI}[PW, T, Ni], E_{PuS}[Nr], DH1$

(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，Niを取り出しユーザ認証を行う．更に，生成したNiと比較する．ユーザ認証後，E_{PuS}[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．

S_I(DH1, E_{PuS}[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．

uID, S_I(DH1, E_{PuS}[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．

uID, S_I(DH1, E_{PuS}[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，サーバ秘密鍵PrSを利用してNrを取り出し，生成したNrと比較する．その後，Diffie-Hellman交換値DH2を生成し，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．

S_S(DH2), K

(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．

S_S(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DH2を取得する．更に，DH1，DH2を利用して共通暗号鍵Kを生成する．

K

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．

第 4 章　SPAIC の実装

第 4.1 節　機能分担

クライアントおよびサーバにおける試作システムの実装概要を図 7 に，IC カード，クライアントおよびサーバのモジュールと主な機能を表 2 から表 4 に示す．

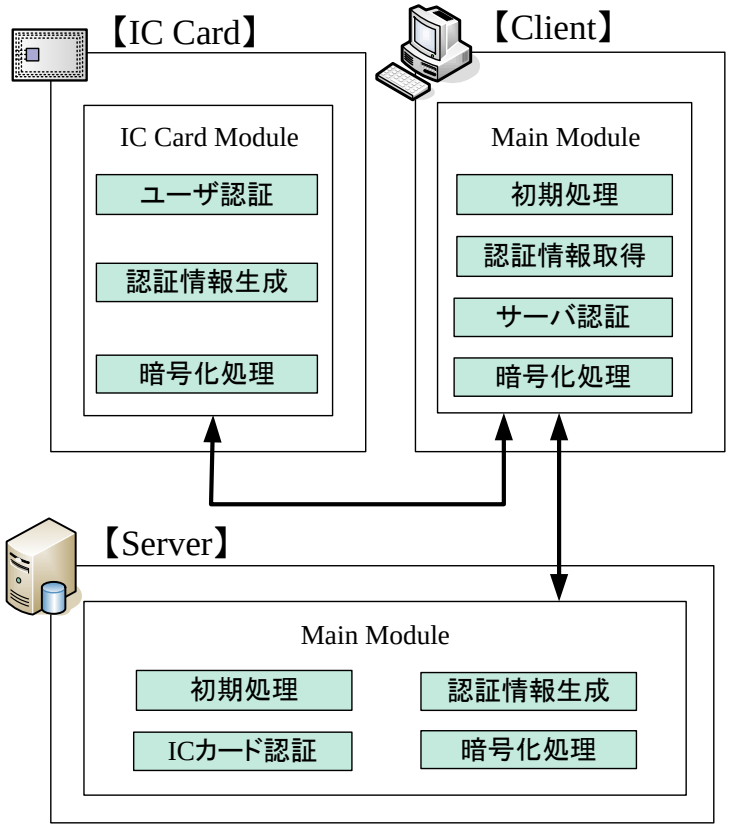


図 7　実装概要図

Figure 7　Outline Figure of Implementation

表 2　クライアント側のモジュール構成と機能

Table 2　Module and Function of the Client

モジュール	機能
メインモジュール	主処理および各サブモジュールの呼び出し
初期処理	初期化処理，IC カードとのコネクション確立
認証情報取得	認証情報となるパスワードの取得
サーバ認証	サーバ署名情報の検証
暗号化処理	暗号化/復号化処理，ディジタル署名の検証

表 3　IC カード側のモジュール構成と機能

Table 3　Module and Function of the IC Card

モジュール	機能
ユーザ認証	パスワードの認証処理
認証情報生成	IC カード認証情報の生成
暗号化処理	暗号化/復号化処理，ディジタル署名の生成

表 4　サーバ側のモジュール構成と機能

Table 4　Module and Function of the Server

モジュール	機能
メインモジュール	主処理および各サブモジュールの呼び出し
初期処理	初期化処理
IC カード認証	IC カード署名情報の検証
認証情報生成	サーバ認証情報の生成
暗号化処理	暗号化/復号化処理，ディジタル署名の生成/検証

クライアントの処理は，メインモジュールと，初期処理，認証情報取得，サーバ認証，および暗号化処理サブモジュールにより構成される．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．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，暗号化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．

IC カードの処理は，ユーザ認証，認証情報生成，および暗号化処理モジュールにより構成される．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．暗号化処理モジュールは通信パケットの暗号化/復号化や，ディジタル署名の生成などを行う．

サーバの処理は，メインモジュールと，初期処理，IC カード認証，認証情報生成，および暗号化処理サブモジュールにより構成される．メインモジュールはクライアントと同様に，一連の処理状態の管理，および各サブモジュールの呼び出しを行う．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．暗号化処理モジュールは通信パケットの暗号化/復号化や，ディジタル署名の生成/検証などを行う．

第 4.2 節　状態遷移図

SPAIC における IC カード，クライアントおよびサーバの状態遷移を図 8 に示す.

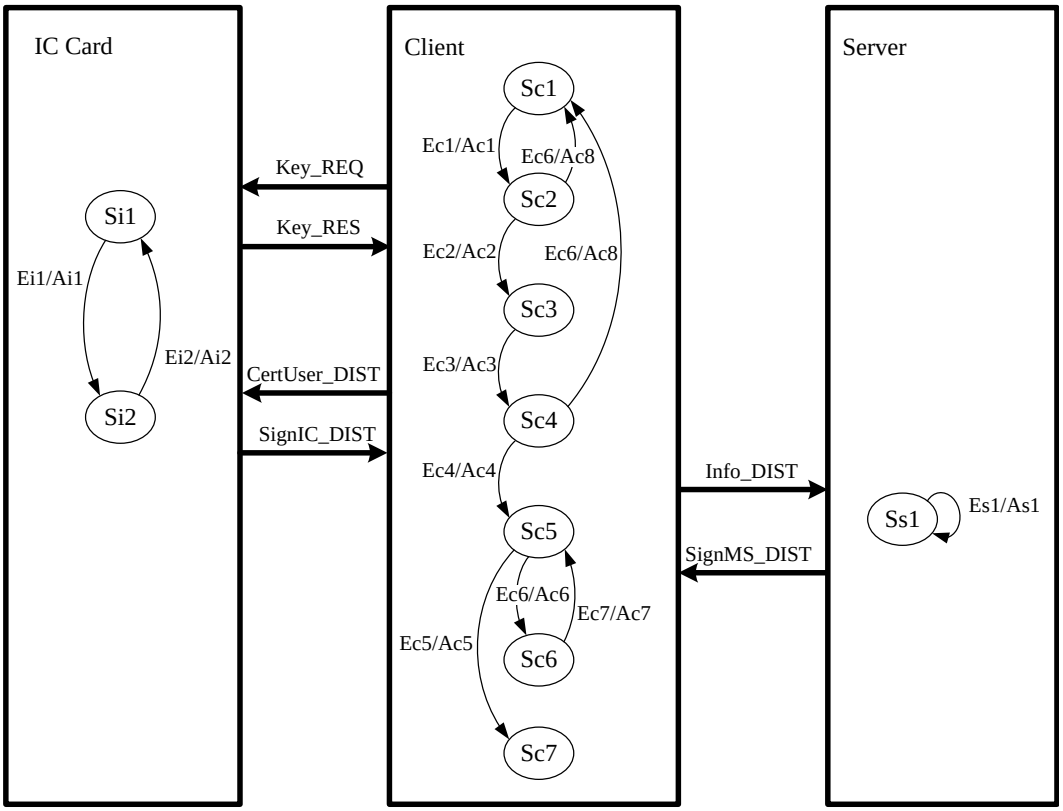


図 8　SPAIC の状態遷移図

Figure 8　State Transition Diagram of SPAIC

状態遷移図におけるそれぞれの記号の意味を表 5 から表 7 に説明する.

表 5　状態定義

Table 5　State Definition

端末	状態記号	意味
Client	Sc1	ログイン指示待ち
	Sc2	Key_RES 待ち
	Sc3	PW 入力待ち
	Sc4	SignIC_DIST 待ち
	Sc5	SignMS_DIST 待ち
	Sc6	Info_DIST 再送指示待ち
	Sc7	一般通信可能状態
IC Card	Si1	Key_REQ 待ち
	Si2	CertUser_DIST 待ち
Server	Ss1	Info_DIST 受信待ち

表 6　イベント定義

Table 6　Event Definition

端末	イベント記号	意味
Client	Ec1	ログイン指示
	Ec2	Key_RES 受信
	Ec3	PW 入力
	Ec4	SignIC_DIST 受信
	Ec5	SignMS_DIST 受信
	Ec6	TimeOut
	Ec7	Info_DIST 再送指示
IC Card	Ei1	Key_REQ 受信
	Ei2	CertUser_DIST 受信
Server	Es1	Info_DIST 受信

表 7　アクション定義

Table 7　Action Definition

端末	アクション記号	意味
Client	Ac1	Key_REQ 送信，タイマ起動
	Ac2	ログイン画面表示
	Ac3	DH1 生成，E_{PuI}[PW]生成， CertUser_DIST 送信，タイマ起動
	Ac4	Info_DIST 送信，タイマ起動
	Ac5	サーバ認証，K 生成
	Ac6	エラー表示＆再送指示画面表示
	Ac7	Info_DIST 送信，タイマ起動
	Ac8	エラー表示＆ログイン指示画面表示
IC Card	Ai1	Key_RES 送信
	Ai2	PrI で E_{PuI}[PW]を復号， CertUser_DIST 生成， CertUser_DIST 送信
Server	As1	PuI で検証，IC カード認証， DH2 生成，SignMS_DIST 生成， SignMS_DIST 送信，K 生成

クライアントは電源投入後，SPAIC 用プログラムを起動する．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， 認証処理を開始する．

(1) クライアントにおける状態遷移

クライアントでは，プログラムが起動されたら，ログイン指示画面を表示し，状態 Sc1 に遷移する．ログイン指示を受けたら，Key_REQ を送信，タイマを起動し，Sc2 に遷移する．Key_REQ を受信したら，ログイン画面を表示し，状態 Sc3 に遷移する．ユーザがパスワードを入力したら，DH1 生成，E_{PuI}[PW]生成，CertUser_DIST 送信，タイマ起動などの一連の処理を行い，状態 Sc4 に遷移する．SignIC_DIST を受信したら，Info_DIST を送信，タイマを起動し，状態 Sc5 に遷移する．SignMS_DIST を受信したら，サーバ認証および K 生成を行い，状態 Sc7 に遷移する．

また，状態 Sc2 と Sc4 に対して，タイムアウトが発生した場合，エラー表示した後，ログイン指示画面に移行して，状態 Sc1 に戻る．状態 Sc5 に対して，タイムアウトが発生したら，エラーおよび Info_DIST 再送指示を表示し，状態 Sc6 に遷移する．

Info_DIST の再送指示はユーザ指示に従うこととした. 再送を繰り返してもタイムアウトを繰り返す場合は，サーバがダウンしていると考えられる．この場合，プログラムの終了はユーザに任せる．

(2) IC カードにおける状態遷移

IC カードには必要な情報が既に書き込まれ，クライアントからの Key_REQ 待ち状態 Si1 であることを前提とする．IC カードは受信待ちだけなので，タイマ処理は必要ない．Key_REQ を受信したら，Key_RES を送信し状態 Si2 に遷移する．CertUser_DIST を受信したら，PrI で E_{PuI}[PW]を復号，CertUser_DIST 生成および送信など，一連の処理を行う．状態 Si2 でも Key_REQ をする受信可能性はありうる（クライアントからの再送時など)．

(3) サーバにおける状態遷移

サーバには必要な情報登録を完了すると，クライアントからの Info_DIST 待ちの状態 Ss1 で待機する．サーバは Info_DIST を受信すると，IC カードの認証を行う．その後，クライアントと同様，一連の処理の中で K を生成し，SignMS_DIST を送信する．サーバは上記処理を繰り返すだけなので，常に同じ状態 Ss1 であり，状態遷移は発生しない.

第 4.3 節　USB トークンによる試作

SPAIC では非接触 IC カードを利用することを前提としているが， IC カードにプログラムを組み込むのは現時点では困難である．そこで，試作システムの実装には，IC カードの代わりに USB トークンを利用することとした．

USB トークンは，IC カード同様にスマートカードリーダとして Windows システム（デバイスマネージャ）上で認識される．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，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．

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 PC 上での開発となる．将来的には，この部分をそのまま IC カードへ移植することが可能である．

第 5 章　評価

第 5.1 節　性能評価

IC カードで公開鍵演算を行うため, IC カードの処理コストが懸念される. そのため, 試作システムにおいて, USB トークンの公開鍵演算時間を測定した. 実験に用いた PC と USB トークンの装置仕様を表 8 に示す.

表 8　装置仕様

Table 8　Specifications of the Devices

	項目	内容
PC	CPU	Core2 Quad 2.40GHz
	Memory	2GB
	OS	Vista Ultimate
	Interface	USB
USB Token	Model	Sony FIU-810-N03
	CPU	ARM7
	Interface	USB
	Power	From USB

実験では, 公開鍵暗号アルゴリズムは RSA (PKCS モード)とし, 暗号化するデータは 40 バイトとした.

表 9 は RSA の鍵長を 512 ビット, 1024 ビット, 2048 ビットごとに, USB トークン内部での暗号化/復号を 100 回行った処理時間の平均値を示したものである.

表 9　演算時間の測定結果

Table 9　Measurement Results of Process Time

鍵長	暗号化時間	復号時間
512bit	63.0	62.8
1024bit	285.7	286.9
2048bit	1934.3	1937.6

単位：ms

測定の結果, RSA の鍵長を 1024 ビットとした時の暗号化時間は 285.7 ミリ秒, 復号時間は 286.9 ミリ秒なった. この処理時間は初期立ち上げ時の認証時間としては許容範囲と言える.

2048 ビットの鍵長に対しては，暗号化/復号時間とも約 2 秒程度となった．この時間は待ち時間としてはやや大きいと思われる．公開鍵暗号処理時間として，長くないと思われる．従って，より高いセキュリティを望む場合にも応用できる．

現時点では，RSA 鍵長は 1024 ビットあればセキュリティ上も十分と言われており，処理時間とのトレードオフから，本システムでは 1024 ビットを選択することとする．

第 5.2 節　PSK 方式との比較

事前共有鍵方式と SPAIC の比較を表 10 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．

表 10　事前共有鍵方式と SPAIC の比較

Table 10　Comparison between the PSK Method and SPAIC

	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	共有鍵の変更が面倒 （×）	ユーザの追加，削除程度 （○）
IC カード/クライアント間の暗号	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 （△）

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 節で述べたように許容範囲と考えられる．

第 6 章　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.

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.

謝辞

本研究に関して, 研究の方向や進め方など終始御熱心なご指導と御教示を賜りました, 名城大学理工学部情報工学科　渡邊晃教授に心より厚く御礼申し上げます.

本研究を進めるにあたり, 研究内容に関して終始御熱心なご指導と御教示を賜りました, 名城大学理工学部情報工学科　小川明教授, 山本新教授, 宇佐見庄五助教に心より厚く御礼申し上げます.

最後に, 本研究を行うにあたり, 有益なご助言, 適切なご検討をいただいた, 名城大学理工学部情報工学科渡邊研究室の皆様に心より感謝いたします.

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－”, 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] OpenSSL Project. The Open Source toolkit for SSL/TLS. http://www.openssl.org/.

研究業績

1.国際学会

Changjun Shu, Hidekazu Suzuki and Akira Watanabe, “Proposal of an Authentication Method ‘SPAIC’ using a Non-contact Type IC Card”, IEEE 7th International Symposium on Communications and Information Technologies (ISCIT2007), Oct.2007.

2.口頭発表

[1] 束長俊, 鈴木秀和, 渡邊晃, “非接触型 IC カードを用いた重要情報の配送方式 SPAIC の検討”, 電気関係学会東海支部連合大会論文集, Sep.2006.

[2] 束長俊, 鈴木秀和, 渡邊晃, “非接触型 IC カードを用いた重要情報の配送方式 SPAIC の提案”, 電子情報通信学会 2007 年総合大会講演論文集, Mar.2007.

[3] 束長俊, 鈴木秀和, 渡邊晃, “非接触型 IC カードを用いた認証方式 SPAIC の提案”, マルチメディア, 分散, 協調とモバイルシンポジウム（DICOMO2005）論文集, Vol.2007, No.1, pp.1332-1337, Jun.2007.

付録 A　　SPAIC の利用シーン

SPAIC（Secure Protocol for Authentication with IC Card）は，非接触型 IC カードの利用を前提として，サーバからクライアントに重要な情報を安全に配送するためのプロトコルである．

GSCIP(Grouped Secure Communication of Internet Protocol：ジースキップ)において，安全にグループ鍵を配送するためにも利用することができる．

SPAIC を GSCIP に適用する場合のシステム構成を図 9 に示す．移動を考慮したユーザまたは重要なサーバにソフトウェアをインストールした GES（GE realized by Software），ルータに機能を実装した GEN（GE for Network），一般の端末 Term，およびグループ鍵を管理する GMS（Group Management Server）から構成される．GEN はサブネットを構成し，配下の一般端末 Term を保護する．GSCIP では同一の共通鍵を所持する GE の集合を同一の通信グループとして定義する．この共通鍵をグループ鍵 GK と呼ぶ．

SPAIC は，GE の起動時に鍵管理装置 GMS（Group Management Server)と通信を行うことによって，同一グループの端末と暗号化通信を行うためのグループ鍵 GK の取得を行う．

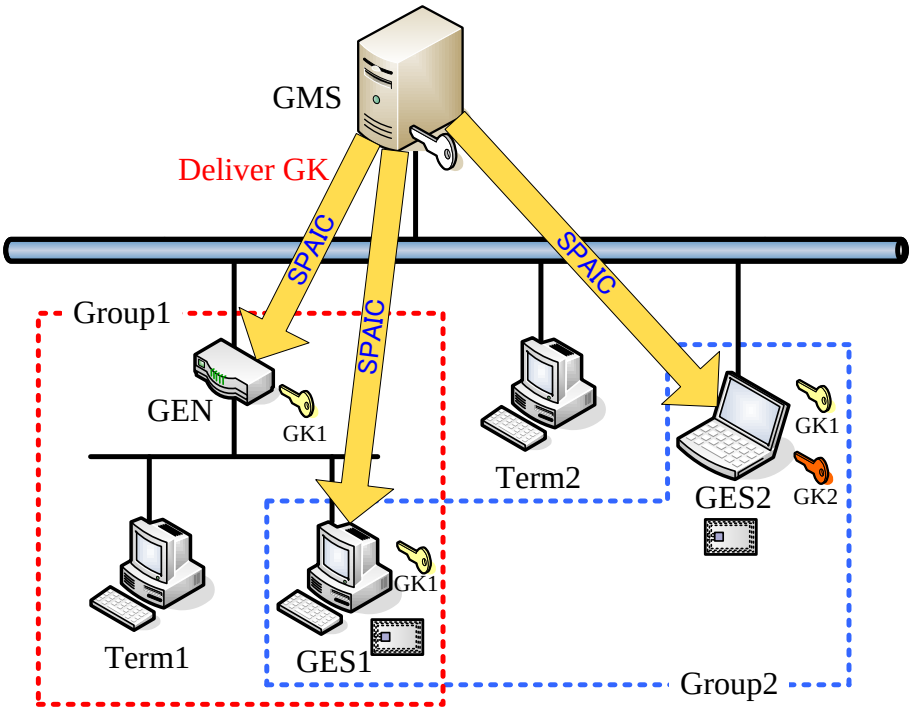


図 9　システム構成

Figure 9　System Model

付録 B　　SPAIC 仕様書

SPAIC を実装するにあたり，仕様書を作成したので付録として添付する．

SPAIC 仕様書

著者

束　長俊　（ソク　チョウシュン）　†

監修

渡邊　晃　（ワタナベ　アキラ）　‡

† 名城大学大学院理工学研究科情報科学専攻

‡ 名城大学理工学部情報工学科　教授

最終更新日：2008 年 2 月 21 日

1. はじめに

SPAIC（Secure Protocol for Authentication with IC Card）は，非接触型 IC カードの利用を前提として，サーバからクライアントに重要な情報を安全に配送するためのプロトコルである．GSCIP（Grouped Secure Communication of Internet Protocol：ジースキップ）において，安全にグループ鍵を配送するために利用することができる．

GSCIP は，GE（GSCIP Element）の起動時に SPAIC を用いて，鍵管理装置 GMS（Management Server）と通信を行うことによって，グループ番号とグループ鍵の取得を行う．

本ドキュメントの実装では非接触型 IC カードの代わりに USB トークンを利用している．

本仕様書に関連するドキュメントの一覧をここに記す．（表 1-1）

表 1-1　関連ドキュメント

ドキュメント名	内容
FPN システム説明書	FPN の概要と考え方について記したドキュメント
GSCIP 基本設計書	FPN を構築するためのセットワークセキュリティアーキテクチャ GSCIP の基本設計書
DPRP 基本仕様書	GSCIP に実装される動的処理解決プロトコル DPRP に関する基本仕様書
PCCOM 基本仕様書	GSCIP に実装される暗号プロトコル PCCOM の基本仕様書
GMS 設計仕様書	GMS を設計，構築するための基本仕様書
Basic 版 810API 説明書 1.00	Sony 製指紋照合装置 FIU-800 シリーズの API 説明書

2. 用語

SPAIC に使用する記号を以下のように定義する.

2.1. GPACK (GSCIP PACKage)

GSCIP を実現するソフトウェア．GPACK はカーネルに実装されるモジュール群から構成される.

2.2. 一般端末 (Term: Terminal)

クライアント端末．GPACK が実装されていない通常の端末で，移動することを考慮していない.

2.3. GSCIP 装置 (GE: GSCIP Element)

GSCIP を実装した装置．移動することを考慮している．GE には次の 3 種類に分類される.

2.3.1. GES (GSCIP Element for Software)

一般端末 Term に GPACK をインストールしたタイプの GE. 主にクライアントやサーバに該当する

2.3.2. GEN (GSCIP Element for Network)

ルータタイプの GE．サブネットワークを構築して，配下の一般端末 Term を保護する.

2.3.3. GEA (GSCIP Element for Adapter)

ブリッジタイプの GE．主に重要なサーバの直前に設置して，サーバを保護する.

2.4. 管理装置 (GMS: Group Management Server)

GE にグループ番号，グループ鍵とグループ共通鍵の情報を配送する装置．GMS は定期的にグループ鍵およびグループ共通鍵を更新する.

2.5. グループ鍵 (GK: Group Key)

GE が保持する暗号鍵．GK 番号とグループ番号が対応する．鍵サイズは 32byte.

2.6. 共通鍵 (CK: Group Common Key)

全ての GE が共通に持つ暗号鍵．鍵サイズは 32byte.

2.7. 鍵情報 (KInfo: Key Information)

GK と CK の情報．この情報は表 2-1 に示す項目から構成される．鍵情報は[Number, Version]のように記述される．グループ番号でグループの同一性確認，鍵バージョンで GK の新旧を確認する．特に GK に関する情報を GKI，GKInfo と表記する.

表 2-1 鍵情報

項目	内容
Number	GK の番号
Version	GK のバージョン

2.8. ユーザ ID (uID: User ID)

各ユーザに一意的に割り当てられている ID.

2.9. パスワード (PW: Password)

ユーザ認証を行うためのユーザパスワード.

2.10. 生体情報テンプレート (T: Biological Information Template)

ユーザ認証を行うための指紋テンプレート.

2.11. IC カード (IC Card)

ユーザが GES を利用するために必要なカード．uID，PrI/PuI，PuS，PW，T を記録する.

2.12. IC カード秘密鍵 (PrI: Private Key of IC Card)

暗号化/復号処理を行うための暗号鍵，IC カード公開鍵とペアとして GMS よって生成される.

2.13. IC カード公開鍵 (PuI: Public Key of IC Card)

暗号化/復号処理を行うための暗号鍵，IC カード秘密鍵とペアとして GMS よって生成される.

2.14. GMS 秘密鍵 (PrS: Private Key of GMS)

暗号化/復号処理を行うための暗号鍵，GMS 公開鍵とペアとして生成される.

2.15. GMS 公開鍵 (PuS: Public Key of GMS)

暗号化/復号処理を行うための暗号鍵，GMS 秘密鍵とペアとして生成される.

2.16. 乱数 Ni (Random Number of Initiator)

リプレイ攻撃を防止するための乱数．IC カードによって生成される.

2.17. 乱数 Nr (Random Number of Responsor)

リプレイ攻撃を防止するための乱数．GMS によって生成される.

2.18. DH1

GES と GMS 間で鍵を共有するための Diffie-Hellman 交換値の一つ．IC カードによって生成される．

2.19. DH2

GES と GMS 間で鍵を共有するための Diffie-Hellman 交換値の一つ. GMS によって生成される．

2.20. K

GES と GMS 間で安全に暗号通信を行うための共通鍵．DH1 と DH2 よって生成される．

2.21. クッキーCi (Cookie of Initiator)

DoS 攻撃を防止するためのクッキー．GE によって生成される．

2.22. クッキーCr (Cookie of Responsor)

DoS 攻撃を防止するためのクッキー．GMS によって生成される．

2.23. 鍵配送要求パケット (Key_REQ: Key-Request)

SPAIC パケットの一つ．ユーザ認証情報などを暗号化するために，GE が IC カードに対して公開鍵などの情報配送を要求する際に生成される．

2.24. 鍵配送応答パケット (Key_RES: Key-Response)

SPAIC パケットの一つ. Key-REQ を受信した IC カードが公開鍵を配送する際に生成される．

2.25. Cookie 配送要求パケット (Cookie_REQ: Cookie-Request)

SPAIC パケットの一つ．DoS 攻撃を防止するために，GE が GMS に対して Cookie の配送を要求する際に生成される．

2.26. Cookie 配送応答パケット (Cookie_RES: Cookie-Response)

SPAIC パケットの一つ. Cookie_REQ を受信した GMS が Cookie を配送する際に生成される．

2.27. ユーザ認証情報配送パケット (CertUser_DIST)

SPAIC パケットの一つ．GES がユーザ認証情報を IC カードに配送する際に生成される．

2.28. IC Card 署名情報配送パケット (SignIC_DIST)

SPAIC パケットの一つ．IC カードが IC カードの署名情報を GES に配送する際に生成される．

2.29. 情報配送パケット (Info_DIST)

SPAIC パケットの一つ．GES が IC カードの署名情報などを GMS に配送する際に生成される．

2.30. GMS 署名情報配送パケット (SignMS_DIST)

SPAIC パケットの一つ．GMS が GMS の署名情報を GES に配送する際に生成される．

2.31. GK と CK 配送要求パケット (GKey_REQ)

SPAIC パケットの一つ. GES が GMS に対して GK と CK の配送を要求する際に生成される．

2.32. GK と CK 配送応答パケット (GKey_RES)

SPAIC パケットの一つ．GKey_REQ を受信した GMS が GK と CK を GES 配送する際に生成される．

3. SPAIC の構成

3.1. システム構成

SPAIC を GSCIP に適用する場合のシステム構成を図 3-1　システム構成に示す．移動を考慮したユーザまたは重要なサーバにソフトウェアをインストールした GES（GE realized by Software），ルータに機能を実装した GEN（GE for Network），一般の端末 Term，およびグループ鍵を管理する GMS（Group Management Server）から構成される．GEN はサブネットを構成し，配下の一般端末 Term を保護する．GSCIP では同一の共通鍵を所持する GE の集合を同一の通信グループとして定義する．この共通鍵をグループ鍵 GK と呼ぶ．

SPAIC は，GE の起動時に鍵管理装置 GMS（Group Management Server)と通信を行うことによって，同一グループの端末と暗号化通信を行うためのグループ鍵 GK の取得を行う．

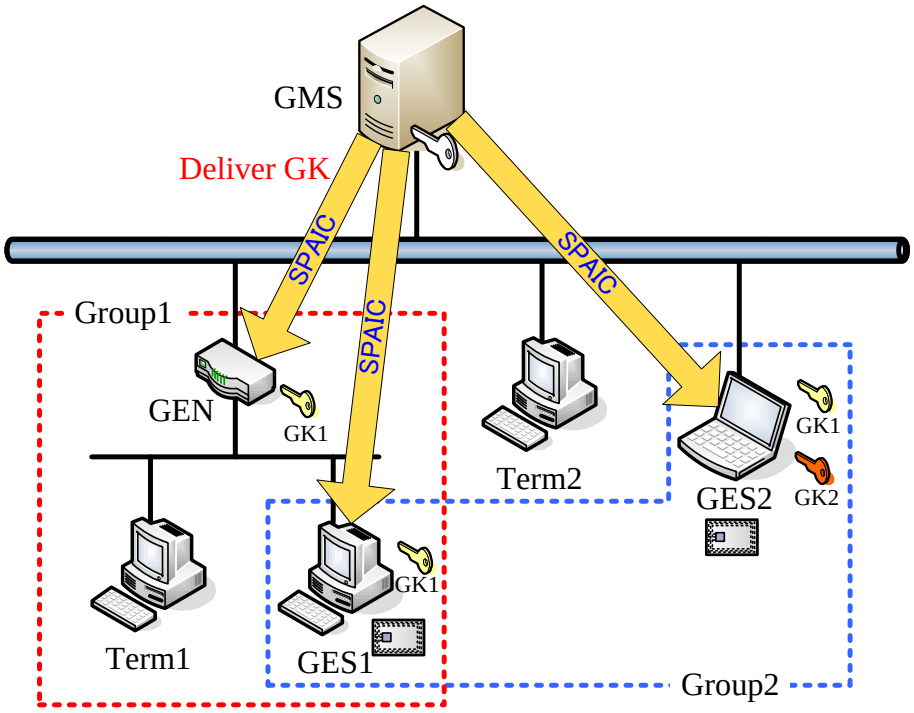


図 3-1　システム構成

3.2. SPAIC シーケンス

SPAIC の詳細シーケンスを図 3-2 に示す.

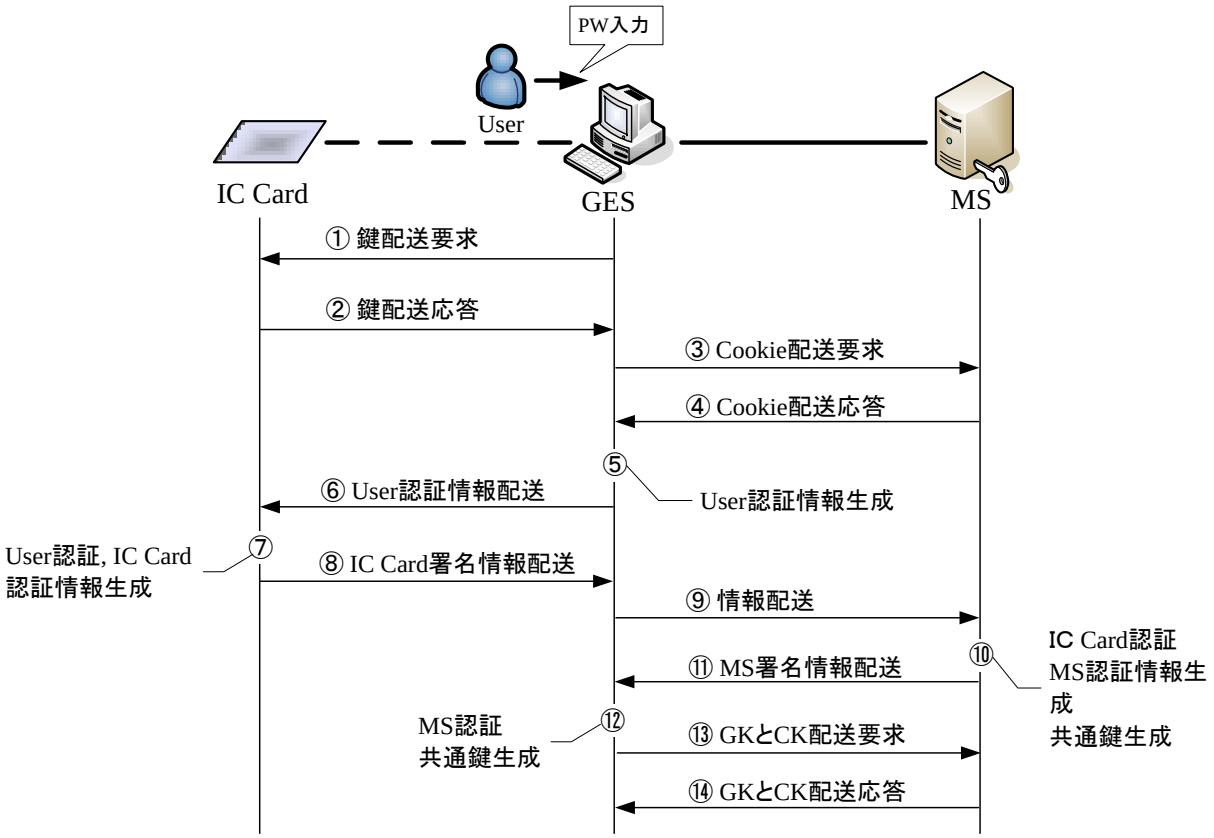


図 3-2 SPAIC のシーケンス

GES の電源を立ち上げ後．IC カードを IC カードリーダにかざすと，GES との間にコネクションが確立され， SPAIC シーケンスを開始する．

(1) 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) ICカードはユーザID(uID), ICカード公開鍵PuI, サーバ公開鍵PuS, 乱数Niを送信する．

(3) 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．

(4) 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．

(5) 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，ICカードから受け取った乱数Niとユーザ認証情報(PW, T)をPuIで暗号化する．同時にサーバから受け取った乱数NrをPuSで暗号化する．また，Diffie-Hellman交換値DH1を生成する．

(6) (5)で作成した認証情報をICカードへ送信する．

(7) ICカードではICカード秘密鍵PrIを利用してPW，T，Niを取り出しユーザ認証を行う．

更に，生成したNiと比較する．ユーザ認証後，PuSで暗号化された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．

(8) (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．

(9) 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．

(10)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，サーバ秘密鍵PrSを利用してNrを取り出し，生成したNrと比較する．その後，Diffie-Hellman交換値DH2を生成し，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．

(11) (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．

(12)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DH2を取得する．更に，DH1，DH2を利用して共通暗号鍵Kを生成する．

(13)GMSと暗号化通信を行うために，GESはGMSにクループ鍵を要求する．

(14)GMSはグループ鍵をGESに配送する．

4. 暗号処理仕様

4.1. 公開鍵暗号処理

公開鍵暗号処理仕様は，表 4-1 のとおりである．

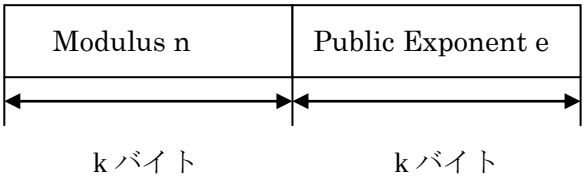
表 4-1　公開鍵暗号仕様

項目	内容
暗号アルゴリズム	RSA (PKCS モード※1)
鍵長	512 ビット，1024 ビット，2048 ビット
暗号化範囲	GES/IC Card 間の認証，署名情報の生成と検証

4.1.1. 公開鍵ファイル

公開鍵暗号方式の公開鍵（Public Key）を格納するファイルである．データ部の形式を図 4-1 に示す.

公開鍵ファイルの作成には，*FIU_GenerateKeyPair()*関数※2，または *FIU_CreateFile()*関数を用いる．このとき，表 4-2 に示すように必要なヘッダ項目を設定する．



k = （キーのビット長）／ 8

図 4-1　公開鍵ファイルのデータ部のフォーマット

※1：このモードでは，入力されたデータにデータ長を復元できるようにパディングを行い，もとのデータ長に復元して出力します．パディング方法については，PKCS#1 の仕様を参照してください.

※2：頭に「FIU_」がついている関数はソニー製指紋照合ユニット FIU-800 シリーズの API 関数である．詳しくは「Basic 版 810API 説明書 1.00」を参照してください.

表 4-2 公開鍵ファイルに必要なヘッダ項目

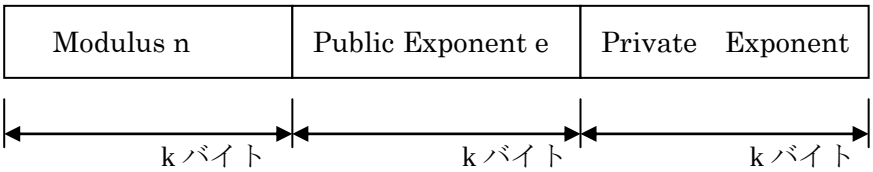
ヘッダ項目	作成時に必要な設定	変更
ファイルタイプ	FIU_PUBLIC_KEY_FILE	×
アプリケーション名	-	○
ラベル	-	○
プライベートフラグ	*2	*1
Sensitive フラグ	*2	*1
Unextract フラグ	*2	*1
ライトプロテクトフラグ	*2	○
ローカルフラグ	(読み出しのみ)	-
AlwaysSensitive フラグ	(読み出しのみ)	-
NeverExtractable フラグ	(読み出しのみ)	-
指の指定	-	×
テンプレートフォーマット	-	×
キータイプ	FIU_KEY_RSA	×
キーのビット長	＜キーのビット長＞	×
キーの使用開始日	-	○
キーの失効日	-	○
サブジェクト	-	○
ID	-	○
暗号化フラグ	*2	×
復号フラグ	-	×
電子署名フラグ	-	×
電子署名の確認フラグ	*2	×
ラップフラグ	-	×
アンラップフラグ	-	×
Certificate タイプ	-	×
Issuer	-	○
シリアル番号	-	○
ユーザー定義項目	-	○

*1 ： FIU_FALSE から FIU_TRUE へのみ，変更可能.
*2 ： デフォルトで FIU_FALSE に設定される．FIU_TRUE に設定も可能.
- ： 設定の必要はないが，任意の値を設定できる.

4.1.2. 秘密鍵ファイル

公開鍵暗号方式の秘密鍵（Private Key）を格納するファイルである．キータイプは RSA 方式にのみ対応している．データ部の形式を図 4-2 に示す．データ部の modulus n，Public Exponent e は，ファイルのアクセス属性にかかわらず，読み出すことができる．Private Exponent d については，アクセス属性に従う．Private Exponent d の読み出しが許可されないときには，データ部の Private Exponent d の部分が 0（0x00）で埋められて出力される．

秘密鍵ファイルの作成には，*FIU_GenerateKeyPair()*関数，または *FIU_CreateFile()*関数を用いる．このとき，表 4-3 に示すように必要なヘッダ項目を設定する．



k = （キーのビット長）／ 8

図 4-2 秘密鍵ファイルのデータ部のフォーマット

表 4-3 秘密鍵ファイルに必要なヘッダ項目

ヘッダ項目	作成時に必要な設定	変更
ファイルタイプ	FIU_PRIVATE_KEY_FILE	×
アプリケーション名	-	○
ラベル	-	○
プライベートフラグ	*3	*1
Sensitive フラグ	*3	*1
Unextract フラグ	*2	*1
ライトプロテクトフラグ	*2	○
ローカルフラグ	(読み出しのみ)	-
AlwaysSensitive フラグ	(読み出しのみ)	-
NeverExtractable フラグ	(読み出しのみ)	-
指の指定	-	×
テンプレートフォーマット	-	×
キータイプ	FIU_KEY_RSA	×
キーのビット長	＜キーのビット長＞	×
キーの使用開始日	-	○
キーの失効日	-	○
サブジェクト	-	○
ID	-	○
暗号化フラグ	-	×
復号フラグ	*2	×
電子署名フラグ	*2	×
電子署名の確認フラグ	-	×
ラップフラグ	-	×
アンラップフラグ	*2	×
Certificate タイプ	-	×
Issuer	-	○
シリアル番号	-	○
ユーザー定義項目	-	○

*1 ： FIU_FALSE から FIU_TRUE へのみ，変更可能.

*2 ： デフォルトで FIU_FALSE に設定される．FIU_TRUE に設定も可能.

*3 ： デフォルトで FIU_TRUE に設定される．FIU_FALSE に設定も可能.

- ： 設定の必要はないが，任意の値を設定できる.

4.2. 共通鍵暗号処理

共通鍵暗号処理仕様は，表 4-4 のとおりである．

表 4-4　共通鍵暗号仕様

項目	内容
暗号アルゴリズム	DES
鍵長	8 ビット，16 ビット，24 ビット
暗号化範囲	GES/GMS 間の暗号処理

4.2.1. 共通鍵ファイル

共通鍵暗号方式のキーを格納するファイルである．キータイプは DES 方式にのみ対応している．8 バイト，16 バイト，24 バイト長の DES キーを作成できる．

共通鍵ファイルの作成には，*FIU_GenerateSymmetricKey()*関数 *FIU_CreateFile()*関数を用いる．このとき，表 4-54-5 に示すように必要なヘッダ項目を設定する．

表 4-5 共通鍵ファイルに必要なヘッダ項目

ヘッダ項目	作成時に必要な設定	変更
ファイルタイプ	FIU_SYMMETRIC_KEY_FILE	×
アプリケーション名	-	○
ラベル	-	○
プライベートフラグ	*3	*1
Sensitive フラグ	*3	*1
Unextract フラグ	*2	*1
ライトプロテクトフラグ	*2	○
ローカルフラグ	(読み出しのみ)	-
AlwaysSensitive フラグ	(読み出しのみ)	-
NeverExtractable フラグ	(読み出しのみ)	-
指の指定	-	×
テンプレートフォーマット	-	×
キータイプ	＜キータイプ＞	×
キーのビット長	-	×
キーの使用開始日	-	○
キーの失効日	-	○
サブジェクト	-	○
ID	-	○
暗号化フラグ	*2	×
復号フラグ	*2	×
電子署名フラグ	-	×
電子署名の確認フラグ	-	×
ラップフラグ	-	×
アンラップフラグ	-	×

Certificate タイプ	-	×
Issuer	-	○
シリアル番号	-	○
ユーザー定義項目	-	○

*1 ： FIU_FALSE から FIU_TRUE へのみ，変更可能.
*2 ： デフォルトで FIU_FALSE に設定される．FIU_TRUE に設定も可能.
*3 ： デフォルトで FIU_TRUE に設定される．FIU_FALSE に設定も可能.
- ： 設定の必要はないが，任意の値を設定できる.

4.3. ディジタル署名処理

ディジタル署名処理仕様は，表 4-6 のとおりである.

表 4-6 ディジタル署名仕様

項目	内容
暗号アルゴリズム	RSA (PKCS モード)
鍵長	512 ビット，1024 ビット，2048 ビット
暗号化範囲	IC カード/GMS の署名の生成・検証

4.3.1. 公開鍵証明書ファイル

RSA 公開鍵の証明書のデータを格納するファイルです. 公開鍵証明書ファイルの作成には，*FIU_CreateFile()*関数を用いる．このとき，表 4-7 に示すように必要なヘッダ項目を設定する.

表 4-7 公開鍵証明書ファイルに必要なヘッダ項目

ヘッダ項目	作成時に必要な設定	変更
ファイルタイプ	FIU_CERTIFICATE_FILE	×
アプリケーション名	-	○
ラベル	-	○
プライベートフラグ	*2	*1
Sensitive フラグ	*2	*1
Unextract フラグ	*2	*1
ライトプロテクトフラグ	*2	○
ローカルフラグ	(読み出しのみ)	-
AlwaysSensitive フラグ	(読み出しのみ)	-
NeverExtractable フラグ	(読み出しのみ)	-
指の指定	-	×
テンプレートフォーマット	-	×
キータイプ	-	×
キーのビット長	-	×
キーの使用開始日	-	○
キーの失効日	-	○

サブジェクト	-	○
ID	-	○
暗号化フラグ	-	×
復号フラグ	-	×
電子署名フラグ	-	×
電子署名の確認フラグ	-	×
ラップフラグ	-	×
アンラップフラグ	-	×
Certificate タイプ	FIU_X509	×
Issuer	-	○
シリアル番号	-	○
ユーザー定義項目	-	○

*1 ： FIU_FALSE から FIU_TRUE へのみ，変更可能.

*2 ： デフォルトで FIU_FALSE に設定される．FIU_TRUE に設定も可能.

- ： 設定の必要はないが，任意の値を設定できる.

4.4. 他の暗号アルゴリズム

ディジタル署名情報，クッキーの生成時に利用するハッシュ関数は MD5 とする.

GES/GMS 間の暗号通信を行うための暗号鍵は，Diffie-Hellman により生成する.

5. SPAIC パケットフォーマット

5.1. パケットの種類

SPAIC パケットは GES と GMS との通信に必要なパケットである．パケット処理シーケンスを図 5-1 に示す．パケットの定義を表 5-1 に示す．

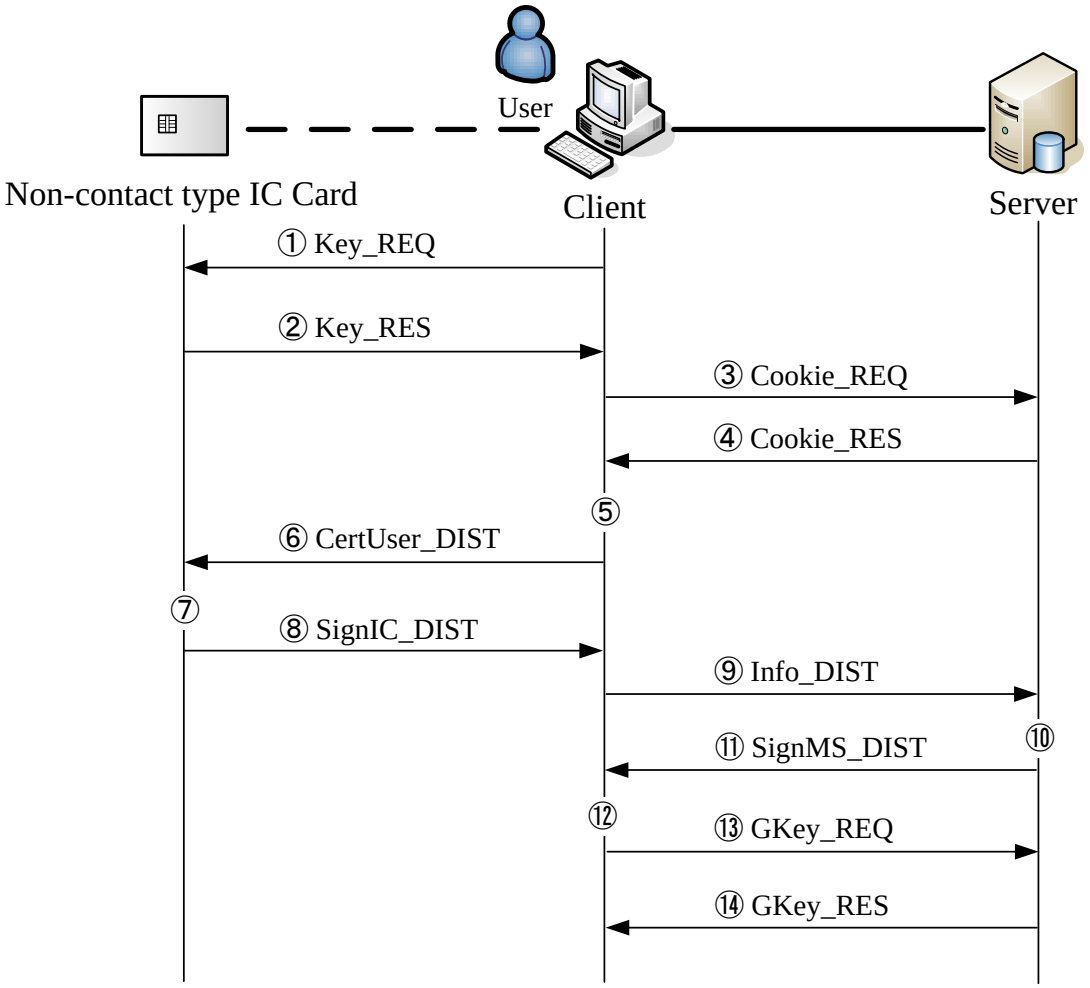


図 5-1　処理シーケンス（パケット名）

表 5-1　パケットタイプの定義

Type	パケット名称	説明
1	Key_REQ	鍵配送要求
2	Key_RES	鍵配送応答
3	Cookie_REQ	Cookie 配送要求
4	Cookie_RES	Cookie 配送応答

5	CertUser_DIST	User 認証情報配送
6	SignIC_DIST	IC Card 署名情報配送
7	Info_DIST	情報配送
8	SignGMS_DIST	GMS 署名情報配送
9	GKey_REQ	GK と CK 配送要求
10	GKey_RES	GK と CK 配送応答
11	SPAIC_ERR	エラー処理

5.2. パケットの構造

SPAIC パケットは UDP をベースに定義されている．パケットの構造を図 5-2 に示す.

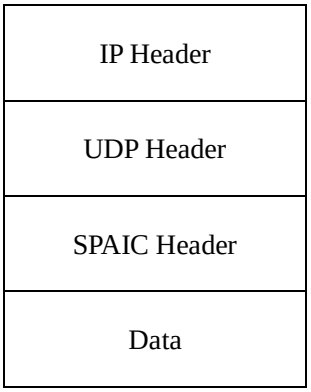


図 5-2　SPAIC パケットの構造

5.3. SPAIC ヘッダフォーマット

SPAIC ヘッダは図 5-3 の構造を持つ.

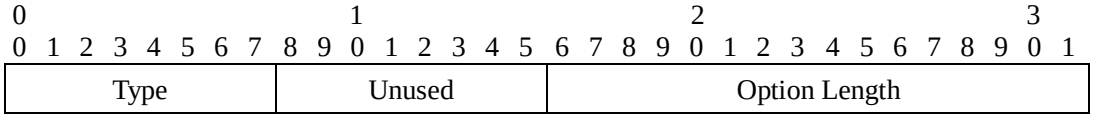


図 5-3 SPAIC ヘッダフォーマット

表 5-2　SPAIC ヘッダフィールド

フィールド	サイズ	値
Type	1	定義（表 5-1）
Unused	1	未使用
Option Length	2	追加されるオプション長

以下，各パケットデータについてのフォーマットを示す.

5.3.1. Key_REQ（鍵配送要求）

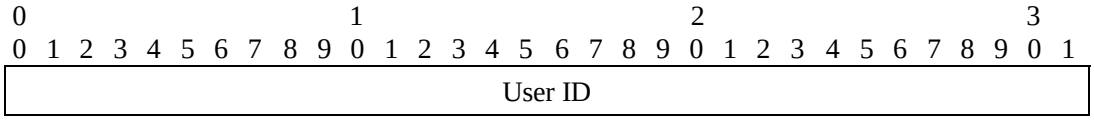


図 5-4 鍵配送要求フォーマット

表 5-3 鍵配送要求フィールド

フィールド	サイズ（byte）	値
User ID	4	ユーザ ID

5.3.2. Key_RES（鍵配送応答）

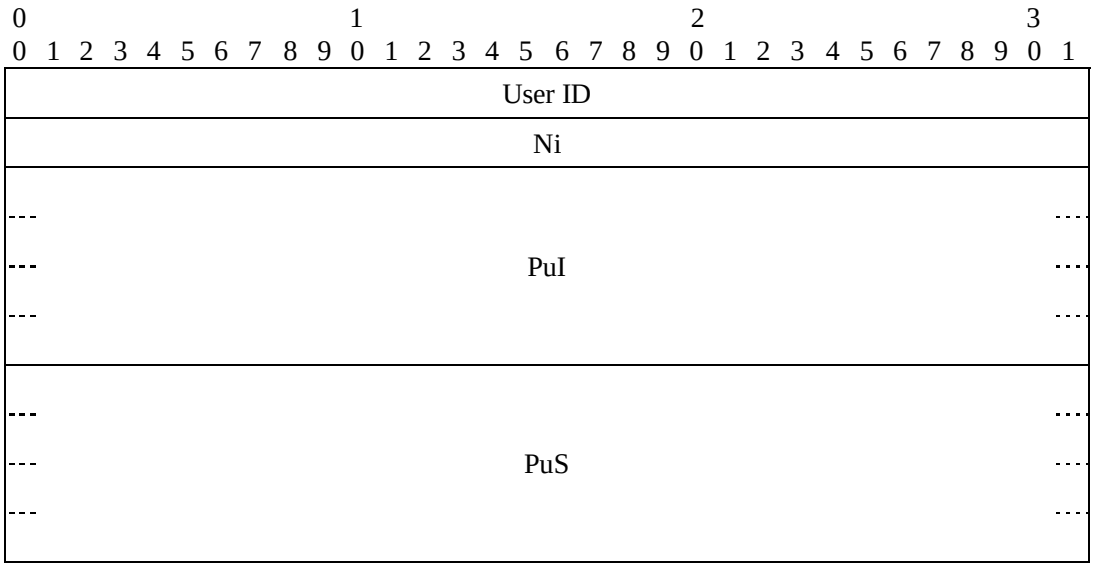


図 5-5 鍵配送応答フォーマット

表 5-4 鍵配送応答フィールド

フィールド	サイズ（byte）	値
User ID	4	ユーザ ID
Ni	4	乱数
PuI	128	IC カードの公開鍵
PuS	128	GMS の公開鍵

5.3.3. Cookie_REQ（Cookie 配送要求）

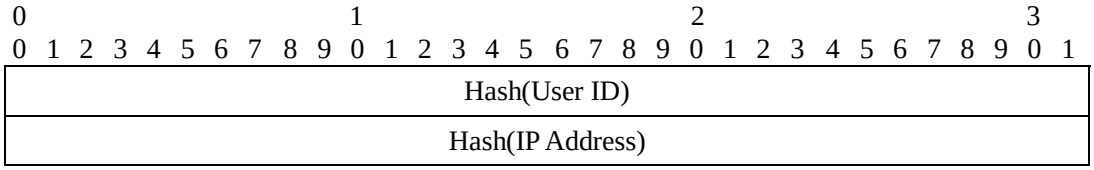


図 5-6　Cookie 配送要求フォーマット

表 5-5　Cookie 配送要求フィールド

フィールド	サイズ（byte）	値
Hash(User ID)	4	ユーザ ID のハッシュ値
Hash(IP Address)	4	GES の IP Address

5.3.4. Cookie_RES（Cookie 配送応答）

図 5-7　Cookie 配送応答フォーマット

表 5-6　Cookie 配送応答フィールド

フィールド	サイズ（byte）	値
Set-Cookie	4	クッキー名（NAME=VALUE）
Domain	2	GMS の Domain_Name
Secure	2	SSL オプション
Ns	4	乱数

5.3.5. Cert_User_DIST（User 認証情報配送）

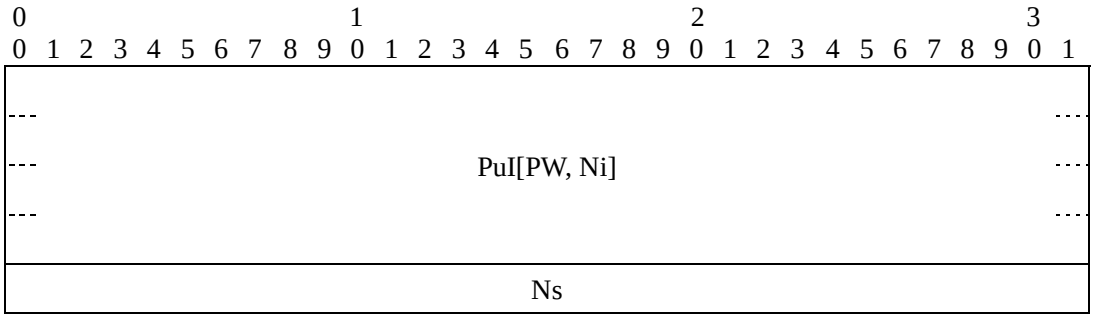


図 5-8　User 認証情報配送フォーマット

表 5-7　User 認証情報配送フィールド

フィールド	サイズ（byte）	値
PuI[PW, Ni]	128	PuI で暗号化した PW と Ni
Nr	4	乱数

5.3.6. Sign_IC_DIST（IC Card 署名情報配送）

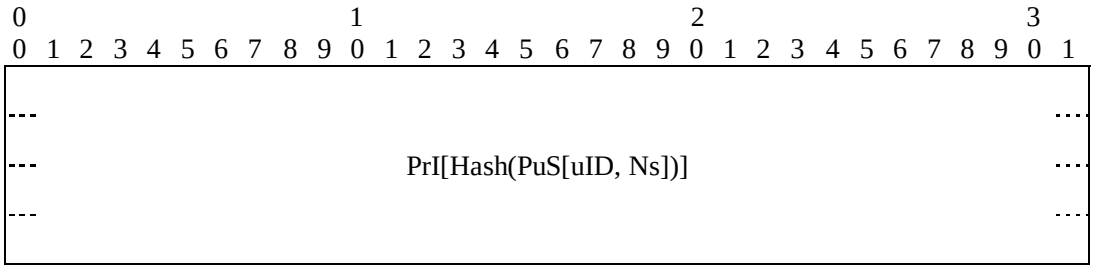


図 5-9　IC Card 署名情報配送フォーマット

表 5-8　IC Card 署名情報配送フィールド

フィールド	サイズ（byte）	値
PrI[Hash(PuS[uID, Nr])]	128	ディジタル署名

5.3.7. Info_DIST（情報配送）

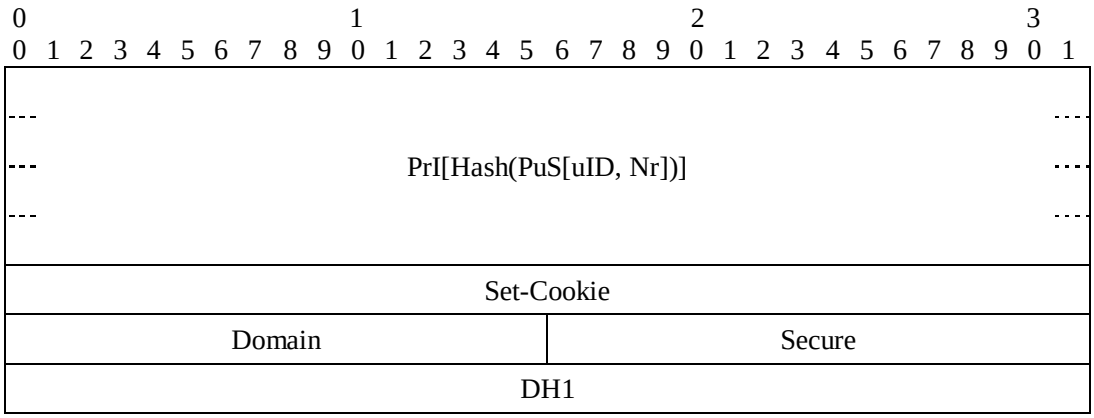


図 5-10　情報配送フォーマット

表 5-9　情報配送フィールド

フィールド	サイズ（byte）	値
PrI[Hash(PuS[uID, Nr])]	128	ディジタル署名
Set-Cookie	4	クッキー名（NAME=VALUE）
Domain	2	GMS の Domain_Name
Secure	2	SSL オプション
DH1	4	DH 交換値 1

5.3.8. Sign_GMS_DIST（GMS 署名情報配送）

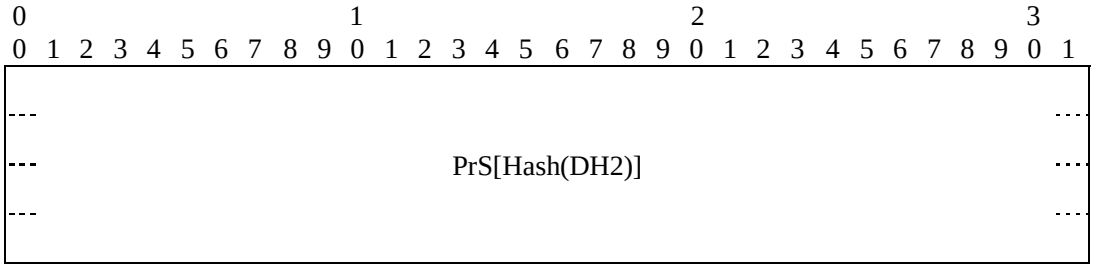


図 5-11　GMS 署名情報配送フォーマット

表 5-10　GMS 署名情報配送フィールド

フィールド	サイズ（byte）	値
PrS[Hash(DH2)]	128	ディジタル署名

5.3.9. GKey_REQ（GK と CK 配送要求）

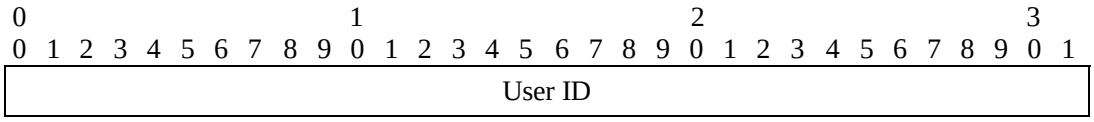


図 5-12　GK と CK 配送要求フォーマット

表 5-11　GK と CK 配送要求フィールド

フィールド	サイズ（byte）	値
User ID	4	ユーザ ID

5.3.10. GKey_REP（GK と CK 配送応答）

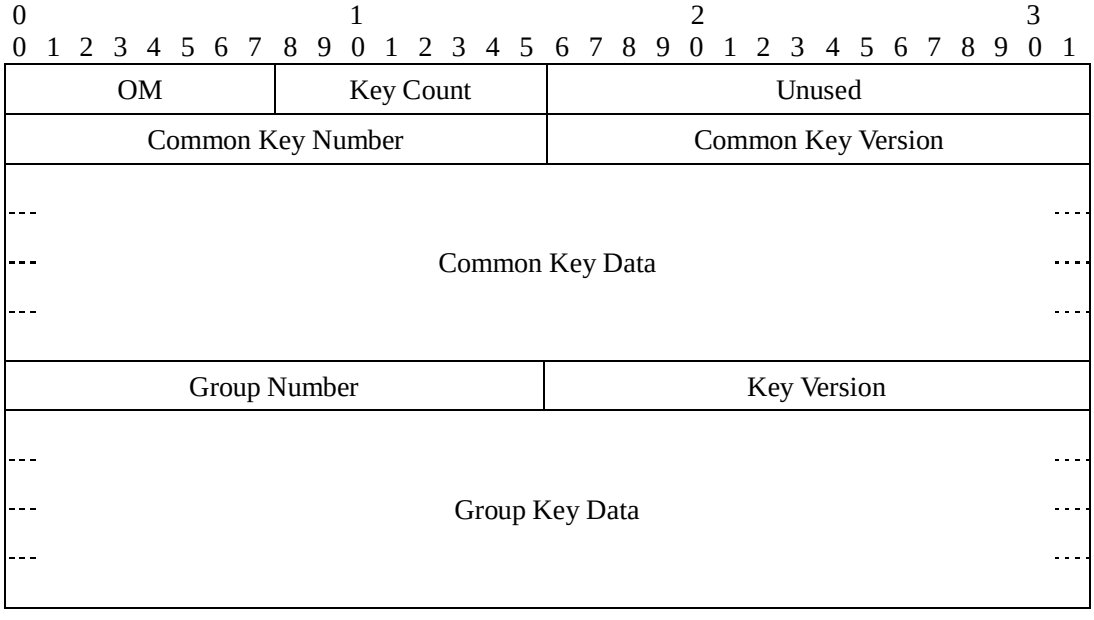


図 5-13　GK と CK 配送応答フォーマット

表 5-12　　GK と CK 配送応答フィールド

フィールド	サイズ（byte）	値
OM	1	GES 動作モード
Key Count	1	GK と CK の数
Common Key Number	2	共通鍵番号　（固定　0）
Common Key Version	2	共通鍵のバージョン
Common Key Data	16	共通鍵
Group Number	2	グループ番号
Group Key Version	2	グループ鍵のバージョン
Group Key Data	16	グループ鍵

6. モジュール構成

6.1. モジュール構成と機能

SPAIC は図 6-1 に示すモジュールから構成される.

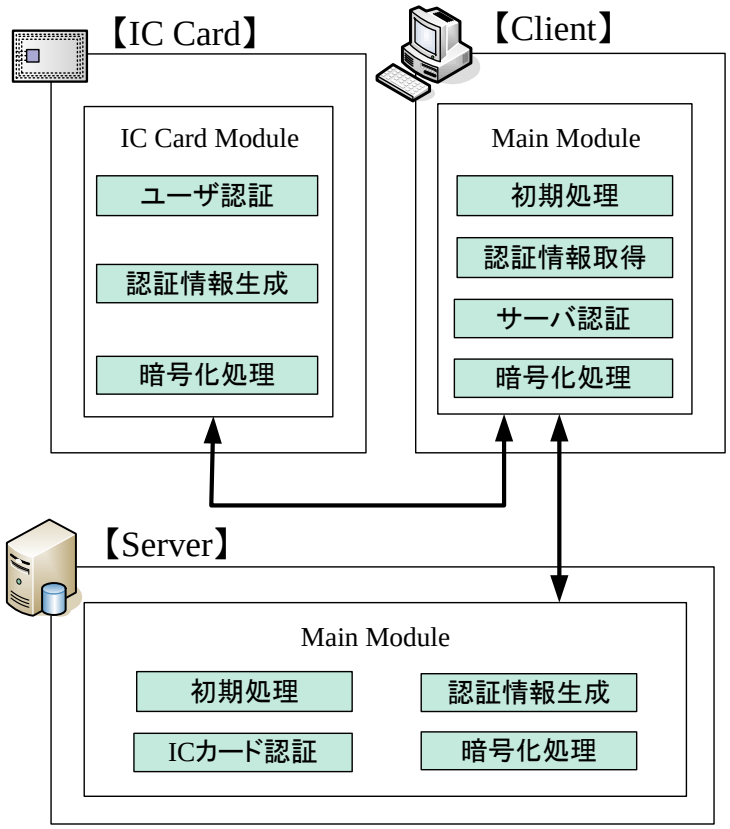


図 6-1 モジュール構成

IC カード，GES および GMS モジュールの主な機能を表 6-1 から表 6-3 に示す.

表 6-1 クライアント側のモジュール構成と機能

モジュール	機能
メインモジュール	主処理および各サブモジュールの呼び出し
初期処理	初期化処理，IC カードとのコネクション確立
認証情報取得	認証情報となるパスワードの取得
サーバ認証	サーバ署名情報の検証
暗号化処理	暗号化/復号化処理，ディジタル署名の検証

表 6-2　IC カード側のモジュール構成と機能

モジュール	機能
ユーザ認証	パスワードの認証処理
認証情報生成	IC カード認証情報の生成
暗号化処理	暗号化/復号化処理，ディジタル署名の生成

表 6-3　サーバ側のモジュール構成と機能

モジュール	機能
メインモジュール	主処理および各サブモジュールの呼び出し
初期処理	初期化処理
IC カード認証	IC カード署名情報の検証
認証情報生成	サーバ認証情報の生成
暗号化処理	暗号化/復号化処理，ディジタル署名の生成/検証

6.2. 状態遷移図

IC カード，GES および GMS の状態遷移を図 6-2 に示す．クライアントは電源投入後，SPAIC 用プログラムを起動する．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，認証処理を開始する．

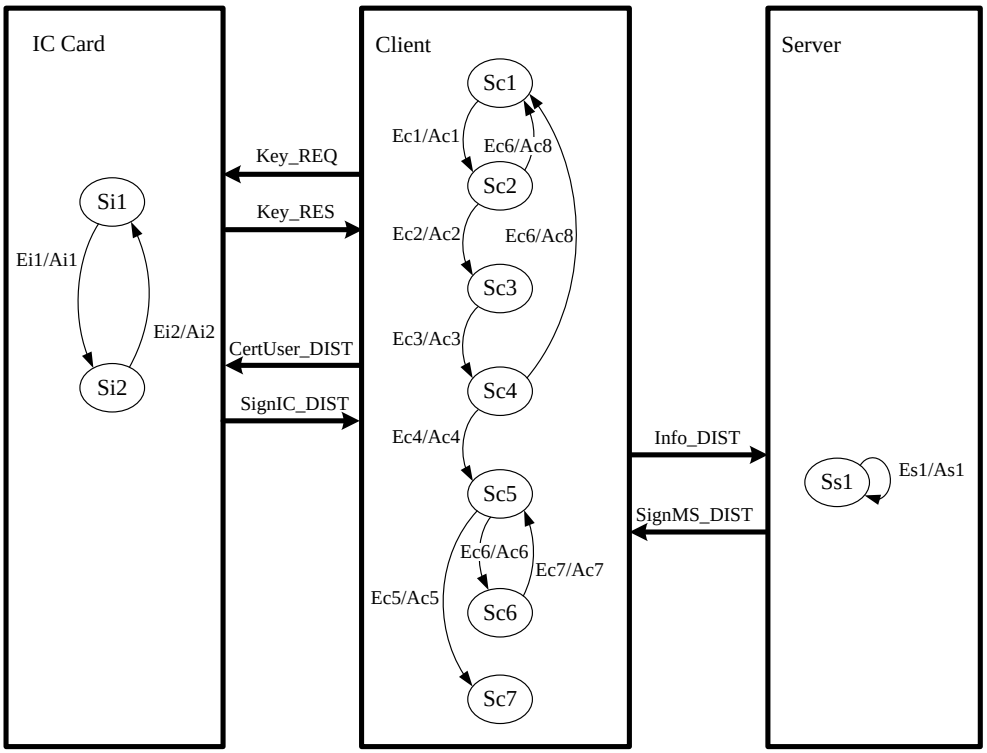


図 6-2　SPAIC の状態遷移図

状態遷移図におけるそれぞれの記号の意味を表 6-4 から表 6-6 に説明する.

表 6-4 状態定義

端末	状態記号	意味
Client	Sc1	ログイン指示待ち
	Sc2	Key_RES 待ち
	Sc3	PW 入力待ち
	Sc4	SignIC_DIST 待ち
	Sc5	SignMS_DIST 待ち
	Sc6	Info_DIST 再送指示待ち
	Sc7	一般通信可能状態
IC Card	Si1	Key_REQ 待ち
	Si2	CertUser_DIST 待ち
Server	Ss1	Info_DIST 受信待ち

表 6-5 イベント定義

端末	イベント記号	意味
Client	Ec1	ログイン指示
	Ec2	Key_RES 受信
	Ec3	PW 入力
	Ec4	SignIC_DIST 受信
	Ec5	SignMS_DIST 受信
	Ec6	TimeOut
	Ec7	Info_DIST 再送指示
IC Card	Ei1	Key_REQ 受信
	Ei2	CertUser_DIST 受信
Server	Es1	Info_DIST 受信

表 6-6　アクション定義

端末	アクション記号	意味
Client	Ac1	Key_REQ 送信，タイマ起動
	Ac2	ログイン画面表示
	Ac3	DH1 生成，E_{PuI}[PW]生成， CertUser_DIST 送信，タイマ起動
	Ac4	Info_DIST 送信，タイマ起動
	Ac5	サーバ認証，K 生成
	Ac6	エラー表示＆再送指示画面表示
	Ac7	Info_DIST 送信，タイマ起動
	Ac8	エラー表示＆ログイン指示画面表示
IC Card	Ai1	Key_RES 送信
	Ai2	PrI で E_{PuI}[PW]を復号， CertUser_DIST 生成， CertUser_DIST 送信
Server	As1	PuI で検証，IC カード認証， DH2 生成，SignMS_DIST 生成， SignMS_DIST 送信，K 生成

6.2.1. GES における状態遷移図

GES における状態遷移を図 6-3 に示す.

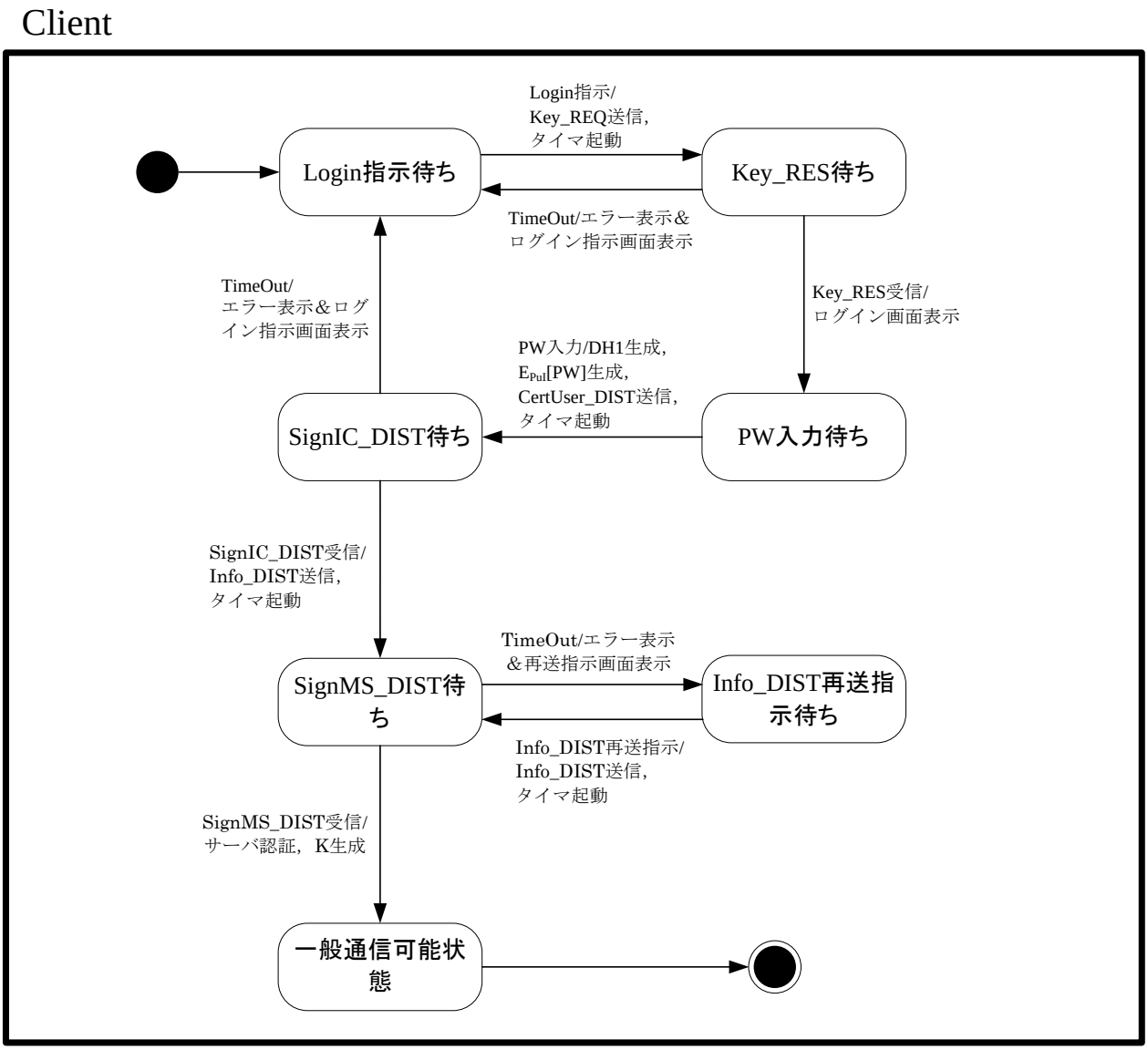


図 6-3 GES の状態遷移図

6.3. 状態遷移表

IC カード，GES および GMS の状態を図 6-4 で示す.

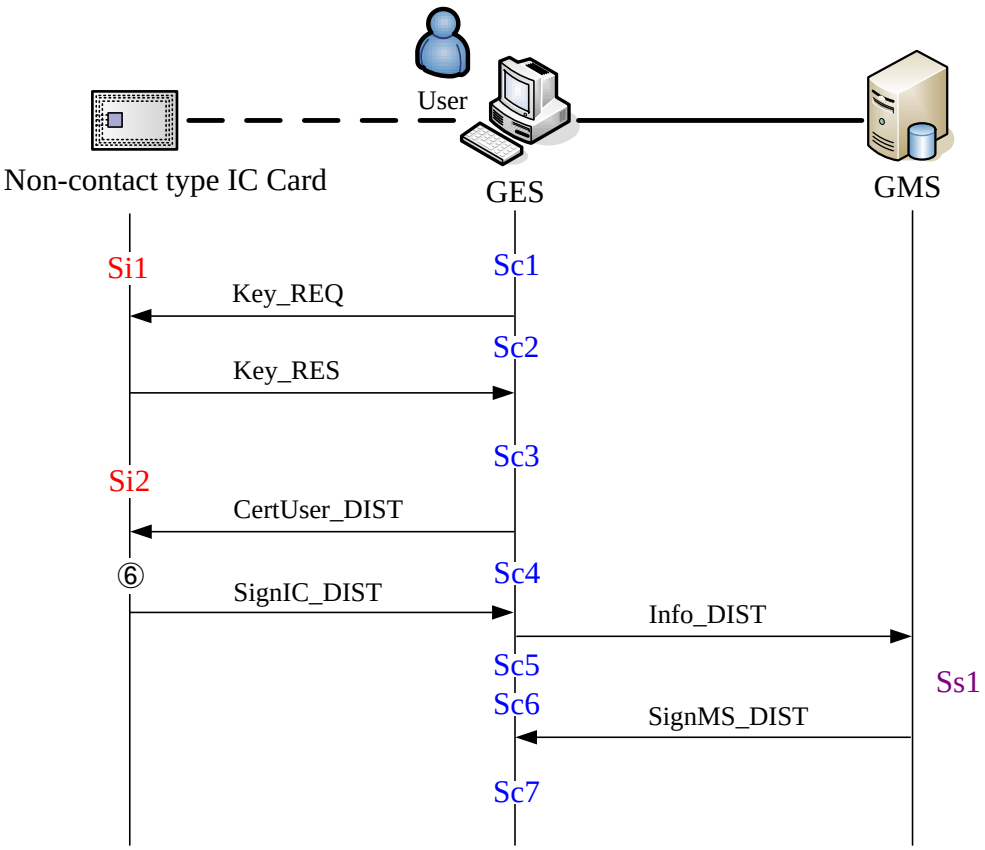


図 6-4　SPAIC の状態

GES は電源投入後，SPAIC 用プログラムを起動する．IC カードを IC カードリーダにかざすと，GES との間にコネクションが確立され， 認証処理を開始する．以降の状態遷移は表 6-7 から表 6-9 に示す.

(1) GES

- プログラムが起動されたらログイン指示画面を表示し状態 Sc1 に遷移する.
- 「－」はありえない事象.
- 「無視」はなにもせず，状態も変化しない.
- タイマ値の大きさは状態により異なる. タイマ起動していなければタイムアウトはありえない事象となる.
- Info_DIST の再送指示はユーザ指示に従うこととした．再送を繰り返してもタイムアウトを繰り返す場合は GMS がダウンしていると考えられる．この場合プログラムの終了はユーザに任せる.
- K 生成は Sc5 の一連の処理の中で実行する.

表 6-7　GES 側の状態遷移

状態 / イベント	Sc1 ログイン 指示待ち	Sc2 Key_RES 待ち	Sc3 PW 入力待ち	Sc4 SignIC_DIST 待ち	Sc5 SignMS_DIS T 待ち	Sc6 再送指示 待ち	Sc7 一般通信 可能状態
ログイン指示	Key_REQ 送信 タイマ起動 Sc2 へ	－	－	－	－	－	－
Key_RES 受信	無視	ログイン画面 表示 Sc3 へ	無視	無視	無視	無視	無視
PW 入力	－	－	DH1 生成 E_{Pul}[PW]生成、 CertUser_DIST 送信 タイマ起動 Sc4 へ	－	－	－	－
SignIC_DIST 受信	無視	無視	無視	Info_DIST 送信 タイマ起動 Sc5 へ	無視	無視	無視
SignMS_DIST 受信	無視	無視	無視	無視	GMS 認証 K 生成 Sc7 へ	無視	無視
TimeOut	－	エラー表示& ログイン指示 画面表示 Sc1 へ	－	エラー表示& ログイン指示 画面表示 Sc1 へ	エラー表示& 再送指示画面 表示 Sc6 へ	－	－
Info_DIST 再送指示	－	－	－	－	－	Info_DIST 送信 タイマ起動 Sc5 へ	－

(2) IC カード

・ 必要な情報が既に書き込まれ，GES からの Key_REQ 待ちであることが前提.
・ IC カードは待ち受けだけなのでタイマ処理は不要.
・ 状態 Si2 でも Key_REQ 受信はありうる（GES からの再送時）.

表 6-8 IC カード側の状態遷移

状態 / イベント	Si1 Key_REQ 待ち	Si2 CertUser_DIST 待ち
Key_REQ 受信	Key_RES 送信 Si2 へ	Key_RES 送信 Si2 へ
CertUser_DIST 受信	無視	PrI で E_{PuI}[PW]を復号 CertUser_DIST 生成 CertUser_DIST 送信 Si1 へ

(3) GMS

・ 必要な情報登録を終え，GES からの Info_DIST 待ちであることが前提.
・ GES は待ち受けだけなのでタイマ処理は不要.
・ K は GES と同様一連の処理の中で生成する.
・ GMS は常に同じ状態 Ss1 であり状態遷移は発生しない.

表 6-9 サーバ側の状態遷移

状態 / イベント	Ss1 Info_DIST 受信待ち
Info_DIST 受信	PuI で検証 IC カード認証 DH2 生成 SignMS_DIST 生成 SignMS_DIST 送信 K 生成

7. 開発ツールおよび開発言語

7.1. 開発ツール

- Visual C++ 2005
- FIU-800SDK Basic，OpenSSL（暗号ライブラリとして使用）

7.2. 開発言語

C++言語