平成25年度　修士論文

邦文題目

# クライアントを自由に選択可能な認証プロトコルTSSAPの提案

英文題目

# Proposal of TSSAP : Terminal Selectable Secure Authentication Protocol

情報工専攻 渡邊研究室

(学籍番号: 123430016)

五島　秀典

提出日: 平成26年1月31日

名城大学理工学部

# 内容要旨

企業においては情報漏洩の防止が重要な課題である. 情報漏洩の原因の 3 割はノート PC 等のモバイル機器の盗難, 紛失によるものと言われている. そこで社外に情報を持ち出さずに, 必要に応じてクライアント PC から社内システムにリモートアクセスする方法が注目されている. このときクライアントは固定されることなく選べることが望ましい. このようなシステムには確実な認証と暗号化が要求される. 本論文では近年普及が著しいスマートフォンに認証情報を保持させ, 初期情報を一切所持しないクライアントを利用可能とするプロトコル TSSAP(Terminal Selectable and Secure Authentication Protocol) を提案する.

# 目 次

# 第1章　はじめに

インターネットの普及に伴い，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増えている．また，企業においては情報漏洩の防止，情報管理の徹底が重要となっている．情報漏洩の原因の 3 割はノート PC 等のモバイル機器の盗難，紛失によるものと言われている [1]．そこで社外に情報を持ち出さずに，必要に応じてクライアント PC から社内システムに安全にアクセスするリモートアクセスが注目されている．社内システムにアクセスするにはクライアントとサーバ間で正しい認証と暗号鍵の共有が必須であり，セキュリティの強度を上げるために，２要素以上の認証が好ましい．２要素以上の認証とは，ID，パスワードだけの認証ではなく，ID，パスワード認証と一緒に，生体認証や，複製不可能，もしくはしづらいものをキーとして認証する認証方法である．

２要素以上の認証では ID，パスワードだけの認証と違い，パスワードの問題点である，パスワードの流出，推測された場合でも認証を否定できる．そして，このようなシステムにはユーザの視点から考えるとホテルのパソコン，自宅のパソコン等，異なるクライアントからでもサーバへアクセスできることが望ましい．

このような認証システムの既存技術の例として，非接触型の IC カードをユーザが所持する方式がある [2-8]．この方式は IC カード，クライアントに共有鍵を持たせることによりクライアント，IC カード間の暗号通信が行える．しかし，この方式はクライアントが共有鍵を保持する必要があり，クライアント共有鍵が漏洩する可能性がある [2-5]．また，クライアントに秘密情報を持たないモデルとして，Keymobile を利用した方法，ワンタイムパスワードを利用した方法がある．ワンタイムパスワードとは，使い捨てのパスワードを生成し，それを利用して認証する方法である．この方法の場合，2 要素目としてユーザはトークンを持ち，認証の際はトークンに表示された乱数を認証の際に，ID，PW の入力と共に入力することで認証できる．また，Keymobile[10] を利用した方法では Keymobile と呼ばれる耐タンパ性のある記憶媒体をスマートフォンに装着し，認証する方法である [11]．しかし，どちらも，この方式はフィッシングを利用した中間者攻撃に弱いとされており，悪意ある第 3 者に乗っ取られる危険性がある．

本論文ではスマートフォンに認証情報を持つデバイスとして利用し，初期情報を一切持たないクライアントに対し、サーバから重要な情報を配送することを可能とするプロトコル TSSAP(Terminal Selectable and Secure Authentication Protocol) を提案する．

# 第2章　既存の技術

## 2.1　IC カードを利用した方法

図 2.1 に非接触 IC カードを利用した認証方式を示す．この方式では IC カード-クライアント間は無線通信で行われるため両者に共有鍵を埋め込む事前共有鍵を利用する方式が JICSAP により定義されている [9]．クライアントと IC カードの間は上記の共有鍵を使って認証と暗号通信を行う．ユーザは IC カードを所有しており，IC カード内の認証情報をクライアントから入力する認証情報で確認することにより認証する．しかし，この方式はクライアントに共有鍵を保持させているためクライアントから秘密情報が漏洩する可能性がある．また，漏洩した場合システム全体に影響を与える可能性がある．さらにセキュリティ面を考えると共有鍵を定期的に更新する必要があり，鍵の管理が煩雑になるという課題がある．

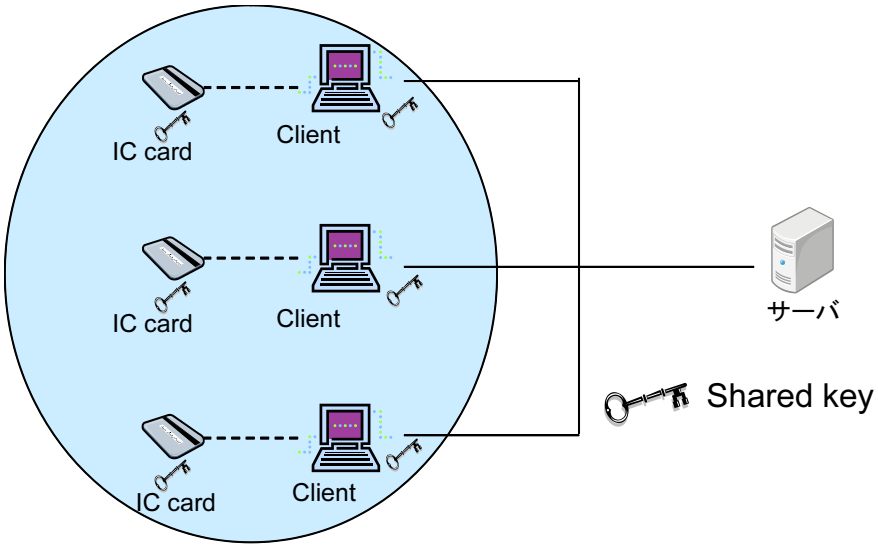


**図 2.1　IC カードを利用した方法**

## 2.2　ワンタイムパスワードを利用した方法

図 2.2 にワンタイムパスワードを利用した方法を示す．
この方式ではユーザはワンタイムパスワードを生成するトークンを所持する．トークンとはワンタイムパスワードを生成させるハードウェア機器である．トークンとクライアント間は USB などで直接接続し，クライアント-サーバ間は SSL を利用して暗号化している．ワンタイムパスワードを生成する際によく使われる方法として時刻同期型である．時刻同期型ではワンタイムパスワード生成の際，現在時刻を種として利用する方法でトークン側でその時刻に合わせてその時有効なパスワードが表示される．このため一定時間ごとにパスワードが変化することとなる．サーバ側では入力されたパスワードと現在，有効なパスワードを比較することで認証している．サーバ側では誰がどのトークンを利用しているのか，各トークンの表示されているパスワードを把握している．この方法ではパスワード認証などと一緒に利用される．

クライアントに秘密鍵を所持しないのでクライアントからの情報漏洩はない．しかし，この方式は接続先のサーバをユーザが確認できないことから正規のサイトではなく，偽のサイトへアクセスしていると気づくことができず，偽サーバへ接続してしまう．このためフィッシングを利用した中間者攻撃に弱い．

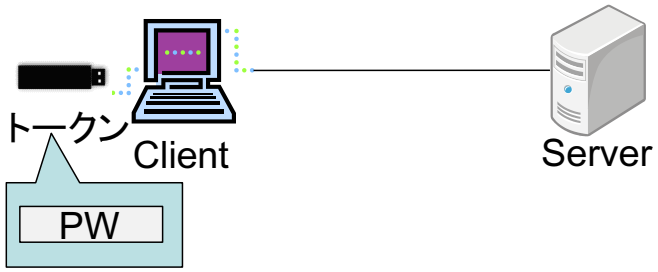


**図 2.2**

# 第3章　TSSAPの提案

TSSAP(Terminal Selectable and Secure Authentication Protocol) はスマートフォンを利用し，秘密情報を一切保持しないクライアントを利用できる認証プロトコルとなっている．クライアントに秘密情報を一切所持しないためユーザが自由にクライアントを選択できることに加えクライアントから秘密情報が漏洩する心配がないという利点がある．

## 3.1　システムモデル

TSSAP で想定するシステムモデルを図 3.1 に示す．本システムの構成要素はスマートフォン，クライアント，サーバである．ユーザは秘密情報を格納したスマートフォンを所持し，クライアントを操作する．スマートフォン，クライアントには Bluetooth に対応し，アプリケーションをあらかじめインストールする必要がある．

クライアント-サーバ間の通信は，任意のネットワークで接続できる．スマートフォン-クライアント間の通信は Bluetooth とする．他に Wi-Fi や ZigBee などの近距離通信が考えられるが Bluetooth は多くのモバイル機器に対応していることや通信距離を変化させることができる．このため TSSAP では Bluetooth を採用した．前提条件としてユーザとクライアントの通信距離が近いため，スマートフォン-クライアント間の中間者攻撃はできないものとする．

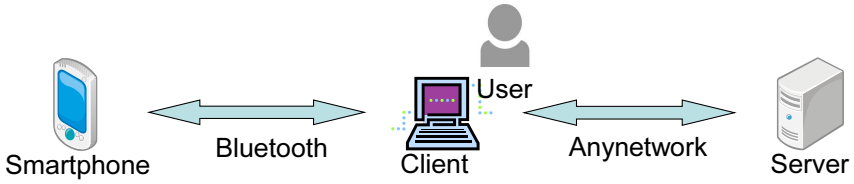


**図 3.1　TSSAP のシステムモデル**

## 3.2 記号の定義

TSSAP で使用する記号を以下のように定義する.

| 記号 | 説明 |
|---|---|
| uID | ユーザ ID |
| PuSP | スマートフォン公開鍵 |
| PrSP | スマートフォンの秘密鍵 |
| PuS | サーバ公開鍵 |
| PrS | サーバ秘密鍵 |
| PW | パスワード |
| Kc | クライアントが生成する共有鍵 |
| Rs | サーバが生成する乱数 |
| Cc | クライアントが生成するクッキー |
| Cs | サーバが生成するクッキー |
| Ex[y] | x で y を暗号化 |
| Sx[y] | x で y にディジタル署名 |
| Key_REQ | 配送要求パケット |
| Key_REP | 配送応答パケット |
| Cookie_REQ | クッキー配送要求パケット |
| Cookie_REP | クッキー配送応答パケット |
| CertUser_DIST | ユーザ認証パケット |
| SignSP_DIST | スマートフォン署名情報配送パケット |
| Info_DIST | 情報配送パケット |
| SignS_DIST | サーバ署名情報配送パケット |

## 3.3 TSSAP の動作

図 3.3 に TSSAP のシーケンスを示す．図中の記号はパケット名を示しており，パケット名後の ( ) の内容はパケットの情報を示している．スマートフォン-クライアント間の Bluetooth のペアリングは完了しているものとする．即ち，この間の通信は Bluetooth の共通鍵で暗号化される．

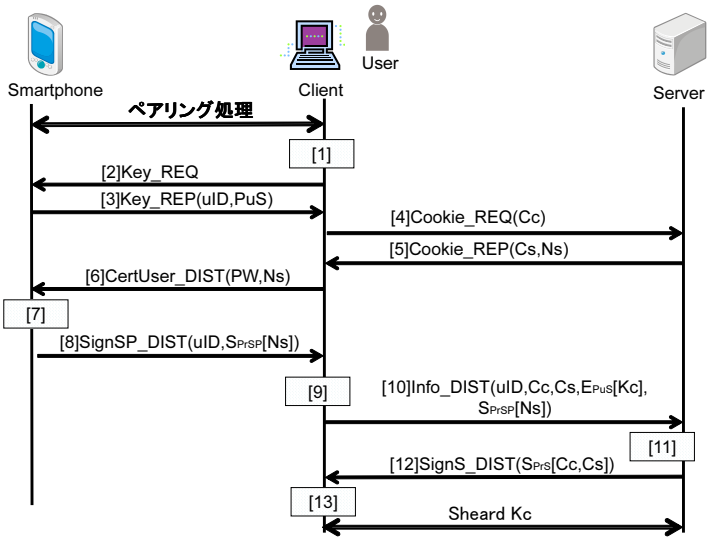


**図 3.2　TSSAP シーケンス**

1. ユーザ情報 (PW) の入力
   クライアント PC の画面に PW 入力画面が表示される．ユーザはこの画面に PW を入力する．

2. スマートフォンへ Key_REQ を送信
   ユーザがパスワードを入力し，OK をクリックすることで，クライアントはスマートフォンへサーバ公開鍵 PuS，ユーザ ID の情報配送を要求する．

3. Key_REP を送信
   スマートフォンはユーザ ID，サーバ公開鍵 PuS を送信する．

4. 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 Cc を生成して，サーバへ送信する．

5. Cookie_REP の送信
   サーバはリプレイアタックに対応するための乱数 Rs と DoS 攻撃防止のクッキー Cs を生成する．この時接続してきたクライアントの IP アドレスと生成したクッキーとを対応づけて記憶させておく．また，クッキー Cs と共に乱数 Rs をクライアントへ送信する．

6. CertUser_DIST の送信
   1. で入力された PW とサーバから受け取った乱数 Rs をスマートフォンへ送る．

7. スマートフォン秘密鍵の取得
   スマートフォンは受け取った PW で $E_{PW}$[PrSP] を復号する．

8. SignSP_DIST の送信
   スマートフォンはクライアントから受け取った乱数 Rs，PW にスマートフォン秘密鍵 PrSP でディジタル署名をし，クライアントへ送信する．

9. 共有鍵 Kc の生成
   クライアント自身が共有鍵 Kc を生成する．Kc をサーバ公開鍵 PuS で暗号化する．

10. Info_DIST の送信
    9. で生成した $E_{PuS}$[Kc] と 8. で生成した $S_{PrSP}$[Rs] と共に uID，Cc，Cs をサーバへ送信する．

11. ユーザ，スマートフォン認証と署名の作成
    受信した Rs とサーバが 4. で生成した Rs を比較する．$S_{PrSP}$[Rs] のディジタル署名をスマートフォン公開鍵 PuSP を用いて確認する．ここでディジタル署名が正しいと確認されれば，7. で復号したスマートフォン秘密鍵 PrSP が正しく復号されたことになるため，ユーザの入力した PW が正しいといえる．これより，ユーザ認証が完了する．また，クッキー Cc，Cs についても比較を行うことでスマートフォン認証が完了する．次に Cc，Cs にサーバ秘密鍵 PrS を用いてディジタル署名を行う．最後に，暗号化された Kc をサーバ秘密鍵 PrS で復号する．

12. SignS_DIST の送信
    シーケンス中の 11. で生成された $S_{PrS}$[Cc，Cs] をクライアントへ送信する．

13. サーバ認証
    $S_{PrS}$[Cc，Cs] のディジタル署名をサーバ公開鍵 PuS で確認することによりサーバ認証を行う．

以上の認証処理を行うことにより認証が完了し，クライアント-サーバ間で共有鍵を安全に共有できる．

## 3.4 TSSAP の初期情報

各機器が所有する初期情報を表 3.1 に示す．スマートフォンにはユーザ ID，ユーザの登録したパスワードで暗号化したスマートフォン秘密鍵が格納されている．次にサーバはサーバ秘密鍵，スマートフォン公開鍵，ユーザ ID が登録されている．そして，クライアントには初期情報が一切ない．ここでユーザの登録したパスワードで暗号化したスマートフォン秘密鍵が登録されている理由はスマートフォンは耐タンパ性を有しないため，生の秘密鍵のデータを保持するのは危険であるため暗号化することで秘密鍵の流出を防ぐことができる．また，初期情報はあらかじめオフラインで設定しておく必要がある．

**表 3.1　TSSAP の初期情報**

| 機器名 | 初期情報 |
|---|---|
| Smartphone | uID<br>Epw[PrSP]<br>PuS |
| Client | - |
| Server | PrS<br>PuSP<br>uID |

## 3.5 Bleutooth のペアリング

TSSAP の認証処理を実行するにあたってスマートフォンとクライアント間で Bluetooth のペアリングを行う必要がある [12,13]．プロファイルは SPP(Serial Port Profile) を使用する．スマートフォン側では Bluetooth を ON にし，端末を他の機器が検出できるように設定する (ペアリングモード)．次にクライアント側で Bluetooth 端末のスキャンを行い，ペア設定を開始する．Bluetooth のバージョンによりペアリングの動作は異なるが，セキュアシンプルペアリング対応のバージョンではクライアントとスマートフォンの画面に乱数が表示される．ユーザの目で両者を確認し，一致すれば両画面の OK をクリックすることによりペアリングが完了となる．Bluetooth ではペアリングを確立すると鍵交換が実行され，自動的にスマートフォン-クライアント間で鍵が共有される．以後の通信は Bluetooth の標準搭載の暗号化で実現する．

## 3.6 ユーザ認証について

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，記憶媒体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．

サーバ型認証は，クライアントで取得した認証情報を，スマートフォンを経由してサーバへ送信し，サーバで確認することで認証を行う．この認証方式では，サーバ側でユーザ認証とスマートフォン認証を一括して行う．ユーザ全員の情報をサーバ側で一括管理できるため，データの管理がしやすいが，サーバの管理体制が重要となる．このため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，クライアントで取得した認証情報をスマートフォンに送信してスマートフォン内でユーザ認証を行い，その後スマートフォン-サーバ間でスマートフォン認証を行う．この認証方式ではサーバにおけるスマートフォン認証がユーザ認証を兼ねこととなる．その理由はクライアントをユーザが操作することとスマートフォンをユーザが所持しているためサーバがスマートフォン認証を行うとユーザ認証と同意となるからである．しかしこの方式の場合，スマートフォンにユーザの認証情報を安全に格納する必要があるため，記憶媒体に耐タンパ性を必要とする．

TSSAP では，スマートフォンを記憶媒体として使用しており，スマートフォンに耐タンパ性を持たせることは難しい．仮にクライアント型認証に耐タンパ性を有しない記憶媒体を使用した際は記憶媒体からユーザ認証情報が流出してしまう危険がある．これらより，TSSAP ではサーバ型認証を採用する．

# 第4章　TSSAPの実装

## 4.1　TSSAPのモジュール構成

TSSAP のモジュールの構成図を図 4.1 に示す．

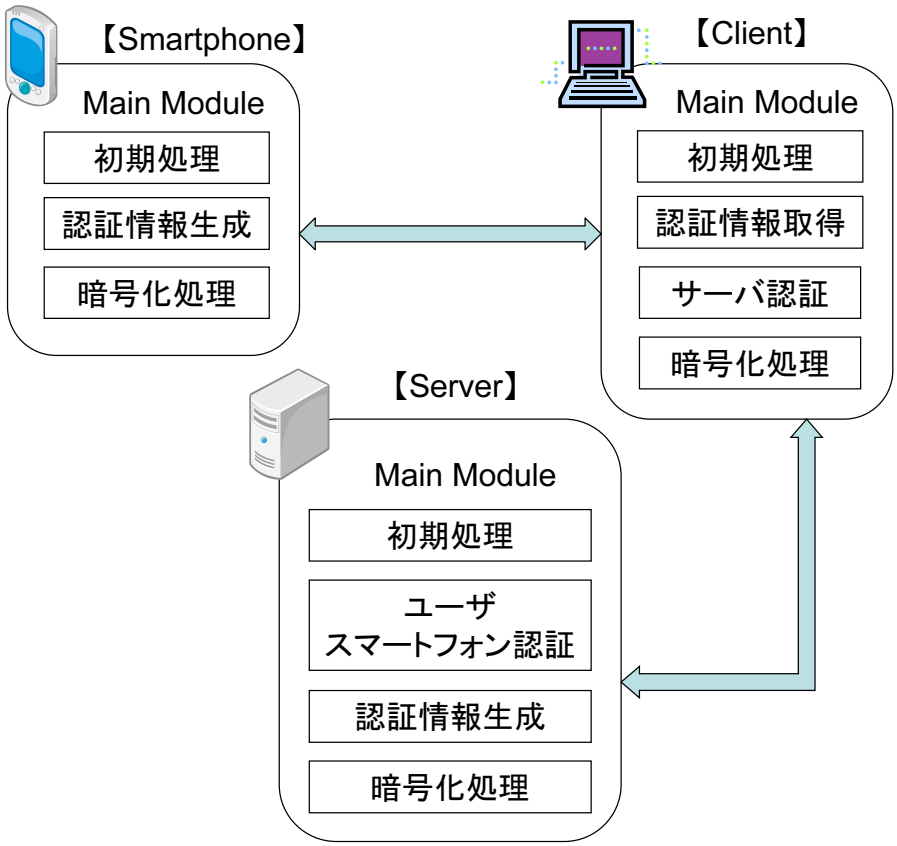


**図 4.1　TSSAP モジュール構成**

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．メインモジュールは処理状態を管理し，状態に対応したサブモジュールを呼び出す．暗号化処理はパケット通信における暗号/復号を行う．初期処理はシステムの初期化を行う．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ユーザが画面にパスワードを入力することで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する．

サーバ固有のモジュールはユーザ，スマートフォン認証，認証情報生成がある．ユーザ，スマートフォン認証モジュールはスマートフォンの署名情報を検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．

スマートフォン固有のモジュールは認証情報生成がある．認証情報生成モジュールはスマートフォン認証に必要な情報を生成する．

# 第5章　評価

## 5.1　実験環境

本実験では，TSSAP のシーケンスを開始し，認証完了するまでの合計時間を計測する．公開鍵暗号のアルゴリズムは，RSA(PKCS モード) 共通鍵暗号のアルゴリズムは AES とし，RSA の鍵は 1024bit，AES の鍵は 256bit とした．実験に用いた PC，スマートフォンの装置仕様を表 5.1 に示し，実験装置のネットワーク構成を図 5.1 に示す．

**表 5.1　装置の仕様**

| | 項目 | 内容 |
|---|---|---|
| PC1 | CPU | Core2 Quad 2.80GHz |
| | Memory | 2GB |
| | OS | Windows7 64bit |
| | language | C++ |
| PC2 | CPU | Core2 2.80GHz |
| | Memory | 2GB |
| | OS | Windows7 64bit(VM) |
| | language | C++ |
| Smartphone | CPU | Snapdragon S4 MSM8960 1.5GHz |
| | Memory | 1GB |
| | OS | Android 4.0 |
| | language | Java |

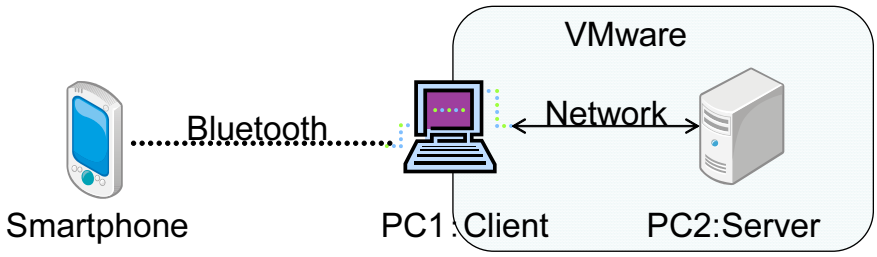


**図 5.1　ネットワークの構成図**

実験には，PC1 と VMware 上に作成した PC2 とスマートフォンの 3 台を用いた．PC1 と PC2 はブリッジ接続でネットワークに接続されており，スマートフォン，PC1 は Bluetooth で接続している．PC1 にはクライアントの処理プログラムを，PC2 にはサーバの処理プログラムを実行させた．

## 5.2　性能評価

アプリケーションを起動し，ユーザがパスワードを入力してから認証が完了するまでの合計時間を測定する．測定方法は，QueryPerformanceCounter 関数 [14] を利用した．QueryPerformanceCounter 関数とは，高分解能パフォーマンスカウンタを取得でき，カウンタの差分を周波数で割ることで処理時間を算出できる．今回は，クライアントがデータ送信し，レスポンスが返っくるまでを分割して測定する．分割したシーケンス図を図 5.2 に示す．また，表 5.2 に計測結果を示す．

(1) はクライアントが Key_REQ の生成から，スマートフォンから Key_REP を受信するまでの時間，(2) はクライアントが Cookie_REQ の生成から Cookie_REP を受信するまでの時間，(3) はクライアントが CertUser_DIST の生成から SignSP_DIST を受信するまでの時間，(4) はクライアントが Info_DIST の生成から SignS_DIST を受信するまでの時間示している．しかし，(3) に関してはスマートフォン内で行うディジタル署名をする部分が未実装であるため (3) では Bluetooth での送受信時間＋スマートフォンの処理を同内容で C 言語で実装したプログラムを PC1 で別途測定した結果を示す．すべての計測結果は 10 回計測した平均値を示している．

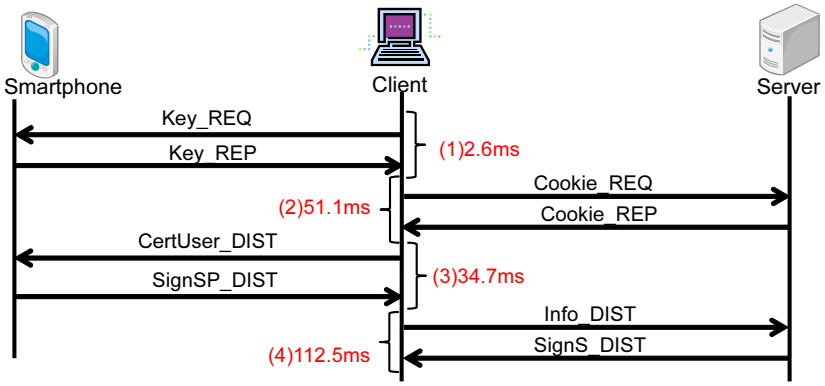


**図 5.2　TSSAP 実測値を示すシーケンス**

**表 5.2　TSSAP の実測値**

| 番号 | 測定内容 | 時間 (ms) |
|---|---|---|
| (1) | Key_REQ 生成から Key_RES 受信まで | 2.6 |
| (2) | Cookie_REQ 生成から Cookie_REP 受信まで | 51.1 |
| (3) | CertUser_DIST 生成から SignSP_DIST 受信まで | 34.7 |
| (4) | Info_DIST 生成から SignS_DIST 受信まで | 112.5 |
| 合計 | | 200.9 |

計測結果は Key_REQ 生成から Key_RES 受信までの時間が 2.6ms，Cookie_REQ 生成から Cookie_REP 受信までの時間が 51.1ms，CertUser_DIST 生成から SignSP_DIST 受信までの時間は Bluetooth 送受信時間が 2.6ms，PC1 を利用し，C 言語でスマートフォンと同内容の処理をさせた時間が 32.1ms，合計で 34.7ms なる．最後に Info_DIST 生成から SignS_DIST 受信までの時間が 112.5ms となり，合計時間が 200.9ms となった．

Cookie_REQ 生成から Cookie_REP 受信までの時間には主な処理としてはクライアント，サーバ側でクッキーの生成する時間が含まれる．CertUser_DIST 生成から SignSP_DIST 受信までの時間には主な処理として AES 復号化，ディジタル署名の生成が含まれる．この処理は本来はスマートフォンが処理すべき処理だが PC1 で処理した時間を示してる．Info_DIST 生成から SignS_DIST 受信までの時間には主な処理としてクライアント側で AES 鍵の生成，RSA 暗号化，ディジタル署名の検証，サーバ側ではディジタル署名の作成，検証，RSA 復号化などの処理が含まれる．

(2) の Cookie_REQ 生成から Cookie_REP 受信までの時間が処理の割に時間のかかっている理由は，クッキー生成する際，乱数の生成，IP アドレスを取得，時間情報を取得し，これらのハッシュ関数に通すのだが解析の結果，IP アドレスの取得に時間がかかっていた．

測定は，仮想マシンによって行ったため，クライアント-サーバ間で，通信遅延はほとんどないが，実ネットワークでは通信遅延が大きく影響を及ぼす．グローバルネットワーク上における国内における RTT(Round Trip Time) は 20ms 程度であるため，加算する必要がある．このため TSSAP を実環境で利用した際の認証にかかる処理時間は 220.9ms と推定できるため，システムの立ち上げ時にかかる処理としては，十分許容範囲となると考える．

## 5.3 既存方式との比較

事前共有鍵方式と TSSAP の比較を表 5.3 に示す．クライアントに格納する情報として事前鍵共有方式では，事前鍵と動作プログラム，ワンタイムパスワードは一切必要としていない，TSSAP 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．これらの情報で漏洩してはいけない情報を保持しているのは事前鍵共有方式の事前鍵であるので，事前鍵共有方式は×とした．

管理負荷では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ため×とした．一方 ワンタイムパスワード，TSSAP 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ため○とした．

認証端末-クライアント間の暗号に関しては，事前鍵共有方式では事前鍵で暗号化し，通信している．ワンタイムパスワードでは直接パソコンにつないでやり取りするため盗聴，中間者攻撃などの攻撃は受けない．TSSAP では Bluetooth の標準暗号化機能 [12] を利用することで暗号化している．すべての方式で安全といえるためすべて○とした．

クライアント-サーバ間の暗号については事前鍵共有方式と TSSAP は公開鍵暗号方式を利用して暗号化しており，ワンタイムパスワードでは SSL を利用した暗号化を行っている．どちらの方法も安全と言えるためすべて○とした．

中間者攻撃への対応では TSSAP，事前鍵共有方式では，エンドエンドでディジタル署名を確認することで防ぐことができる．接続先サーバの正当性を証明できるため○とした．ワンタイムパスワードの場合，ユーザが正規のサイトに似せた偽サイトにログインしてしまうことでフィッシングを利用した中間者攻撃が成立する可能性があるため×とした．

利便性については，事前鍵共有方式では IC カード，IC カードリーダ，事前鍵の格納されたクライアントが必要となるため×とし，ワンタイムパスワードはトークンのみを持ち歩くため○とした．TSSAP では普段持ち歩くスマートフォンのみだが，場合によっては Bluetooth のアダプタが必要になり，クライアントにもアプリをインストールする必要があるため△とした．以上より提案した TSSAP は既存技術よりも有用である．

**表 5.3　既存技術と TSSAP の比較**

| | IC カード | ワンタイムパスワード | TSSAP |
|---|---|---|---|
| クライアントに格納する情報 | × | ○ | ○ |
| 管理負荷 | × | ○ | ○ |
| 認証端末-クライアント間の暗号 | ○ | ○ | ○ |
| クライアント-サーバ間の暗号 | ○ | ○ | ○ |
| 中間者攻撃への耐性 | ○ | × | ○ |
| 利便性 | × | ○ | △ |

# 第6章　むすび

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，スマートフォン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 TSSAP の提案を行った．スマートフォンに暗号化した暗号鍵を持たせ，サーバで認証することでクライアントに初期情報を持たなくとも，スマートフォン-クライアント-サーバ間で確実な認証を可能にした．　性能評価においては TSSAP が認証にかかる時間を測定した．結果，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，名城大学院理工学研究科情報工学専攻 渡邊晃教授に心より厚く御礼申し上げます．

本論文を制作するにあたり，終始御熱心なご指導と御教示を賜りました，名城大学院理工学研究科情報工学専攻 吉川雅弥教授，旭健作助授，鈴木秀和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，名城大学院理工学研究科情報工学専攻 渡邊研究室，鈴木研究室の皆様に心より感謝いたします．

## 参考文献


[1] NPO 日本ネットワークセキュリティ協会セキュリティ情報セキュリティ大学院大学, 2011 年情報セキュリティインシデントに関する調査報告書

[2] 磯部義明，三村昌弘, 瀬戸洋一，菊地良知，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，Vol.99-CSEC-4, No.24，pp.55—60.

[3] 磯部義明，三村真一，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，99-CSEC-4,Vol.99,No.24,pp.55-60.

[4] 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 .

[5] 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.

[6] 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 .

[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，121-C, No．9，pp．1429-1438.

[8] 束長俊，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（DICOMO2007）シンポジウム論文集，情報処理学会シンポジウム，No．3，pp．304-307.

[9] 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.

[10] 岡崎 司，畠山 誠基，佐藤 隆一，KeyMobile を用いた安全なデータ持ち出し，日立 TO 技報第 15 号

[11] 梅澤 克之，手塚 悟， 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，電子情報通信学会論文誌，J94-B No．4

[12] Specification of the Bluetooth System—Version 2．0 + EDR.

[13] Bluetooth Test Specification—RF， Part A，For Specification 2．0， Revision 2．0.

[14] Microsoft developer Network，http://msdn.microsoft.com/ja-jp/library/cc410966.aspx

# 研究業績

## 学術論文

なし

## 研究会・大会等

1. 五島 秀典，旭 建作，鈴木 秀和，渡邊 晃秘密情報を保持しないクライアントを用いた 認証プロトコルの提案電気関係学会東海支部連合大会，Sep.2011.
2. 五島 秀典, 旭 建作, 鈴木 秀和, 渡邊 晃秘密情報を一切保持しないクライアントを利用できる認証 プロトコルの提案情報処理学会第 74 回全国大会論文集, Mar.2012.
3. 五島 秀典, 旭 建作, 鈴木 秀和, 渡邊 晃クライアントを自由に選択できる認証プロトコル TSSAP の提案情報学ワークショップ 2012（WiNF2012）論文集，WiNF2012, Vol.2012，pp.105-108，Dec.2012.
4. 五島 秀典，旭 建作，鈴木 秀和，渡邊 晃クライアントを自由に選択可能な認証プロトコル TSSAP 情報学ワークショップ 2013（WiNF2013）論文集，WiNF2013, Vol.2013，Dec.2013.

# 付 録A　状態遷移図

## A.1　TSSAP の状態遷移図

TSSAP の状態遷移図を図 A.1 に示す.

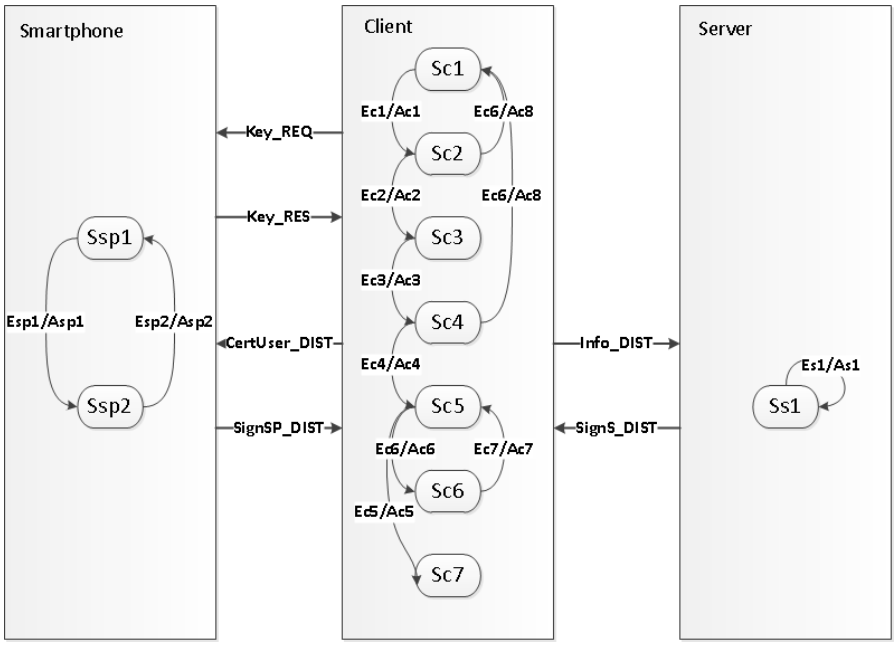


**図 A.1　TSSAP の状態遷移図**

状態遷移図におけるそれぞれの記号の意味を表 A.1 から表 A.3 に説明する.

**表 A.1　状態定義**

| 端末 | 状態記号 | 意味 |
|---|---|---|
| Client | Sc1 | ログイン指示待ち |
| | Sc2 | Key_REQ 待ち |
| | Sc3 | PW 入力待ち |
| | Sc4 | SignS_DIST 待ち |
| | Sc5 | SignS_DIST 待ち |
| | Sc6 | Info_DIST 待ち |
| | Sc7 | 一般通信可能状態 |
| Smartphone | Ssp1 | Key_REQ 待ち |
| | Ssp2 | CertUser_DIST 待ち |
| Server | Ss1 | Info_DIST 待ち |

**表 A.2　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_REQ 受信 |
| | Ec3 | PW 入力 |
| | Ec4 | SignS_DIST 受信 |
| | Ec5 | SignS_DIST 受信 |
| | Ec6 | TimeOut |
| | Ec7 | Info_DIST 再送指示 |
| Smartphone | Esp1 | Key_REQ 受信 |
| | Esp2 | CertUser_DIST 受信 |
| Server | Es1 | Info_DIST 受信 |

表 A.3　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 | CertUser_DIST 生成，CertUser_DIST 送信，タイマ起動 |
| | Ac4 | Info_DIST 生成，Info_DIST 送信，タイマ起動 |
| | Ac5 | サーバ認証 |
| | 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 |
| Smartphone | Asp1 | Key_REQ 送信 |
| | Asp2 | SignSP_DIST 生成，SignSP_DIST 送信 |
| Server | As1 | スマートフォン認証，SignS_DIST 生成，SignS_DIST 送信 |

## A.2　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，CertUser_DIST を生成し，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 を生成，Info_DIST を送信，タイマを起動し，状態 Sc5 に遷移する．SignS_DIST を受信したら，サーバ認証を行い，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．

## A.3　スマートフォンにおける状態遷移

スマートフォン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 Ssp1 であることを前提とする．スマートフォン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 Ssp2 に遷移する．CertUser_DIST を受信したら，CertUser_DIST 生成および送信など，一連の処理を行う．状態 Ssp2 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）．

## A.4　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，スマートフォンの認証を行う．その後，SignS_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．