4Q-1

IC カードによる認証システムの構築法

齊藤祐輔† 萩原洋一‡ 櫻田武嗣‡ 並木美太郎††
東京農工大学工学部情報コミュニケーション工学科†
東京農工大学総合情報メディアセンター‡ 東京農工大学大学院共生科学技術研究部††

1. はじめに

近年、認証系の用法は多岐に渡り、方式も様々なものがある。中でも FeliCa[1]に代表される IC カードは、カード内に複数のデータを個別に格納できることから、認証の鍵として[2]や、SmartOn[3]、NSAS[4]などで利用されている。SmartOn、NSAS は共に認証セキュリティシステムであり、IC カードを PC に接続したカードリーダへ挿入することで Windows へログオンでき、IC カードを抜くと PC がロックされ、退席時の不正利用を防ぐことができる。しかし、FeliCa との通信をシステムが占有するため、他のアプリケーションから FeliCa を利用することはできない。

本論文では、この問題を解決したシステムの構築法について述べる。

2. 目標

複数の認証処理を単一の IC カードを認証の鍵として行なうことが本構築法の目標である。本論文では例として認証の鍵に FeliCa を採用し、FeliCa 内に格納した一つあるいは複数のアカウントとパスワード情報にアクセスして、Windows ログオン認証、HTTP 基本認証の処理を実現する。

3. 全体構成

本構築法では図 1 に示すように、IC カードとの通信は Authentication Server Daemon（以下 ASD）が占有して行い、各認証系から認証処理を横取りする Authentication Client Program（以下 ACP）が各々、ASD とソケットなどを用いた通信を行い、認証処理を代行する。このようなクライアント－サーバモデルを採ることによって、サーバが IC カードとの通信管理を行なうことができ、通信路が占有されていて、他のアプリケーションが IC カードの内容を読むことができないといったトラブルを回避することができる。また、仮に認証の鍵となる IC カードを変更する際も、このモデルならば ASD のみを書き換えるだけで、新システムへの移行が可能である。

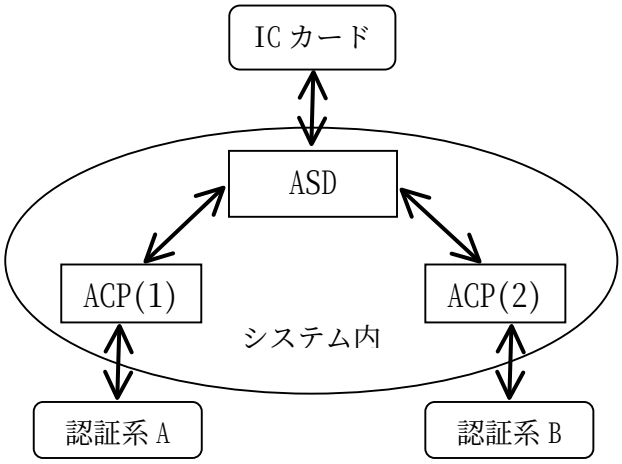

図 1. システムの全体図

4. システムの設計

4.1 使用する IC カード

試作システムでは使用するICカードにFeliCaを採用する。FeliCaはC/C++APIライブラリ、SDK for FeliCaを利用することで、複数の情報を個別に書き込み、読み出しすることができる。

4.2 プログラムの主な機能

ASDは、以下の機能を持つサービスプロセスである。

(1) OS起動時からシステムに常駐し、ACPからの接続待ち状態にある
(2) 接続してくるACP毎に処理スレッドを生成
(3) スレッドでACPから認証系の情報を受信
(4) 認証系の情報からFeliCa内のどこへアクセスするかを検索
(5) FeliCaにアクセスしデータを受信
(6) 結果をACPに送信

一方、各ACPは次の機能を満たすものとする。

(1) 認証時にASDへ接続
(2) 認証系情報をASDへ送信
(3) アカウントとパスワードを受信
(4) 認証系へデータを渡す

A Software Architecture of Authentication Systems using an IC Card.
† Yusuke SAITO, Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology (TUAT)
‡ Yoichi HAGIWARA and Taketsugu SAKURADA, General Information Media Center, TUAT
†† Mitaro NAMIKI, Department of Computer, Information and Communication Sciences, Graduate School of Technology, TUAT

試作システムではWindowsログオンと、HTTP基本認証をACPとして実装する。Winlogonからログオンを横取りするため、独自のGinaDLLを作成する。またWebブラウザから基本認証を横取りするため、ローカルにACP機能を持つプロクシサーバを作成し、インターネットへの接続はこのローカルプロクシを介するように設定する。

4.3 プログラム間のプロトコル

クライアント－サーバ間で交わされるプロトコルにはLDAPv3[5]を採用する。LDAPは9つのオペレーションを持つ。このうち試作システムではBind、Unbind、Searchのみを実装する。

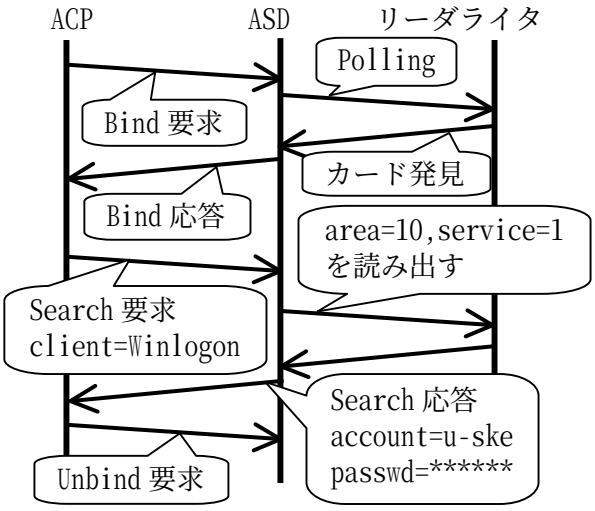

図 2．認証処理の流れ

（1）ACPはLDAPに従いBind要求をASDへと送る
（2）ASDはリーダライタを起動させ、FeliCaを探し、認識した時点でBind応答をACPへ返す
（3）ACPはアプリケーション名（client属性）を検索条件にSearch要求をASDへと送る
（4）ASDは受け取った条件でエントリを検索し、FeliCa内のどこへアクセスするかを決定し、読み出しを行い、結果をSearch応答として返す
（5）ACPがUnbind要求をASDへと送信し、セッションを終了する

表1．オブジェクトクラス（属性値は例）

属性名	属性値	説明
client	Winlogon	アプリケーション名
host		基本認証用。値はURL
area	10	FeliCa内のデータの位置を示す値
service	1	
account	u-ske	認証情報。FeliCaに格納されている
passwd	******	

試作システムでは独自のスキーマを使用し、各エントリは、表1で示されるオブジェクトクラスに従う。

5．試作システムと評価

実装はWindows2000上でVisual C++を用いて行なった。試作システムの構成は図3である。

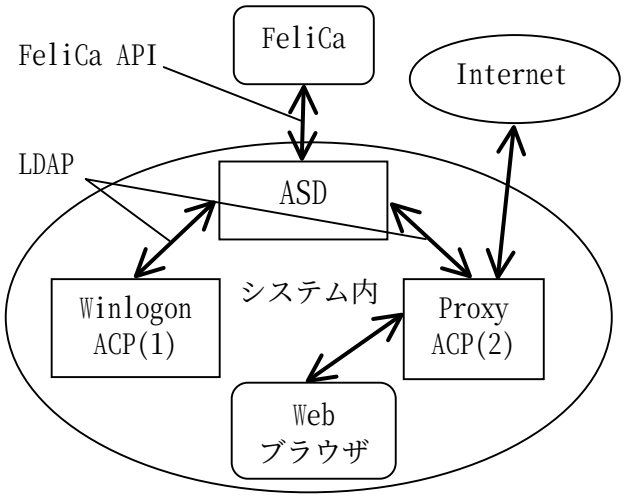

図 3．試作システムの全体図

ACP（1）は常にポーリングを行い、FeliCaの有無を監視しているが、その間も基本認証の認証処理をACP（2）により行なうことに成功した*。コードの規模は、約5000行（うちASDが約3000行）程度となった。

6．おわりに

SmartOn、NSASにおいて、FeliCaでWindowsにログオンした際、他のアプリケーションからFeliCaへのアクセスができなかった問題を、本構築法では、FeliCaとの通信を専用サーバプログラムが中央管理することにより解決した。これにより、複数の認証処理を、単一のICカードを認証の鍵として行なうことに成功した。

参考文献

[1]http://www.sony.co.jp/Products/felica/
[2]飯塚重善，小川克彦，中嶌信弥:セキュアなテレワーク支援システムとシステム利用時の安心感についての考察，情処研報，IS-89，pp.31-38(2004)
[3]http://net.soliton.co.jp/products/soliton/smarton/smarton.html
[4]http://www.nettime.co.jp/
[5]RFC2251，www.ietf.org/rfc/rfc2251.txt

*試作システムではアカウントやパスワードの FeliCa への書き込み機能は設けず、書き込みは別途作成したプログラムで事前に行なった。