卒業論文 2000年度(平成12年度)

セキュリティを考慮した 情報家電機器の共有機構の開発

指導教員
慶應義塾大学環境情報学部
徳田　英幸
村井　純
楠本　博之
中村　修

慶應義塾大学 環境情報学部
杉田 洋介
sugita@sfc.wide.ad.jp

卒業論文要旨 2000年度(平成12年度)

セキュリティを考慮した情報家電機器の共有機構の開発

本論文では，セキュリティを考慮した情報家電機器の共有機構である**SCNA**(Security Control on Networded Appliances)を提案し，その設計と実装，評価について述べる．

近年の情報技術の発達により，家電機器の機能の多様化，多機能化が進んでおり，家電機器の中には，ネットワークへの接続機能の追加や，コンピュータの組み込まれるのも出現している．本論文では，そのような特徴を持つ家電機器を情報家電機器と定義する．

今後，家庭内やオフィス内で，情報家電機器によって構成されるネットワークが普及すると，それらの連携や自動化された動作，遠隔利用などが容易となり，快適な生活環境の構築が実現できると考えられる．しかし，それらを安全に実現する上での基盤技術は，まだ無い．そこで，SCNAのような情報家電機器の共有を支援する機構の普及は重要となる．

そこでSCNAでは，情報家電機器をネットワークを介して複数のユーザが安全に共有し，利用するための機構を提供する．情報家電機器の共有利用に必要となる機能として，本研究では情報家電機器によるユーザの認証機能，情報家電機器の機能へのアクセスの管理，情報家電機器上で提供される資源の管理を実現している．また，SCNAは，情報家電機器上の計算資源への非依存性，ネットワーク上での耐故障性，開発の効率化を考慮して設計されている．

本研究で提案する情報家電機器の共有システムは，認証サーバ，情報家電機器制御機構，ユーザクライアントの3つから構成される．認証サーバは，情報家電機器がユーザアントを認証するための情報を管理するもので，主に暗号化の秘密鍵の情報や，アクセス権限に関する表に関する情報を提供する．そして，情報家電機器制御機構は，認証サーバが提供する情報をもとに機能の提供方針を決定し，情報家電機器上の機能資源の状態や性質に基づいて機能の排他制御を実現する．ユーザクライアントは，本システム上で共有利用をされている情報家電機器を，ユーザが利用する際のユーザインタフェースとして機能する．ユーザクライアントは，認証サーバや情報家電機器と通信し，情報家電機器から認証を受けるために暗号化や復号の処理を行なうが，それらの複雑な処理をユーザが意識しなくても利用できるように設計されている．

本論文では，はじめに情報家電機器全般の説明，家電機器をネットワーク上の分散オブジェクトとして扱う上での関連技術，セキュリティを保つための関連技術について述べる．その後SCNAを構築する際の設計について述べ，その設計に基づいた実装方法について述べる．そして最後にSCNAに対する評価を加え，今後の課題の指摘，全体のまとめを行なう．

慶應義塾大学　環境情報学部
杉田 洋介

Abstract of Bachelor's Thesis

Security Control System for Networked Appliances

This paper presents Security Control on Networked Appliances (SCNA) which supports users to share appliances by using networks, and describes the design, the implementation and the evaluation of it.

Today, some appliances have network interface or embedded computer, and we can construct the networks which consist of the appliances. In this thesis we define these appliances as networked appliances. These network makes us possible to share the appliances each other. When sharing appliances on the network, we should take their security under consideration. Because security control system especially on networked Appliances has not exists, we propose SCNA.

To provide the security system on networked appliances it is supposed that authentication system, access control system and resource Management system is required. SCNA achieves this requirements.

SCNA consists of three software modules, authentication server, Appliance controller and user client. Authentication server manages Encryption keys and access control lists. Appliance controller controls access from users based on information from authentication server and manages their resources. User client supports users to use Appliances by representing cryptography authentication processes.

In this paper, first we explain general technologies for distributed objects, and security control, and then describe the design and implementation Of SCNA. We also evaluated the processing performance and satisfaction of the required functions on SCNA. Finally we show future works, and conclude this paper.

Yousuke Sugita

Faculty of Environmental Information
Keio University

目 次

図目次

表 目 次

第 1 章

序論

1.1　本研究の背景

家電機器は，我々の生活空間に多数存在し，身近なものである．また，近年の情報技術の進歩により，家電機器にコンピュータ機能やネットワーク接続機能が付与されるようになった．本研究の背景としては，以下の4つの要素が挙げられる．

- 家電機器の普及率の高さ
- 家電機器自体の持つ性能の変化
- 家電機器を取り巻く環境の変化
- 家電機器の利用形態の変化

以下に，それぞれに関する説明と，最近の動向について述べる．

家電機器の普及率

家庭やオフィスなどを含めた我々の生活空間の中には，様々な家電機器が多量に存在している．そして，家電機器と呼ばれるものは，通常高い普及率を示しているものが多い[1]．従って，それらの家電機器を利用して有用なシステムを構築することによる我々の生活への影響は大きい．図1.1に，代表的な家電機器，または，注目すべき機能を保持する家電機器の普及率に関するデータを示す．図中で取り上げた家電機器は，特に普及率が高く，パソコンの普及率に比べると倍近い割合で家庭に普及しているものも多く，ほぼ100パーセントの普及率を示すものもある．

家電機器自体の持つ性能の変化

家電機器の性能は年々向上し，本来持っている機能の高度化に加え，それぞれの機器の多機能化が進んでいる．近年では，多機能化された家電機器の例として，多様なネットワー

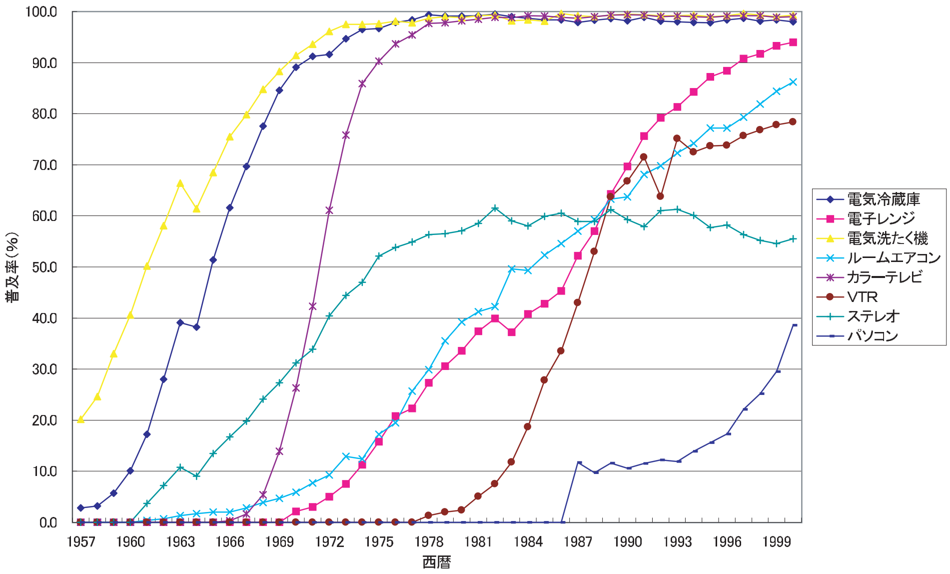


図 1.1: 家電機器の普及率の変化

クインターフェースを備えたものや，Java をはじめとするプログラミング言語を用いて，アプリケーションの開発が可能となっているものが登場している．その結果，家電機器は，コンピュータネットワーク上のノードとして接続可能となり，家電機器とネットワークを利用したアプリケーションの開発が容易となった．

家電機器を取り巻く環境の変化

近年の家電機器を取り巻く環境の変化としては，家庭内 ·オフィス内におけるネットワーク環境の充実，伝送メディアの多様化が挙げられる．具体的な例としては，10BASE-T や 100BASE-TX を中心としたネットワークをはじめ，IEEE802.11[2]，IEEE1394[3]，Bluetooth[4]，IrDA[5] などの多様なネットワーク規格が登場し，家庭内 ·オフィス内への導入率は増している．それと同時に，従来のようにコンピュータを中心にネットワークを形成する以外に，多様なデバイスや，家電機器を相互に接続するようなネットワークが形成されるようになった．一方，家庭やオフィスの外部との接続方法に関しても，アナログ電話回線による接続の他に，ISDN[6]，ADSL[7]，CATV[8]，衛星通信 [9] と多様化が進み，通信速度の面でも充実してきている．

家電機器利用形態の変化

従来，ユーザが家電機器を利用する際には，その家電機器に直接触れたり，専用のリモコンなどを使用して利用していた．つまり，家電機器から見ると固有の操作インターフェースと，それを操作するユーザは，基本的に一対一であった．しかし，家電機器がネットワー

クに接続されることによって，ネットワークに接続された様々なコンピュータ端末や他の家電機器から，操作可能となり，ユーザと家電機器の間で，お互いに多対多の利用形態が成立するようになった．

1.2 問題意識

家電機器がネットワークに接続され，ネットワーク上での制御や共有が可能となった場合，今までの家電機器の利用形態では発生し得なかった問題が発生する．新たに発生すると考えられる問題点としては，以下の2点を指摘できる．

- 部外者による不正使用
- 利用者間における制御の競合

部外者による不正使用

様々な家電機器をネットワーク上に接続し，単にネットワークを経由して個々の家電機器の機能を遠隔制御できるようなシステムを構築した場合，常にその家電機器が本当に適切なユーザによって制御されているということを保証できない．つまり，ネットワーク上で家電機器が複数のユーザによって共有して利用される場合には，利用者として認められていないユーザによる不正使用や，悪意を持ったユーザによる攻撃を受ける可能性がある．

利用者間における制御の競合

ネットワーク上の家電機器の機能が共有されているとき，その機能へのアクセスが発生するタイミングを予測できない．場合によっては，あるひとつの機能に対して複数のユーザによるアクセスが発生し，そのユーザ間で操作の競合が発生する．このような無秩序な利用によって，家電機器の持つ機能自体が破綻してはならない．例えば，対象とする家電機器がディスプレイのように，ひとりのユーザが占有して使うような性質のものであれば，画面を表示するという機能は，物理的に同時に使えない．従って，一方のアクセスを排除し，ディスプレイとしての機能が失われる事態を防ぐ処理が必要である．

1.3 本研究の目的

本研究では，家庭内またはオフィス内に家電機器及びコンピュータによるネットワークを形成することを想定し，家電機器の機能を複数のユーザで安全に共有することを目指す．具体的には，以下の三つの内容を目標とし，ネットワークを介して家電機器を利用していく際に発生する前述の問題の解決策として提供する．

- 家電機器によって構成されるネットワーク環境に適したセキュリティモデルの提案

- 情報家電機器の安全な共有機構を容易に構築するための開発ミドルウェアの提供
- 実際の情報家電機器を利用した応用システムの構築とその評価

1.4 本論文の構成

第2章では，本研究で扱う情報家電機器を定義し，それらをネットワーク上の分散オブジェクトととして扱うモデルについての説明と，それらを扱う研究の動向について述べる．
第3章では，第2章で述べたモデルを用いて情報家電機器の共有利用を実現する際に生じる問題を明らかにし，それらを解決するために必要なセキュリティモデルや技術について説明する．
第4章では，情報家電の共有機構**SCNA(Security Control on Networked Appliances)**を提案し，その設計方針，全体構成，動作手順，各構成要素について述べる．
第5章では，SCNAの実装環境，システムの詳細な実装方法，応用システムの具体的な実装例について述べる．
第6章では，実際にSCNAを利用して構成したシステムを用い，その定量的な測定結果を基にした評価や，本システムによって対応可能な攻撃や対応不可能な攻撃について検証し定性的な評価を行う．
第7章では，本論文の結論を述べ，今後の課題や可能性について言及し，全体を締めくくる．

第 2 章

情報家電機器と分散オブジェクト

本章では，情報家電機器を定義し，その実現手段としてのポストPCについて説明する．そして，情報家電機器を扱う上でのソフトウェア上での手段として，分散オブジェクトの管理に関連する技術であるJiniやCORBAを挙げ，本研究との関係について述べる．また，家電機器産業内での動向として家電機器の相互利用規格であるHAVi，ECHONETを挙げ，本研究との関係について述べる．

2.1 情報家電機器

本研究では，対象とする情報家電機器を，以下のように定義する．

- 小型コンピュータとの接続や，家電機器自身へのコンピュータの組み込みなどの手段により，機能の制御が可能な家電機器
- ネットワーク接続性を持ち，分散オブジェクトとして他のノード上のソフトウェアから制御が可能な家電機器

本節では，情報家電機器の特徴，制限，実現について詳しく説明する．

2.1.1 情報家電機器の特徴

情報家電機器とは，ネットワークに接続可能で，ネットワークを介して通信し合うソフトウェアとしての側面と実際の機能を果たす家電機器としての側面の両方を兼ね備えたものである．情報家電機器の概念図を図2.1に示す．情報家電機器が従来の家電機器と違う点は，家電機器にコンピュータが付加したり，組み込まれたりしている点である．そのコンピュータは，家電機器の機能の制御やネットワーク入出力，通信されるデータの処理を行なう．ネットワークへの接続としてはEthernetやIEEE1394，無線LANなどのインタフェースを備え，多種多様なネットワーク環境を構成する．家電機器上のコンピュータには，Java言語などによる仮想機械(VM, Vertual Machine)などが実装され，VM内には，その家電機器を抽象化したオブジェクトとして情報家電オブジェクトが保持される．そして，家電機器を制御するための命令は，情報家電オブジェクト内のメソッドとして結びつけられる．従って，情報家電オブジェクトをネットワークを通じて外部から参照し，そのメソッドを実行することで，家電機器の遠隔制御が可能となる．また，情報家電オブジェクトをネットワーク環境上の分散オブジェクトとしてモデル化することにより，家電機器を利用した分散システムを容易に開発できる．

2.1.2 情報家電機器の制限

情報家電機器の機能はネットワーク上の分散オブジェクトとして取り扱えるが，そこには情報家電機器固有の制限が発生する．例えば，情報家電機器に付加する計算資源として，一般のPC(Personal Conputer)やWS(Work Station)と同等の性能のものを使用することは，コストやサイズの面で難しい．なぜなら，情報家電機器に付加されるコンピュータに関しては，メモリサイズやストレージの容量，処理速度などの点において貧弱なものになる可能性が高いからである．よって，情報家電機器を扱うシステムでは，情報家電機器が持つコンピュータ部分の性能面での制限を考慮した設計を意識しなくてはならない．

また，もう一つ情報家電機器の持つ本質的な問題点としては，各家電機器の実体は実世界に一つしか存在しないという点である．情報家電機器を分散オブジェクトとして，むやみに複製して参照できないという点がある．つまり，実際の家電機器をオブジェクト化して

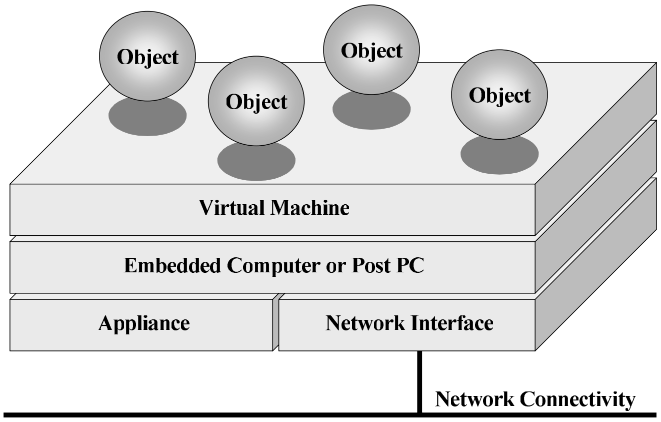


図 2.1: 情報家電機器

いる場合，そのオブジェクトへの参照は物理的な家電機器の動作と結び付くため，仮想空間上では，同時に複数の処理の命令を出せたとしても，実際の家電機器の振る舞いが，機能として破綻してしまっている可能性がある．

2.1.3 情報家電機器の実現

本研究では，情報家電機器にネットワーク接続機能や演算機能を提供するコンピュータをポストPCと定義する．ポストPCは，ネットワーク上の分散オブジェクトとして情報家電機器を存在させ，情報家電機器を制御するソフトウェアの実行環境を提供する．そして，ポストPC上で動作する環境としては，Sun Microsystems社のJava Virtual Machineの仕様である，J2ME[10]，KVM[11]，Embedded Java[12]，Java Card VM[13]などが挙げられる．これらの環境は，マルチプラットフォームへの対応や，オブジェクト指向プログラミングなどの特徴を持ち，情報家電機器によるシステム開発環境として注目される．

従来のコンピュータとポストPCとしてのコンピュータとの違いを，表2.1に示す．従来のPCは，ユーザの実行したい作業をコンピュータで処理したいときに利用するものであり，キーボードやマウスで操作し，ディスプレイやプリンタなどで出力結果を得るというような用途で使われることが多い．つまり，ユーザがコンピュータを意識的に利用するという側面が強いので，一般の家電機器のように生活空間内に遍在し，その入出力装置や演算装置を純粋に利用するコンピュータとしては形状の面で適しておらず，耐久性にも乏しい．

一方，ポストPCは，日常生活でその存在をユーザに意識させないくらい小型なものになる．

また，ハードウエアとしても，ある程度の強度を持ったコンピュータとなっていくこと

表 2.1: 従来型のPCとポストPCの特徴比較

項　　目	従来型のPC	ポストPC
形状	大型	小型
機能	多機能	特化された機能
ユーザインターフェース	充実	特に必要としない
使用感	操作を意識	無意識
コスト	高い	低い

で，生活空間内に遍在することに適応する．具体的な例としては，Duonus[14]，TINI[15]といった，小型のハードウェアが挙げられる．将来的には情報家電機器自体に，そのようなコンピュータ部分が組み込まれ，その存在をユーザに意識させない一方で，コンピュータ資源が至るところに存在しているという環境が構築されると考えられる．つまり，情報家電機器は，従来の家電機器のように手軽に利用できる一方で，ネットワーク接続機能を用いた遠隔制御や共有利用といったサービスが利用可能になる．

2.1.4 情報家電機器によるネットワークの特徴

情報家電機器によるネットワーク(以下，情報家電ネットワーク)は，利用者の多様性や，接続される機器の多様性，構成要素の動的な変更などの特徴を持っている．

利用者の多様性

情報家電ネットワークは，一般家庭やオフィスをはじめ，様々な生活空間内で構築されるため，想定される利用者としては，年齢層は高齢者から幼年者まで幅広く，ネットワーク設定やコンピュータシステムに関する知識に関しても初心者から熟練者まで様々である．

接続される機器の多様性

それぞれの情報家電機器は，テレビやビデオであったり，冷蔵庫，洗濯機，電話，照明など，多種多様であり，それらが提供する機能も多岐にわたる．また，それぞれの情報家電機器は，異なるベンダから提供されており，様々な規格や操作方法が混在する．

多様な伝送メディア

1.1節で述べたように，情報家電機器を相互に接続する手段には多様な伝送メディアが存在する．従って，情報家電機器によって構成されるシステム上では，それらが混在することを意識する必要がある．

構成要素の動的な変化

情報家電機器は，頻繁にネットワークから切断されたり，接続地点を変更されたりする可能性が高い．そのため，情報家電ネットワーク上で構築されるシステムでは，ネットワーク構成の動的更新を意識する必要がある．

2.2 分散オブジェクトの管理と問題点

2.1節で述べたように，情報家電機器は，ネットワーク上の分散オブジェクトとしての側面を持っている．そこで，本節では，分散オブジェクトを扱うための注目すべき技術として，JiniとCORBAを挙げ，その特徴について述べる．これらの技術は，情報家電機器をネットワーク上の分散オブジェクトとして扱う際に有用なものである．

しかし，実際に情報家電機器を複数ユーザで共有する際の処理や，情報家電機器の性質を考慮した制御に関する統一的な手法については，特に規定されていない．従って，その部分を補い，共通の開発ミドルウェアを提供することが本研究の目的の一つである．

2.2.1 Jini

Jini[16]では，ネットワーク上の分散オブジェクトをサービスとし，ネットワーク上でのサービスの提供やサービスの利用を支援する機構を提供している．Jiniの概要を図2.2に示す．Jiniでは，Lookupサービスと呼ばれる特別なサービスによって，他のサービスの検索，利用を可能にしている．Lookupサービスには，各サービスに関する情報が登録されており，各サービスのスタブオブジェクトを保持する．サービスの登録者や利用者は，Discoveryプロトコルによって Lookupサービスを発見し，提供するサービスの登録や，使用したいサービスの検索に利用する．また，登録したサービスを一定時間更新しないと廃棄するというLeaseプロトコルにより，サービスの動的な変化に対応している．従って，家電機器をシステム内に追加・削除する際の設定の自動化(プラグアンドプレイ)によるシステムの再構成をサポートしやすい．

しかし，Jini自体に情報家電機器によるサービスの共有利用に特化した仕組みは用意されておらず，ユーザ認証や機器のリソース管理などを提供するためのミドルウェアとしての機能は，提供されていない．

2.2.2 CORBA

CORBA[17]は，異なる言語で開発されたソフトウェアを分散オブジェクトとして透過的に扱うために設計された規格である．CORBAの概要を図2.3に示す．オブジェクトの提供者とそれを利用したいクライアントが分散システム上に存在する時，ORB(Object Request Broker)がそれらの仲介をするモデルになっている．開発言語による差異を吸収するために，IDL(Interface Definition Language)と呼ばれる言語でインタフェースの記述を行い，定義された機能をそれぞれの言語が実装していくことで，開発言語非依存な分散オブジェクト

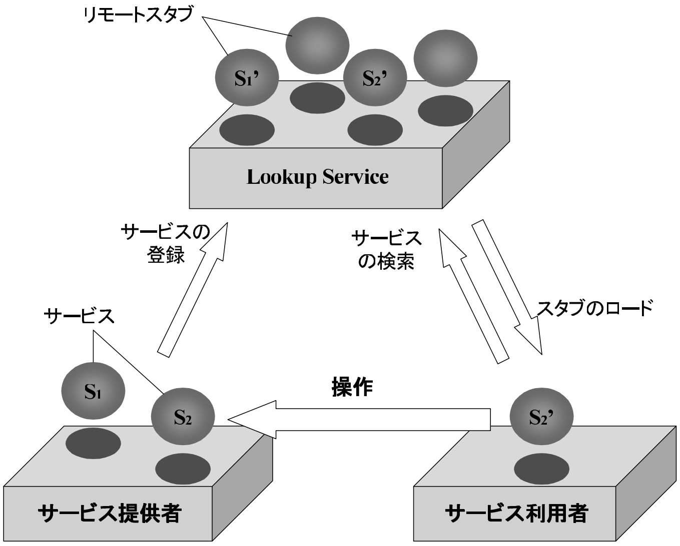


図 2.2: Jini の概要図

管理を実現している．家電機器の場合は，開発したメーカーやそれぞれの機器の特徴によって機能の提供方法が違う可能性があるので，それらを透過的に扱う手段として CORBA を用いたシステムは有効である．

しかし，CORBA 自体は，分散オブジェクトを扱う上での規格を定めているだけで，実際に分散オブジェクトを用意し，システムを構築する際には，それぞれの実装方法に依存する部分が多い．そこで，CORBA に対応して情報家電機器によるシステムを構築していく際にも，情報家電機器の共有利用に特化してシステムの開発を支援するミドルウェアを提供すれば，効率的な開発が可能となる．

2.3 家電機器の相互接続のための規格と問題点

特定分野の家電機器メーカ間では，それぞれの提供する家電機器間の相互接続を実現する統一的な規格を制定する動きが盛んになっている．以下に代表例として HAVi と ECHONET を挙げ，その特徴と問題点について述べる．

2.3.1 HAVi

HAVi[18] は，主に AV 家電機器によるホームネットワークをサポートする規格であり，異なるベンダから提供される機器間での相互接続性について規定している．HAVi の概要を図 2.4に示す．伝送メディアとして IEEE1394 を全面的に利用しているという点と，仕様全体が AV 家電機器に特化されているという点から，汎用的な情報家電機器ネットワークの規格というよりは，AV 機器制御系として，情報家電機器ネットワークの一部として位置づ

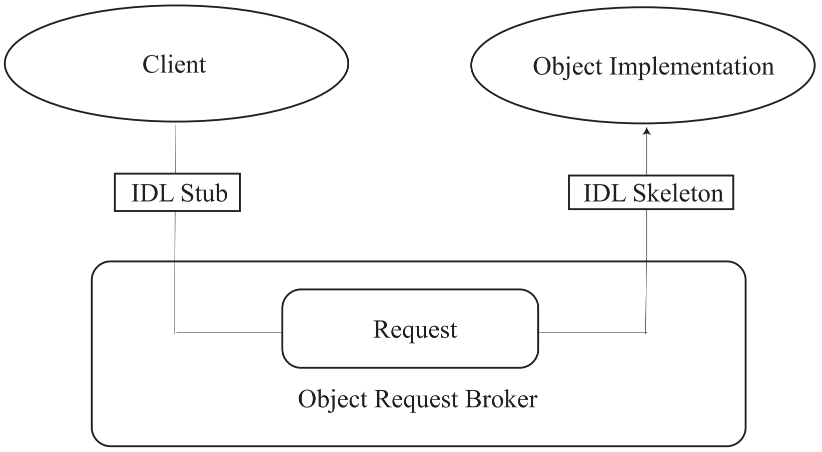


図 2.3: CORBA の概要図

けられる．また，各機器の操作インタフェースを異なる機器上に移送することで，シームレスに機器を利用できる．

しかし，HAVi によって構築されるシステムは，ベンダ非依存に再構成可能な AV コンポを提供するような仕組みとなり，ユーザと AV システムの関係としては，従来型の家電機器と同様に，一対一の対応となる．従って，複数のユーザで家電機器を共有利用するといった側面は特に考慮されていない．そこで，ユーザの認証やアクセス権限を考慮し，家電機器の安全な共有利用を実現するための機構を補う必要がある．

2.3.2 ECHONET

ECHONET[19] は，家庭内に存在する幅広い種類の家電機器を想定し，それらの相互運用性を確立するような柔軟性を持った規格となっている．ECHONET の概要を，図2.5に示す．ネットワークを構築する上では，電力線や無線，赤外線などの多様な伝送メディアの混在をサポートしている．アプリケーション構築の際には，必要な家電機器の制御機能を集めたコントローラと呼ばれるモジュールを構成し，アプリケーションによる家電機器の制御はそのコントローラを通じて行われる．コントローラとの通信の際には，ECHONET ゲートウェイと呼ばれるモジュールが間に置かれ，通信内容のフィルタリングや，機器へのアクセス制御が可能である．

しかし，ECHONET ゲートウェイの仕様では，ユーザのなりすましを防ぐ機構までを具体的に規定していない．従って，家電電器の安全な共有利用を実現する際には，暗号化を用いた認証機構などを補う必要がある．

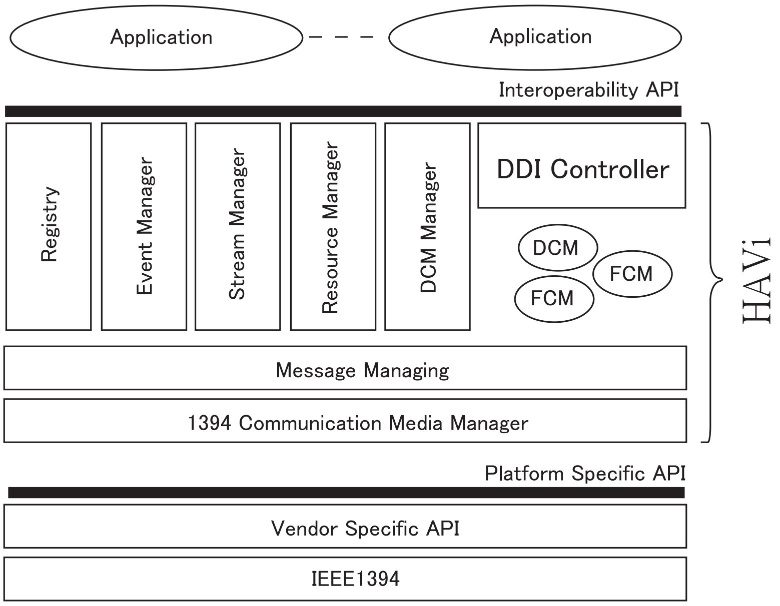


図 2.4: HAViの概要図

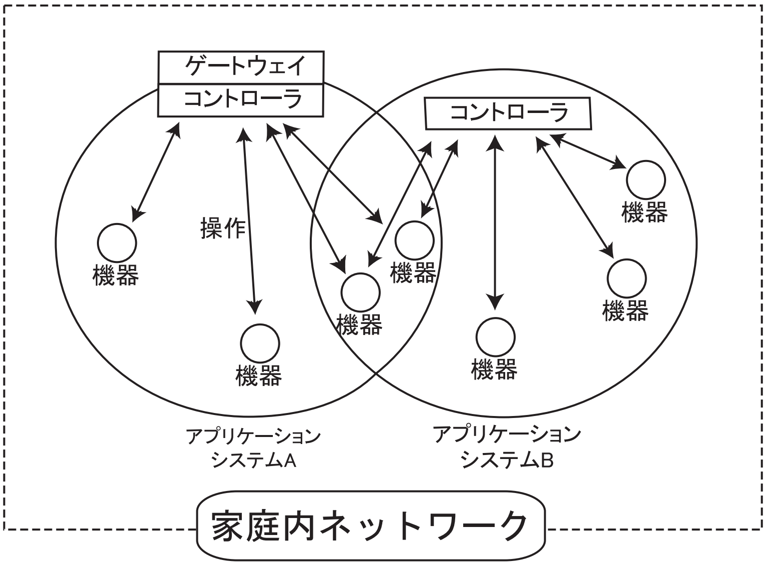


図 2.5: ECHONETの概要図

2.4　本章のまとめ

本章では，情報家電機器を定義し，家電機器の機能が分散オブジェクトとしてソフトウェア上で扱われることについて述べ，情報家電機器環境を実現するポストPCについて説明

した．

そして，分散オブジェクトを扱う際の関連技術や，情報家電機器を対象とした既存の統一規格を参照し，その問題点を指摘した．それらの技術や規格はそれぞれ，情報家電機器によるシステムの構築する上で有用なものだが，情報家電機器の安全な共有利用を考慮すると，セキュリティに関する概念を付け加える必要がある．

次章では，情報家電機器を共有利用を実現する際に導入すべきセキュリティの概念について述べる．

第 3 章

情報家電機器の共有利用とセキュリティ

情報家電機器は，ネットワークに接続されることによって，その機能を複数のユーザによって共有利用させることを可能としている．しかし，その機器の安全な共有利用を考えた場合，使用権の正当性や動作の一貫性を保証しなくてはならない．そのためには実際の情報家電制御システム上に，セキュリティに関する新たな概念を付け加える必要がある．本章では，情報家電機器の共有利用に必要となる概念として，ユーザの認証，アクセスの管理，資源の管理の3点について言及し，関連するセキュリティ技術について述べる．

表 3.1: コンピュータシステムによるユーザ個人の認証方法の比較

項　目	記憶による確認	携帯物による確認	身体的特徴による確認
実装	容易	比較的容易	困難
コスト	低い	やや高い	高い
ユーザへの負担	大きい	小さい	小さい
家電システムとの相性	悪い	良い	良い
具体例	パスワード入力	Java Ring, Java Card	指紋，声紋

3.1　ユーザ本人の認証

情報家電機器を利用するシステム上で，ユーザの利用権限を定めるには，システムがユーザを識別し，その正当性を確認できなくてはならない．

コンピュータシステムが，ユーザが本物であるかを確認する手段は一般的に，

1. ユーザの記憶による確認方法
2. ユーザの携帯物による確認方法
3. ユーザの身体的特徴(バイオメトリクス)による確認方法

の 3 つに分類される．1 の具体例としては，ユーザ名とパスワードを入力してコンピュータ端末にログインするような場合が挙げられる．2 では，JavaCard[13] や JavaRing[20] のような小型デバイスを専用のリーダに挿入することによってユーザを識別する方法が挙げられ，3 では，指紋や声紋，網膜のパターンから人物を特定する方法が挙げられる．コンピュータシステムがユーザを認証するための方法と特徴をまとめたものを表3.1に示す．

パスワードの入力による認証は実装上最も容易であるが，家庭やオフィス内で情報家電機器を利用するためのシステムには適した方法と言えない．なぜなら，情報家電ネットワークによるシステムを構築する際には，ユーザにコンピュータを利用することをなるべく意識させないことが望ましいのに対し，煩わしい入力作業をユーザに強いることになるからである．よって，情報家電機器によるシステムの構築の際には，携帯物やバイオメトリクスを利用した認証方法が適していると考えられるが，本研究では携帯物を利用した認証方法の利用を想定する．なぜならば，バイオメトリクスは導入コストが高く，また4.5.2節で後述する秘密鍵の管理を行なう場合に適していないからである．

3.2　分散システムの認証

分散システムとして，ネットワーク上に接続された情報家電機器やコンピュータ同士が互いに通信する場合，交換されるメッセージが誰からのものなのかを確認する手段が必要

である．これは暗号を利用した分散システム間における認証で実現できる．

3.2.1 暗号化による認証

暗号化による分散システム間の認証方式としては，対称鍵を利用した秘密鍵暗号系と非対称鍵を利用した公開鍵暗号系との2つに大きく分類される．以下でそれぞれの認証方法を比較する．

秘密鍵暗号系による認証

秘密鍵暗号系を利用する場合，AとBの2台のコンピュータ(またはコンピュータ上のユーザ)が存在するとしたら，2者はお互いしか知り得ない同一の秘密鍵K_{AB}を共有する．その鍵で暗号化されたデータは，その鍵によってのみ復号できる．そして，AからBに認証の要求があったとき，Bは適当な乱数Rを選び出し，Aに送る．AはK_{AB}でRを暗号化した$K_{AB}\{\mathrm{R}\}$をBに送り返す．それを受け取ったBは，その$K_{AB}\{\mathrm{R}\}$をK_{AB}で復号しRを得て，もとのRと比較し，同一であることを確かめることで，相手が本当にAであることを確認する．その過程を図3.1に示す．秘密鍵による暗号化アルゴリズムとしては，DES[21]やIDEA[22]などが挙げられる．

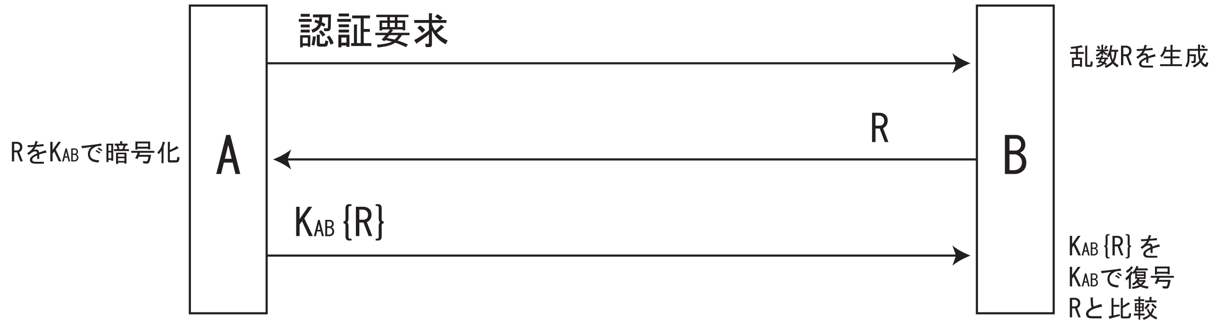


図 3.1: 秘密鍵暗号系による認証の流れ

公開鍵暗号系による認証の場合

公開鍵暗号系を利用する場合，1つのコンピュータ(またはコンピュータ上のユーザ)に関して，公開鍵と非公開鍵のペアを用意する．公開鍵は，データの暗号化のための鍵で，誰もが取得できる．一方，非公開鍵は公開鍵によって暗号化されたデータの復号をする鍵で，その持ち主以外には知られないようにする．例えば，AとBの2台のコンピュータが存在し，AがBに認証を求めた場合，Bは適当な乱数RをAの公開鍵E_Aで暗号化した$E_A\{\mathrm{R}\}$をAへ送る．Aは受け取った$E_A\{\mathrm{R}\}$をAの非公開鍵D_Aで復号してR'を得て，Bへ送り返

す．Bは受け取ったR'をもとのRを比較し，同一であることを確かめることで，相手が本当にAであることを確認する．公開鍵を利用した認証の流れを，図3.2に示す．公開鍵による暗号化アルゴリズムとしては，RSA[23]やDSA[24]などが挙げられる．

図 3.2: 公開鍵暗号系による認証の流れ

一般に，公開鍵暗号系アルゴリズムは，秘密鍵暗号系アルゴリズムに比べて，処理が複雑で，時間がかかる傾向にある．公開鍵を利用したDSAによる署名・検証と，秘密鍵を利用するDESによる暗号化・復号では，表3.2に示す環境のもとで，64Byte長の同一データに対する処理時間を測定した結果を表3.3に示す．

表 3.2: 測定環境

項目	環境
CPU	AMD K6-III 450MHz
メモリ	128MB
OS	Windows 98 Second Edition
JDKバージョン	Sun JDK 1.2.2_006 win
暗号化ライブラリ	Sun JCE 1.2.1

このように，秘密鍵暗号アルゴリズムの方が公開鍵暗号アルゴリズムより，比較的高速である．今回の測定環境では，一般的なPCを利用しているため，処理の遅さを感じる程の差は見られないが，実際に情報家電機器上のコンピュータでの低速な実行環境を想定すると，この差は顕著に現れてくると考えられる．従って，情報家電上で暗号化アルゴリズムを使用する際には，計算量が少ないものの方が良いので，秘密鍵による暗号化アルゴリ

表 3.3: DES と DSA による処理時間の違い

アルゴリズム	鍵方式	暗号化に要した時間	復号に要した時間
DES	秘密鍵	4ms	1ms
アルゴリズム	**鍵方式**	**署名に要した時間**	**検証に要した時間**
DSA	公開鍵	25ms	40ms

ズムを利用した方が適している．また，署名を利用してユーザを認証する場合，情報家電機器上で行われるべき処理である検証に特に時間を要している．従って，情報家電機器上で処理される暗号化方式としては，秘密鍵暗号アルゴリズムの方が負荷が少ないため適していると考えられる．

3.2.2 KDC による認証システム

秘密鍵による暗号化を利用する場合，通信相手の数だけその相手との秘密鍵を共有する必要があるので，ノードの数が増えると大量の秘密鍵を管理しなくてはならない．特に，ストレージ容量に乏しい情報家電機器上で大量の鍵を記憶する機能を期待するのは望ましくない．解決方法としては，KDC(Key Distribution Center) による鍵配布システムの利用が考えられる．KDC による処理の流れを図3.3に示す．

KDC は，管理するネットワーク上の全てのノードそれぞれと親鍵と呼ばれる秘密鍵を共有している．そして，A というノードが B というノードと通信しようとする場合を考えると，KDC は，A と B が通信するためのセッション鍵 K_{AB} を生成し，A の親鍵 K_A で K_{AB} を暗号化した $K_A\{K_{AB}\}$ と，B の親鍵 K_B で K_{AB} を暗号化した $K_B\{K_{AB}\}$ (チケットと呼ばれる) を A に送る．それらを受けとった A は，$K_A\{K_{AB}\}$ を復号できるので，K_{AB} を得られる．そして B は，通信の際に A からチケット $K_B\{K_{AB}\}$ を受けとると，それを復号できるので，セッション鍵 K_{AB} を得られる．その後，A と B との間での通信での暗号化には，セッション鍵 K_{AB} が使われる．つまり，K_{AB} を利用して相手を認証すると，その相手が KDC による認証を受けていることが保証されるので，A と B は KDC の信頼の基に安全な通信ができる．

3.3 アクセスの管理

情報家電機器をネットワーク上で共有して利用する場合，不正なユーザによる利用を防ぐために，それぞれの情報家電機器に対して，利用するユーザの権限を設定しなくてはならない．本節では，ユーザによる機器へのアクセス管理について述べる．

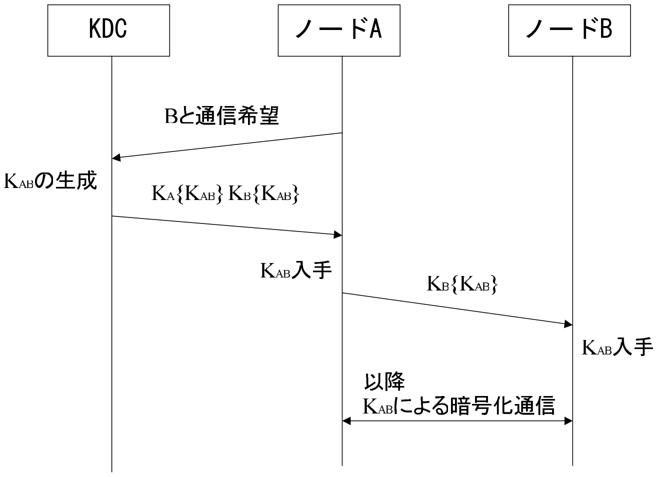


図 3.3: KDC による認証の流れ

3.3.1 アクセスコントロールリスト

一般にコンピュータ資源に対するアクセス権限を管理するためには，アクセスコントロールリスト(ACL, Access Control List)を利用した方法が用いられる．ACL は，図3.4のように，各ユーザと各資源を掛け合わせた表として保持され，あるユーザが持つある資源へのアクセス権限を得る時には，表中でその交差した部分に記述された内容を参照する．

3.3.2 ACL を管理する際の問題点

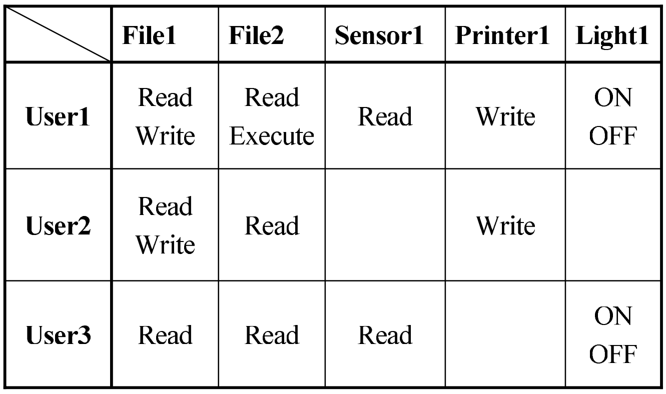
前節で述べた通り，ACL を管理する際の問題点としては，その大きさが増大していく点である．特に，情報家電機器上で ACL を管理することを考えると，各情報家電機器上に ACL を保持する機能を用意する必要があるが，ストレージは不足する可能性がある．

また，ACL を用いてアクセス制御を行なうためには，アクセスコントロールを行なうコンピュータ上で，ユーザの管理が行なわれる必要がある．そして，ユーザの管理をするためには，ユーザに関する情報を記した表を管理する必要が生じる．それも，ACL と同じようにユーザの数が増えるに従って，その大きさが増大するという性質を持つ．

このように，情報家電機器上で，従来のコンピュータのようなアクセス管理を行なうことを想定すると，ACL やユーザの情報を含む表を管理する際に，その大きさが増大することを考慮しなくてはならない．

3.4 資源の管理

情報家電機器をネットワーク上で共有して利用する場合，その機器上に存在する資源の性質や，使用状態を考慮した制御が必要となる．本節では，そのような資源の管理方法に



	File1	**File2**	**Sensor1**	**Printer1**	**Light1**
User1	Read Write	Read Execute	Read	Write	ON OFF
User2	Read Write	Read		Write	
User3	Read	Read	Read		ON OFF

図 3.4: ACLの例

ついて述べる．

3.4.1 排他制御

1.2節で述べたように，情報家電機器上の機能の中には，複数のユーザが同時に利用できないため，利用者間で操作の競合を起こす資源が存在する．情報家電機器の共有システム上では，そのような競合を回避する排他制御機構を必要とする．

ソフトウェア上でコンピュータ資源の排他制御を実現する方法としては，ロック変数を用いた方法が挙げられる．排他制御が必要な資源に対してロック変数を用意し，0で初期化しておく．その後，その資源を個々のユーザが利用する時には，ロック変数が1ならば0になるまで待機し，ロック変数が0であれば，その値を1に設定して資源を利用し始める．また，資源の利用を終了する時には，1に設定していたロック変数を0に戻す．この場合，ロック変数の1という値は，現在資源を利用しているユーザが存在しているため，利用できないことを示す．逆に0という値では，競合する利用者が存在せず，利用できることを示す．

3.4.2 リース

ロック変数による排他制御を行なう際，ロックをかける作業とロックをはずす作業が必ず対になって実行されないと，ロックをかけたままの状態が続いてしまう可能性がある．分散システムでこのような作業を行なった場合，ネットワーク上の事故やロックをかけたシステムでの障害などの可能性が高いので，それらに対処する機構が必要となる．

ロックをかけたままの状態の回避方法としては，ロックをかけて資源を占有できる時間をリース時間として設定し，その時間が経過すると自動的にロックが解除されるようにする方法がある．そして，リース時間を越えてロックを持続する場合は，ロックをかける側のシステムが定期的にリース時間の更新を要求する．これによって，ロックをかける側のシステムとかけられる側のシステムの間の通信が可能な限り，ロック変数による排他制御が行なわれ，途中で通信が途絶えてしまった場合でも，一定時間経過後にロックが解除されることを保証できる．

3.5 関連技術

本節では，ネットワークを介した分散システムで暗号を利用した認証を実現するための関連技術として，Kerberos[25] と IPsec[26] について述べる．そして，情報家電ネットワークを想定した場合の問題点を指摘する．

3.5.1 Kerberos

Kerberos は，3.2.2節で述べた KDC による認証システムに加え，チケット交付チケット (TGT, Ticket-Granting Ticket) を認証に利用するシステムである．TGT とは，ノード A が KDC と通信する際にもセッション鍵を利用した通信をするために使われるもので，KDC から発行されるチケットのことである．3.2.2節の例で考えると，ノード A がノード B との通信を希望することを KDC に通知した後，KDC からノード A に渡されるものである．

しかし，Kerberos の中で利用される TGT は，KDC 以外のノードどうしでの通信をする際のプロトコルと，各ノードが KDC と通信する際のプロトコルを同じにするために採用されているが，ノード A とノード B の間での認証に関しては，安全上も機能上も特に利点がない．なぜなら，KDC とノード A は，もともとノード A の親鍵を共有しているので，セッション鍵を利用した認証をしなくても，ノード A の親鍵を利用した認証ができるからである．そこで，情報家電機器を利用するシステムを構築する上で，Kerberos のように TGT を利用する機能は冗長となるので，採用する必要はない．

3.5.2 IPsec

IPsec は，IP パケットにセキュリティに関する情報を追加することにより，ネットワーク層でのセキュリティ管理機能を提供している．IPsec では，セキュリティシステム構成の柔

軟性を提供し，暗号アルゴリズム、認証方式、鍵交換アルゴリズム、証明書発行アルゴリズム等をユーザのポリシに従い選択して使用できる。

しかし，IPsec が提供するセキュリティ機能は，ネットワーク層で IP を利用したネットワークを介したアプリケーションでしか利用することができない．つまり，必ずしも IP による通信が想定されていない情報家電ネットワーク上では，より上層のプロトコル上で，セキュリティ機能を提供する必要がある．

3.6　本章のまとめ

本章では，情報家電機器の共有利用に必要となる概念として，ユーザの認証，アクセスの管理，資源の管理の 3 点を取りあげた．ユーザの認証に関しては，暗号化を利用した認証システム，アクセスの管理に関しては，ACL によるアクセス管理方法，資源の管理に関しては，ロック変数による資源の管理方法について述べた．また，分散システム間での認証システムの関連技術として，Kerberos と IPsec について説明し，情報家電機器の共有システム上で適用する上での問題点について述べた．

次章では，これらの概念を考慮し，情報家電機器の共有システムを構築する際の設計について述べる．

第 4 章

情報家電共有機構SCNAの設計

本章では，情報家電機器共有機構SCNAに関する設計について説明する．前半ではSCNAに必要な機能として，認証機能，アクセス管理機能，資源の管理機能が挙げ，その設計方針について述べる．後半ではSCNAに含まれる各構成要素を認証サーバ，情報家電機器制御機構，ユーザクライアントの3者に分け，個々にそれらの詳細な設計について説明する．

4.1 設計方針

家庭やオフィス内に情報家電ネットワークを構築することにより，様々な情報家電機器を複数のユーザ間で共有して利用できる．

しかし，その際には，ネットワーク経由での個々の情報家電機器を操作するための機構を提供するだけでは不十分である．例えば，情報家電機器を適切に動作させるためには，個々のユーザの情報家電機器へのアクセス管理や，情報家電機器のリソース管理が必要となる．そこで本研究では，情報家電機器の共有利用を支援するための機構**SCNA(Security Control on Networked Appliances)**を提案する．以下に，認証，アクセス管理，資源の管理，開発者の負荷軽減の問題意識について，SCNAが想定する設計方針を述べる．

4.1.1 情報家電機器の共有に必要な認証機能

情報家電機器の共有システム上のノード間での認証を提供する段階としては，以下のレベルが想定できる．

レベル0 認証なし

レベル1 情報家電機器からユーザクライアントの片方向の認証

レベル2 情報家電機器とユーザクライアントの双方向の認証

本研究では，情報家電機器をネットワークを経由して共有利用する際に，不正なユーザによる情報家電機器の利用を防いだり，違ったアクセス権限をもったユーザを区別することが，目的のひとつとなっている．従って，レベル1までの認証機構が構築できれば十分であり，レベル2までの認証機能を要求することは重要でない．なぜなら，情報家電機器の共有利用を考えたとき，情報家電機器がユーザを認証して，ユーザに応じた動作を決定することが必要であるが，ユーザが悪意のある情報家電機器を危惧することは，オフィスや家庭内ネットワーク上では考えにくいからである．例えば，ユーザがクライアントとなる端末から部屋の照明に向けてをONの命令を出した場合，照明になりすましてその命令を受け，勝手にONになるテレビが，自宅やオフィスに存在するとは考えにくい．

4.1.2 情報家電機器の計算負荷を考慮したアクセス管理

ユーザを認証した後，情報家電機器上では，そのユーザが持っているアクセス権限に基づいて，そのユーザから要求のあった動作の可否を決定しなくてはならない．そのためには，情報家電機器上で各ユーザに対するACLを保持し，要求のあったユーザに対応するエントリをACLから検索しなくてはならない．従って，情報家電機器がACLを保持し，検索，追加，変更するための機能を持つことが必要となる．しかし，この方法では，情報家電機器上のコンピュータで実行される処理の負荷を増加させてしまい，ユーザの追加とともに増大していくACLの情報を保持するための装置を情報家電機器上に用意する必要が生

じる．つまり，情報家電機器上に必要となる計算資源の条件が厳しくなるため，その条件を満たせない情報家電機器に関しては，共有システムに適応不可能となる．

そこで本システムでは，個々の情報家電器上のコンピュータの性能の制限をできるだけ受けないように，ユーザの認証とアクセス管理を実現する機構を提案する．つまり，ACL を保持したり，ACL に対して操作したりする役割を情報家電機器とは別のノードで行うようにシステムを構築する．情報家電機器が外部から操作の要求を受けたとき，情報家電機器はアクセスしたユーザが誰であるのかを必ずしも知る必要はなく，あるユーザから受けた操作要求に対して，適切な動作を実行するための情報さえ得られればよい．具体的には，あるユーザからの要求と同時に，そのユーザに対応する ACL のエントリが正しく情報家電機器側に送られれば，その ACL エントリをもとに情報家電機器はアクセスコントロールポリシを決定できる．

その際に問題となるのは，情報家電機器が受け取った ACL エントリの内容の正当性である．本システムでは，ACL エントリの内容の完全性を保証する機構を提供し，情報家電機器が適切なアクセスコントロールポリシの下に運用できることを目的とする．

4.1.3 情報家電機器の利用形態に適した資源の管理

情報家電機器をネットワーク上で共有して利用する場合，利用者間での操作に競合が発生する場合がある．また，情報家電機器上で複数の機能が利用可能な場合では，同一の機能だけでなく，違った機能の間でも競合による問題が発生する．例えば，ビデオデッキが仮想家電機器として抽象化されていて，その中では，再生，停止，早送り，巻き戻し，といった機能が提供されているとする．ここで，あるユーザがビデオデッキのテープの内容を再生している間に，他のユーザによるテープの再生のみを排除しても，停止操作が実行できると，再生機能は停止機能の競合を受け，ビデオを見ていたユーザは再生機能の利用中に妨害を受けてしまう．

以上のような競合によって情報家電機器が正常に動作しなくなるような操作が実行されるのを防ぐには，情報家電機器の機能を排他的に利用するための機構が必要である．本システムでは，家電機器が提供する機能の特徴を分析し，家電機器制御に適した排他制御機構を提供する．

4.1.4 開発者の負担の軽減

本研究の目的のひとつは，様々な情報家電機器を安全に共有するためのシステムを容易に構築可能とするミドルウェアを提供することにある．まずは，情報家電機器上やその他コンピュータ上で行われる通信内容や処理内容を定義し，各ノード上での処理をモジュール化し，それらのモジュール同士でのインターフェースを提供する．そして，そのインターフェースに基づいた特定の実装を提供することで，開発者の負担を軽減できる．また，本研究が提供する各インターフェースを利用して開発を進めることで，本研究が提供する特定の実装以外の実装への移行が容易となる．例えば，他の暗号アルゴリズムや，新たな情報家電機器を導入する際には，それらに対応したモジュールを交換するだけで良い．

4.2 SCNA の概要

SCNA は，ネットワークを介して安全に情報家電機器を共有利用することを支援するシステムである．本節では，本システムが想定する情報家電機器の利用形態と，システム内でのソフトウェア構成，使用するハードウェア構成，全体の動作手順について述べる．

4.2.1 SCNA の利用範囲

SCNA では，すべての情報家電機器への操作は，情報家電オブジェクトを経由して実行されることを前提とする．なぜなら，情報家電ネットワーク上でのユーザのアクセス管理や，機器上の資源の管理は，情報家電オブジェクトがポストPC上またはネットワーク上に存在する他のプログラムと通信することで実現されているからである．情報家電機器が，従来の家電機器として保持するボタンやスイッチなどの，本システムが対応しないインターフェースから操作を受けた場合，コンピュータ上の処理による情報家電機器の共有環境を実現できない．そこで，SCNA では，情報家電機器への操作は，コンピュータ上の情報家電オブジェクトへのインタフェースを介してのみ行われるものとする．

4.2.2 ソフトウエア構成

SCNA は，情報家電機器，ユーザクライアント，認証サーバの3種類のノードから構成される．SCNA における3者の関係を，図4.1に示す．情報家電機器認証サーバは，特定のネットワークセグメント単位で設けられ，そのセグメント内のユーザクライアント・情報家電機器間で行われる通信のセキュリティを提供する．認証サーバは，認証に必要な情報を提供するため，ユーザの認証のための鍵のデータベースと，ACL のデータベースを集中管理している．これは，個々の情報家電機器に対して，それらのデータを保持し，認証やアクセスを制御することの負荷を軽減し，全体としての管理を簡便にすることを考慮したものである．

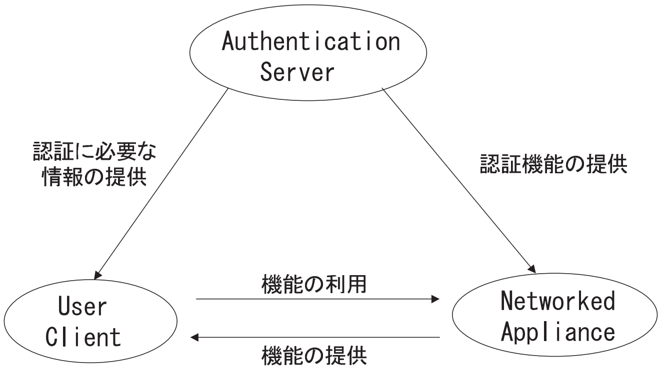


図 4.1: 認証サーバ，ユーザクライアント，情報家電機器の三者の関係

4.2.3　ハードウェア構成

SCNAによって構成するシステムでは，図4.2に示すハードウェア環境を想定している．認証サーバとしては，比較的大型の計算機で，処理性能に不足しないようなものが適している．ユーザクライアントとしては，一般的なPCやWS，さらには携帯端末としてPDAなどのハードウェアを想定し，情報家電共有システムに対するユーザインタフェースとして利用する．実際の機能を提供する情報家電機器としては，Java VMが動作するポストPCが接続されているか，組み込まれているテレビやVCR(Video Cassette Recorder)を想定する．

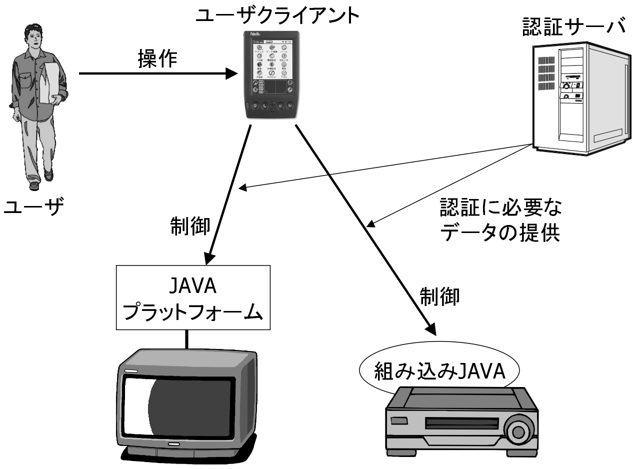


図 4.2: 想定するハードウエア構成

4.2.4　システム全体の動作手順

本システムを利用して，ユーザクライアントからネットワーク上の情報家電機器を利用するまでの処理の流れは以下の通りである．また，動作手順を図4.3に示す．

1. ユーザクライアントが，特定の情報家電機器を利用したいという内容を認証サーバに対して問い合わせる
2. ユーザからの要求を受け取った認証サーバは，保持しているデータベースを参照し，そのユーザの情報家電機器に対するアクセス権限に関する情報と，認証に必要な鍵を検索
3. 認証サーバは，検索の結果得られた情報をもとに，応答メッセージを必要な暗号化処理を行なって作成し，ユーザクライアントに返す
4. 認証サーバからの応答メッセージを受けとったユーザクライアントは，情報家電機器の利用要求とともに，認証サーバから与えられたアクセス権限や，鍵の情報を，情報家電機器に送る
5. 情報家電機器はユーザクライアントからの利用要求とともに送られる情報をもとにそのユーザに対する機能の提供の可否を決定する

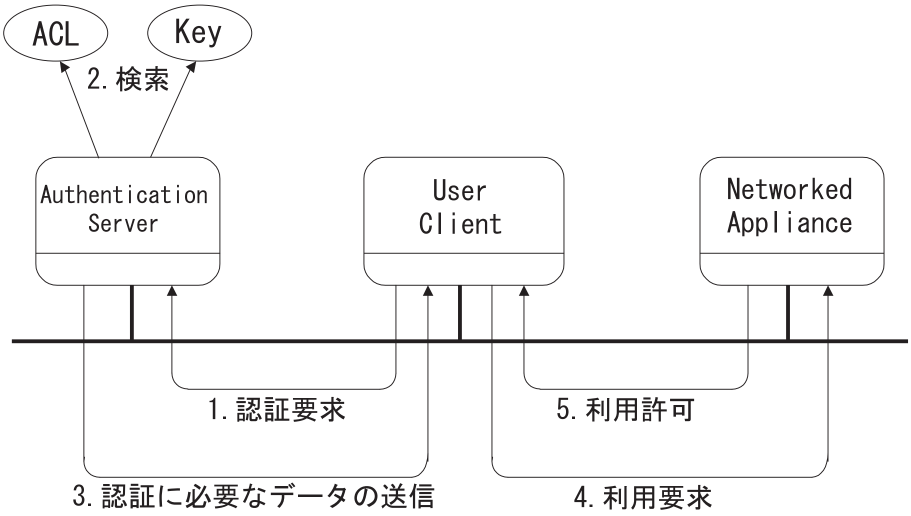


図 4.3: システム全体の動作手順

4.3　認証サーバの設計

認証サーバは，ユーザクライアントからの要求に対して，特定の情報家電機器を利用するために必要な情報を検索し，その情報をユーザクライアントに返す役割を持つ．

以下に，認証サーバに必要となる機能とその設計について説明し，その動作手順について述べる．

4.3.1 暗号鍵データベースの保持・検索

認証サーバは，システムが管理する情報家電機器とユーザそれぞれに対応した秘密鍵である親鍵を共有し，データベースとして保持する．そして，それらの鍵の追加，変更，削除などを管理する機能を持つ．

ユーザクライアントから認証サーバへ送られる要求メッセージの中には，ユーザ名と，利用したい情報家電機器名が含まれている．そのメッセージを受けとった認証サーバは，メッセージ内のユーザ名と機器名に対応した親鍵を鍵のデータベースから検索する．ユーザ名に対応する親鍵は，そのユーザに対応したクライアントにのみ送られるべきメッセージを暗号化するのに利用し，情報家電機器名に対応する親鍵は，その情報家電機器にのみ送りたいメッセージを暗号化するのに利用する．つまり，認証サーバによる応答メッセージ自体は，ユーザクライアントに送られるが，そのメッセージ内の情報で，情報家電機器の親鍵で暗号化されている部分は，指定された情報家電機器以外では意味をなさず，内容を破壊できても意図的に改竄することは困難である．一方，ユーザは，自らを認証するためのデータを敢えて破壊することはないと考えられる．

また，それぞれのユーザや情報家電機器と共有する親鍵の受け渡しに関しては，新たなユーザや情報家電機器をシステム上に追加するときに，オンラインまたはオフラインで登録を行う必要がある．オフラインで登録作業を実行する場合は，特に問題ないが，オンラインで鍵の受け渡しを行うには，その安全性を保つための機構が必要である．

本研究では，システム全体の簡便性を考え，認証サーバがユーザや情報家電機器と共有する秘密鍵は，ユーザ登録や情報家電機器の追加の際に，オフラインにその受け渡しが行われるものとし，以降認証サーバと共有する秘密鍵は，安全に管理されていることを前提とする．

4.3.2 セッション鍵の生成

認証サーバは，ユーザクライアントと情報家電機器の間で行なわれる通信の際の認証で利用するセッション鍵を，必要に応じて生成し，ユーザクライアント側と情報家電機器側に提供するKDCとしての役割を持つ．ここで，ユーザクライアントや，情報家電機器から見たとき，認証サーバは，十分に信頼できるノードとしてみなされることが前提となる．

認証サーバは，指定したユーザクライアントおよび情報家電機器だけが，生成されたセッション鍵を使用可能であることを保証しなくてはならない．そのために，認証サーバは，鍵データベースで保持する各ユーザおよび情報家電機器の親鍵を利用する．つまり，認証サーバによる応答メッセージに含まれるセッション鍵の情報は，セッション鍵を受け取るべきノードの親鍵により暗号化された状態で提供される．

4.3.3 ACLの保持・検索

認証サーバは，ユーザの情報家電機器に対するACLを保持している．そして，ユーザから認証サーバへ送られる要求メッセージの中には，そのユーザ名と，利用したい情報家電

表 4.1: ユーザクライアントと情報家電機器上で利用される情報

内容	ユーザクライアント	情報家電機器
ユーザ名	×	○
機器名	×	×
セッション鍵	○	○
鍵の有効期限	○	○
アクセス権限	×	○

○ ... 必要，× ... 必要なし

機器名が含まれるので，認証サーバは，要求メッセージ内にあるユーザ名と情報家電機器名に対応するACLのエントリを検索する．ACLのエントリに関する情報は，対応する情報家電機器側に送るべき内容なので，認証サーバによる応答メッセージの中では，その情報家電機器の親鍵で暗号化したデータとして返される．

4.3.4 返答メッセージの作成

認証サーバからユーザクライアントに返答されるメッセージの内容は，大きく分けて二つの内容から構成される．一つは，ユーザクライアント自身が利用するためのデータであり，もう一つは，ユーザを経由して情報家電機器に送られることで情報家電機器上で利用されるデータである．返答メッセージ内の情報の中で，ユーザクライアント，情報家電機器のそれぞれによって利用される情報を，表4.1にまとめる．

返答メッセージ内の情報の中で，ユーザクライアントや情報家電機器に受け取られるべきセッション鍵と，情報家電機器に受け取られるべきアクセス権限に関する情報に関しては，それぞれが指定されたノードに安全に渡すために，各親鍵による暗号化が行われる．認証サーバがユーザクライアントに返信するメッセージの中に含まれるデータの構造と，その暗号化の様子を図4.4に示す．そして，本研究では，情報家電機器の親鍵で暗号化したデータを認証チケットと定義する．

4.3.5 認証サーバの動作手順

認証サーバが，ユーザクライアントによる要求に応答する間に，認証サーバ内で行われる処理は，以下の通りである．その様子を図で表すと，図4.5のようになる．

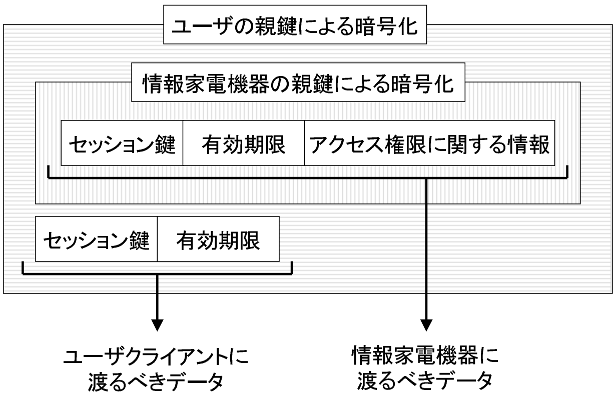


図 4.4: 認証サーバによる応答メッセージの構造

1. ユーザクライアントから使用ユーザ名と利用したい情報家電機器名に対応する認証チケットの要求を受ける
2. 受け取ったユーザ名および情報家電機器名をもとに，それぞれに対応する親鍵をデータベースから検索
3. 受け取ったユーザ名および情報家電機器名をもとに，その二者に対応する ACL のエントリをデータベースから検索
4. ユーザクライアントおよび，情報家電機器の二者が通信するときに使用するセッション鍵を生成
5. 情報家電機器上で使用される情報を，情報家電機器の親鍵で暗号化
6. ユーザクライアントへ送信されるメッセージを，使用するユーザの親鍵で暗号化
7. 要求メッセージを送信してきたユーザクライアントに対し，作成された応答メッセージを返信

4.4 情報家電機器制御機構の設計

情報家電機器制御機構は情報家電機器上のコンピュータ上で動作し，ユーザクライアントからの利用要求を受け，それに応じたサービスを提供する役割を持つ．また，その際には，ユーザのアクセス権限や，情報家電機器上の資源の状態に応じて，要求に対する動作に制限をかける機能を持つ．

以下に，情報家電機器制御機構上で必要となる機能とその設計について詳しく説明し，そ

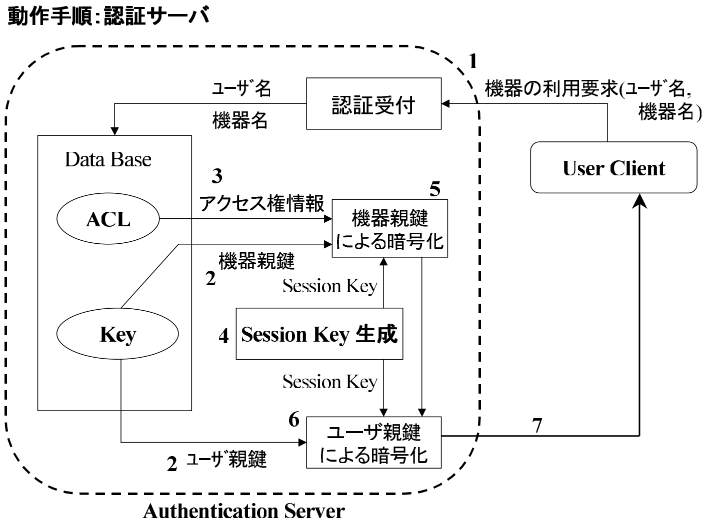


図 4.5: 認証サーバがクライアントの要求に応答するまでの流れ

の動作手順について述べる．

4.4.1 情報家電機器オブジェクトの提供

情報家電機器は，ポストPC上で情報家電オブジェクトとして抽象化され，ネットワーク上の分散オブジェクトとして他のノードから参照を受ける．本研究では，情報家電オブジェクトをJava RMI[27]のリモートオブジェクトとして実装し，情報家電機器の機能は，情報家電オブジェクト内のリモートメソッドに対応させる．例えば，情報家電機器としてのテレビを想定すると，そのテレビ上のポストPC内では，電源ON／OFF，チャンネル切替，ソース切替などのメソッドをもったリモートオブジェクトが存在し，他のノードからのリモートメソッド呼び出しにより，それらの機能が実際の動作を始める．

本システムでは，リモートメソッドを実行する際には必ず，ユーザのアクセス権を検証するモジュールと，機器上の資源の状態を管理するモジュールが参照され，情報家電機器の動作を制限する処理が実行される．例えば，リモートメソッド呼び出しを行ったユーザが，そのメソッドの実行権を持っていない場合，アクセス権の検証モジュール内で例外が発生し，メソッド内に含まれる実際の家電機器への動作命令は実行されない．同様に，機器上の資源の状態を管理するモジュールでは，資源の利用可能性が検証され，家電機器への動作命令の可否に反映される．

4.4.2 認証チケットの登録

初期状態の状態で，情報家電機器上にはユーザを認証するための情報は保持されていない．なぜなら，2.1.2節で述べたように，ポストPC上のストレージ機能の制限をなるべく受けずに，多種多様な情報家電機器をシステム内で扱うことを想定しているからである．情報家電機器上で認証やアクセス管理に必要な情報は，ユーザクライアントからのメッセージに含まれる認証チケットを復号することで得られる．認証チケットは必要に応じて，ユーザクライアントから情報家電機器に渡されるので，情報家電機器上では，あらかじめ認証やアクセス管理のために利用する情報を保持しておく必要がない．

情報家電機器側では，ユーザクライアントから認証チケットが登録されるのを常に待ち受けている．認証チケットから得られる情報としては重要なのは，ユーザクライアントと情報家電機器間での認証のために利用するセッション鍵，そのユーザと情報家電機器に対応するACLのエントリである．認証チケットは，ユーザクライアントから受け取ったものだが，もとは認証サーバによって作成されたもので，その内容は認証サーバと情報家電機器で共有されている親鍵によって暗号化されている．従って，情報家電機器は，それを復号し，その構造の妥当性をチェックすることで，内容の正当性を確認できる．

認証チケットの登録の際，情報家電機器は自分の親鍵で認証チケットの内容を復号し，セッション鍵とACLエントリの情報を得る．そして，乱数(以下チャレンジ)を生成し，認証チケットから得たセッション鍵やACLエントリと共に，認証エントリとして記憶する．一方，ユーザクライアント側には，認証エントリへのポインタとチャレンジの組を返す．

4.4.3 ユーザの認証とアクセス権の検証

ユーザクライアントから，情報家電機器オブジェクト内のリモートメソッドを実行する際には，まずユーザの認証とアクセス権の検証が行なわれる．情報家電機器上での認証には，4.4.2節で説明した認証チケットから得られるセッション鍵を利用する．同じく，アクセス権の検証に関しても，認証チケットから得られるACLエントリを利用する．

ユーザクライアントが情報家電機器を利用する際には，事前に渡されたチャレンジをセッション鍵で暗号化したものと，同時に指定され認証エントリのインデックスが，情報家電機器に渡される．それらのデータを受け取った情報家電機器は，暗号化されたチャレンジをセッション鍵で復号したデータと，インデックスが指定する認証エントリ内のチャレンジのデータとを比較する．それらが同一のデータであった場合，ユーザのアクセス権限が，インデックスが指定する認証エントリ内のACLエントリに記述されているということが確認できるので，そのACLエントリの内容通りにアクセス制御が行われる．

また，1度利用された認証エントリは，メッセージの不正な再利用を防ぐために無効化される．しかし，それと同時に，その中のセッション鍵とACLエントリの情報および新たに生成されたチャレンジを用いて，新しい認証エントリが作成される．そして，新しい認証エントリを次回の認証で利用するために，そのインデックスとチャレンジの組がユーザクライアント側に返される．

4.4.4　機器上の資源の割り当て

情報家電機器上で，ユーザに対する認証やアクセス権限の検証が行われた後，機器上の資源の状態を基にした利用の可否が決定される．以下に，情報家電機器の利用に適した資源の管理方法について説明する．

本システムでは，情報家電オブジェクトが提供する個々のリモートメソッドを機器上の資源と定義し，断続的機能，持続的機能の二つの特徴をもった資源に分類する．断続的機能と持続的機能の説明を以下に示す．

断続的機能

　一時的に操作を受けて機能を提供し，完結するもの

　例:状態の取得，設定の変更，電源ON／OFF

持続的機能

　操作を受けてから持続的に機能を提供するもの

　例:ディスプレイの表示，メディアの再生

本システム上で，持続的機能を提供する場合には，その機能の開始メソッドと，それに対応した終了メソッドを定義する．そして，開始メソッドの実行から，終了メソッドの実行までの間に，実際の情報家電機器による機能の提供が持続的に行われることとする．例えば，VCRを考えると，テープの再生命令が開始メソッドに対応し，停止命令が終了メソッドに対応する．また，再生命令の他に，録画命令などが存在する場合は，メソッド内で同じ処理を行うとしてもそれぞれに対応した停止命令として終了メソッドを定義する．

そして，あるユーザによって持続的機能の開始メソッドが呼び出されると，その機能の性質に応じた機能群がロックされ，終了メソッドが呼び出されると，ロックされた機能群が解放される．

このモデルでは，持続的機能へのロックに関しては，ロックをかけたユーザを含め，全てのユーザによる開始メソッドおよび終了メソッドの実行が禁止される．一方，断続的機能に関しては，その性質に応じて2通りのロックの方法が存在する．一つ目は，他の持続的機能のロック方法と同じように全てのユーザに対して実行が禁止されるような方法で，二つ目は，ロックをかけたユーザ以外のユーザに対して実行を禁止する方法である．二つ目の方法でロックされる断続的機能としては，その性質上，開始された持続的機能の実行中に利用できないものが該当する．そして，二つ目の方法でロックされる断続的機能は，開始された持続的機能に付随するする機能として，同時に利用することを想定されたもので，持続的機能を利用したユーザが占有して利用することになる．

例えば，オーディオコンポによる情報家電機器を想定し，そこには，持続的機能として，CD再生機能やMD再生機能，断続的機能として，CDの取り出し，音量レベルの変更，音質の変更，時刻の参照，時刻の変更などの機能が存在するとする．この場合，あるユーザがCD再生機能を開始すると，MD再生機能やCDの取り出し機能は，全てのユーザが利用できないようにロックされ，音量レベルの変更や音質の変更は，CD再生機能を利用しているユーザ以外の命令を受け付けないようにロックされる．時刻の参照機能や時刻の変更機

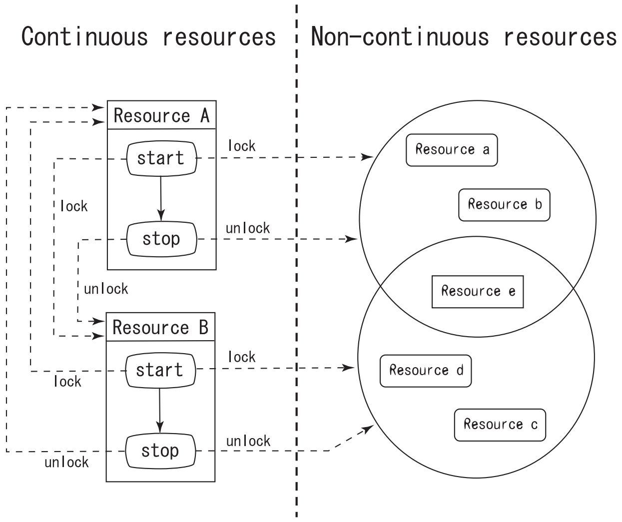


図 4.6: 情報家電機器上の機能どうしのロックの様子

能については，CD 再生機能に特に関係した機能ではないため，ロックかけられることはない．

もし，その情報家電機器上のもう一つの持続的機能として，アラーム CD 再生機能が存在したとすると，その機能が開始される時，今度は，持続的機能として通常の CD 再生機能と MD 再生機能がロックされ，断続的機能として音量レベルの変更と音質の変更に加えて，時刻の変更機能もロックされることになる．情報家電機器上での機能のロックに関するモデルを図 4.6 に示す．

4.4.5　資源管理ポリシの保持

情報家電機器上で，資源同士でのロック処理を行うには，ロックする資源やロックされる資源の関係を定義する情報が必要となる．ここで，その情報を資源管理ポリシと定義する．

資源管理ポリシは，その情報家電機器の性質に応じてほぼ静的に決定されるもので，一度定義された後は，変更される頻度は少ない．また，その情報量としても，機能数が増減しない限り変化することはなく，容量の大きな変化は起こらないと考えられる．そこで，本システムでは，資源管理ポリシは，各情報家電機器上で記憶可能とする．

4.4.6　機能のリース

ユーザクライアントが持続的機能の開始メソッドを実行し，その後確実に対応した終了メソッドを実行するならば，機能のロックとその解放の処理は正常に動作するが，ネットワーク上の障害や，クライアント側のエラーが起こり，正しく終了メソッドが実行されない状態が続いてしまうと，情報家電機器上の機能が，ロックされたまま永久に解放されな

いという状態が発生する．

そのような事態を防ぐために，本システムでは機能を一定時間リースするというモデルを利用する．他の機能をロックするようなメソッドの実行要求があった場合，情報家電機器はその機能のリース時間を決定し，クライアント側に提示する．提示したリース時間が経過すると，情報家電機器が自動的にその機能の終了メソッドを実行し，ロックされた機能群は解放される．また，クライアント側から，明示的にリース時間の更新要求があった場合は，リース時間を延長する．従って，クライアント側が正常に動作している限りは，適切に機能のロックが行われ，クライアント側が異常終了してしまっても，ロックの状態は，一定時間経過後にもとの状態に戻ることを保証できる．このリースによるモデルを導入することにより，システム全体の耐故障性は向上する．

4.4.7 情報家電機器制御機構の動作手順

情報家電機器制御機構が，ユーザクライアントから認証チケットの登録を受け，その後，ユーザクライアントからの機能の利用要求の可否を判断する間に行なわれる処理は，以下の通りである．また，この動作手順を図4.7 に示す．

1. ユーザクライアントから認証チケットの登録を受け付ける
2. 認証チケットを情報家電機器の親鍵で復号することで，ユーザクライアントの認証の際に利用するセッション鍵と，ユーザのアクセス権限を表すACL エントリを得る
3. チャレンジを生成し，セッション鍵，ACL エントリと共に認証エントリとして保持し，ユーザクライアントには，チャレンジと認証エントリのインデックスを返す
4. ユーザから，セッション鍵で暗号化されたチャレンジと，そのチャレンジに対応する認証エントリのインデックスと共に，機能の利用要求を受ける
5. 暗号化されたチャレンジをセッション鍵で復号し，対応する認証エントリ内のチャレンジと比較することで，ユーザを認証する
6. ACL エントリの内容を基に，ユーザによる機能の利用権限を検証する
7. 機器上の資源の利用状態を確認し，その機能が利用可能であるかを検証
8. 機能の提供が可能な場合は，必要に応じて資源のロック作業を行う
9. ユーザクライアントから利用要求のあった機能が実際に実行される

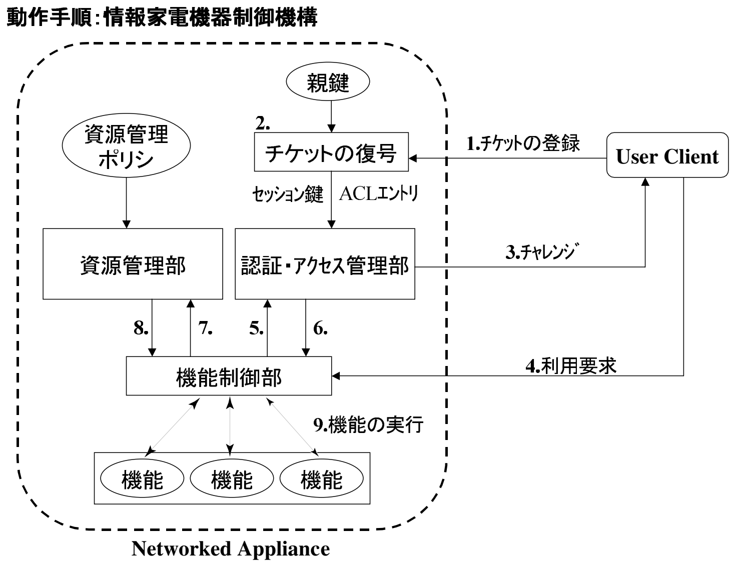


図 4.7: 情報家電機器制御機構による認証と機能提供の流れ

4.5 ユーザクライアントの設計

ユーザクライアントは，ユーザが情報家電機器の共有システムを利用する際のインターフェースとして，GUI(Graphical User Interface)などの機能を提供し，情報家電機器の利用者側端末としての役割を果たす．ユーザクライアントでは，情報家電機器の利用の際に，認証サーバから認証チケットを受け取り，その情報に基づいた認証やアクセス制御を情報家電機器から受ける．

以下に，ユーザクライアントに必要となる機能とその設計について説明し，その動作手順について述べる．

4.5.1 認証チケットの取得と登録

ユーザクライアントから情報家電機器を利用する際には，利用する情報家電機器名とユーザ名を認証サーバに送り，返答メッセージを受け取る．返答メッセージの内容は，図4.4で説明したような構造で暗号化されており，ユーザクライアント側では，

1. 対象となる情報家電機器と共有するためのセッション鍵
2. 情報家電機器の親鍵で暗号化された情報

の2つを取得できる．2については鍵を持っていないユーザクライアント上では，その内容を知れない．2は情報家電機器がユーザクライアントを認証するために必要な情報であり，4.4.2節で定義した認証チケットに相当する．認証チケットは，ユーザクライアントから

利用したい情報家電機器へ送信され，以後その認証チケットの有効期限が切れるまで，情報家電機器上に登録される．登録が完了した際には，次回の認証に必要なチャレンジと，それを含んだ認証エントリのインデックスが返される．

4.5.2 ユーザの親鍵の管理

前節で述べたように，ユーザクライアントには，サーバから受け取った返答メッセージをを復号する機能が必要である．そして，ユーザクライアント上でメッセージの復号を行う場合，ユーザの親鍵が必要である．しかし，複数のユーザクライアントを想定すると，ユーザの親鍵がユーザと認証サーバの間でのみ共有されるといった前提に反することになる．すなわち，ユーザの親鍵が他者に漏洩する危険性が生じる．なぜなら，ユーザクライアント上でユーザの親鍵を使用する場合，各ユーザクライアントがユーザの親鍵を記憶しているか，利用の際にユーザクライアント上にコピーされなくてならないからである．

ユーザクライアント側の端末が，マルチユーザに対応していて，ファイルやメモリの管理の上で鍵の情報のセキュリティを保つ機能を提供しているとしても，鍵の情報が分散してしまうと，その機密性が低下してしまうのは明らかである．また，全てのユーザクライアント端末で，十分なセキュリティ機能を期待できるとは限らない．

そこで，本システムでは，ユーザの親鍵の管理には，Java CardやJava Ringなどの小型ICデバイスを利用する．ICデバイス上では，ユーザの親鍵自体を記憶するだけでなく，それを利用した暗号化や復号の処理も可能である．従って，ユーザがICデバイスを端末に差し込んだときだけ，ユーザクライアントは各ユーザの親鍵による暗号化や復号が可能になる．ユーザクライアントは，暗号化や復号したいメッセージをICデバイスに渡すことで，その結果を取得できるので，各ユーザの親鍵を保管する必要はない．また，ユーザから見ても，自分の親鍵が，使用したユーザクライアント上にコピーされる心配がないので，親鍵の機密性を保てる．

4.5.3 情報家電機器の利用

認証チケットが情報家電機器上に登録されると，ユーザクライアントは，取得したセッション鍵を用いて情報家電機器から認証を受けることが可能になる．情報家電機器の機能の利用は，情報家電オブジェクト上のリモートメソッドの呼び出しという形で実行される．ユーザクライアントは情報家電機器から送られてきたチャレンジを，認証サーバから取得したセッション鍵で暗号化し，そのチャレンジに対応する認証エントリのインデックスと共にリモートメソッドに渡す．そして，情報家電機器側での認証が成功し，アクセス権限が認められると，実際の情報家電機器への命令が実行される．また，セッション鍵の有効期限切れ以外の場合は，命令の実行の可否に関わらず，次のアクセスに必要な次のチャレンジと，それを含む認証エントリのインデックスを情報家電機器側から受け取る．セッション鍵の有効期限切れの場合は，新たな認証チケットを登録するために，4.5.1節で示した認証サーバへの問い合わせの作業を繰り返す．

4.5.4　リースの更新，解放

情報家電機器上の資源が，情報家電機器制御機構内での排他制御の対象となっている場合，ユーザクライアントがその資源を利用する際に，情報家電機器制御機構からその資源に対するリース時間が提示される．提示されたリース時間内では，そのユーザクライアントが，その資源を占有して利用できる．しかし，リース時間経過後には，それを保証できないので，リソースの使用時間が，リース時間を上回る場合は，ユーザクライアントが明示的に情報家電機器側にリース時間の更新を要求する．また，リース時間経過後に自動的に資源は解放されるが，資源の使用が終了したときに，明示的に資源を解放する機能も提供する．

リースの更新や解放は，情報家電機器上の特別な機能として提供され，その実行の際にも，4.5.3節で述べたように，セッション鍵を利用した暗号化による認証が行われる．

4.5.5　ユーザクライアントの動作手順

ユーザクライアントが，認証サーバから，問い合わせに対する返答を受け取り，その後，情報家電機器を利用するまでに行う処理は，以下の通りである．また，この動作手順を図4.8に示す．

1　ユーザがクライアント端末にICデバイスを差し込む

2　認証サーバに，ユーザ名と利用したい情報家電機器名を問い合わせ，それに対応した返答メッセージを受け取る

3　認証サーバからの返答メッセージをICデバイスを用いて復号し，セッション鍵と，認証チケットを取得する

4　認証チケットを情報家電機器側に登録し，チャレンジと認証エントリのインデックスを受け取る

5　セッション鍵でチャレンジを暗号化したデータと，そのチャレンジを含む認証エントリのインデックスを情報家電機器に渡し，機能の利用を試み，その結果を得る (セッション鍵の有効期限切れの場合は6b，それ以外の場合は6aへ)

6a　次回のアクセス時に利用するためのチャレンジと，認証エントリのインデックスを受け取り，5の処理を繰り返す

6b　2以降の処理をやり直す

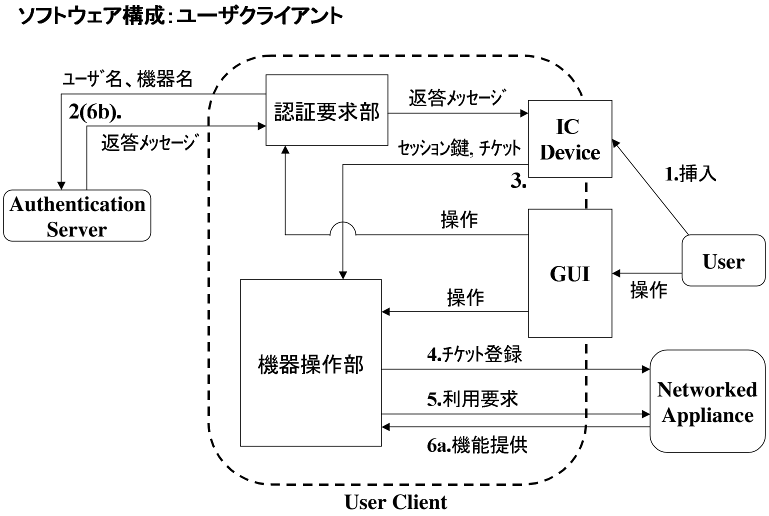


図 4.8: ユーザクライアントから機器を利用するまでの流れ

4.6　本章のまとめ

本章では，ネットワークを介して情報家電機器を共有利用を支援するシステムを構築する際に必要とされる認証機能や資源管理機能を提供し，情報家電機器に対する計算負荷の軽減や，開発者への負荷の軽減を実現する機構**SCNA** を提案した．本章後半では，SCNA を構成する上で主要な要素となる認証サーバ，ユーザクライアント，情報家電機器制御機構の3者に関する設計，動作手順について詳しい説明を述べた．

次章では，本章で述べた設計を基に，具体的にSCNA の実装を行い，それを用いた応用システムの構築例について述べる．

第 5 章

情報家電共有機構SCNAの実装

本章では，4章で述べた設計に基づき，情報家電機器共有機構SCNAを実装する方法について説明する．前半では，認証サーバ，情報家電機器制御機構，ユーザクライアントの個別の実装に関して説明する．後半では，それらの実装をパッケージ化したAPIについて説明し，応用システムの構築例について述べる．

5.1 実装の概要

本節では，SCNAの具体的な実装を行なうにあたって，本研究で使用した開発環境に関する説明を述べ，今回実装した主な項目についての概要を示す．

5.1.1 実装環境

SCNAの実装は，表6.1に示す開発環境上で行なった．開発言語に関しては，マルチプラットフォーム性を持ったJava言語を用いることで，情報家電ネットワーク上の様々な家電機器，およびコンピュータ上での，SCNAを利用した統一的なセキュリティシステムの開発が容易となった．

表 5.1: 実装環境

項　　目	環 境
CPU	K6-III 450MHz
Memory	128MB
OS	RedHat Linux 6.2
JDK version	Sun JDK 1.2.2 006 linux i386
Cryptography library	Sun JCE 1.2.1

5.1.2 実装項目

SCNAの実装は，認証サーバ，ユーザクライアント，情報家電機器制御機構の3つの部分に分けられ，それらを利用した開発のためのAPIの提供，実例としての応用システムの提供といった項目から構成される．各部分の詳細な構成は，以下のようになっている．

認証サーバ

認証サーバの持つ機能は，親鍵の管理，ACLの管理，セッション鍵の生成，メッセージの作成の4つに細分化される．本研究では，それぞれに関する実装を提供する．

情報家電機器制御機構

情報家電管理機構の持つ機能は，モートオブジェクトの提供，チケットの解読，アクセスの管理，機器上の資源の管理，ユーザクライアントとの通信の5つに細分化される．本研究では，それぞれに関する実装を提供する．

表 5.2: 認証サーバの機能と実装クラスの対応

機能	クラス
親鍵の管理	`KeyDBImpl`
ACLの管理	`AclEntry`, `AclDBImpl`
セッション鍵の生成	`KeyGenerator`
メッセージの作成	`TicketContainer`, `Ticket`

ユーザクライアント

ユーザクライアントの持つ機能は，認証サーバとの通信，情報家電機器との通信，リースの更新，ユーザインタフェースの4つに細分化される．本研究では，それぞれに関する実装を提供する．

API

本システムが提供する機能を実際に用いてシステムを開発する際に，開発者が利用するAPIを提供する．

応用システム

本システムを利用することによって開発可能な情報家電機器の共有システムの一例として，実際に利用可能なアプリケーションを構築する．

5.2 認証サーバの実装

本節では，認証サーバの機能を提供するために必要なJavaのクラスの実装について述べる．まず，認証サーバのクラス構成を示し，認証サーバの各機能ごとに，対応するクラス名を挙げる．その後，各クラスに関する詳しい説明を述べる．

5.2.1 認証サーバの機能と実現クラス

図5.1に認証サーバのクラス構造を示す．また，認証サーバに必要な機能と，それを実現するクラスの対応を，表5.2に示す．

5.2.2 認証サーバを構成するクラス

以下に認証サーバを構成する各クラスの詳細について述べる．

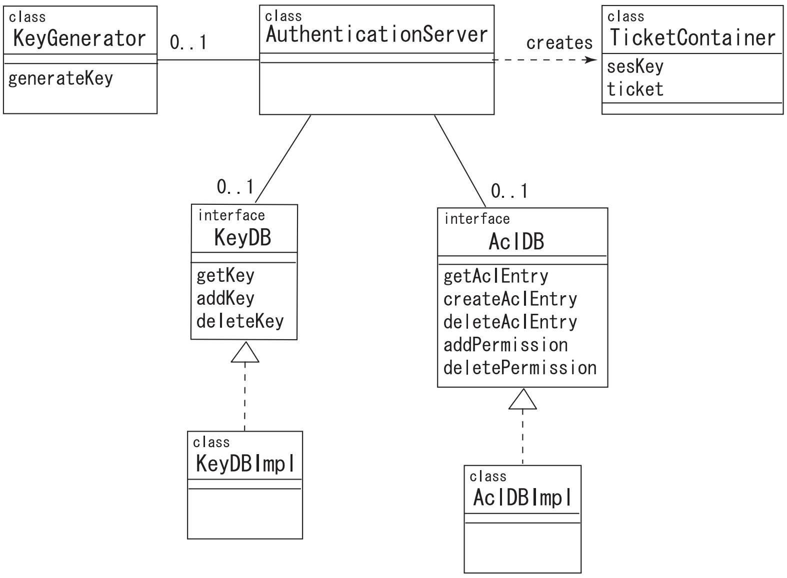


図 5.1: 認証サーバのクラス構成図

`KeyDBImpl` クラス

`KeyDBImpl` クラスは，暗号鍵のデータベースを構成するクラスとして，以下のような機能を持つ．

- ユーザクライアントからの問い合わせに含まれるユーザ名や情報家電機器名から，対応する親鍵を検索する
- ユーザや情報家電機器の追加/削除の際に，対応する親鍵の追加/削除を行う

これらの機能は，`KeyDB` インタフェースで定義されており，`KeyDBImpl` クラスは，`KeyDB` インタフェースの実装クラスである．`KeyDBImpl` クラスでは，Java の `Hashtable` クラスを利用した実装がされているが，`KeyDB` インタフェースを用いれば，RDB(Relational Data Base) などを利用した実装も可能となる．`KeyDB` インタフェースの内容を図5.2に示す．

`AclDBImpl` クラス

`AclDBImpl` クラスは，暗号鍵のデータベースを構成するクラスとして，以下のような機能を持つ．

- ユーザクライアントからの問い合わせに含まれるユーザ名と情報家電機器名に対応する `AclEntry` を検索する
- ユーザや情報家電機器の追加/削除の際に，対応する `AclEntry` の追加/削除を行う

```
package jp.ac.keio.sfc.ht.kmsf.scna.server;
import java.security.*;
public interface KeyDB
{
  public Key getKey(String name);
  public synchronized void addKey(String name);
  public synchronized void deleteKey(String name);
}
```

図 5.2: KeyDB インタフェース

- 指定されたユーザに対する，情報家電機器の各機能へのパーミッションの追加／削除

これらの機能は，AclDB インタフェースで定義されており，AclDBImpl クラスは，AclDB インタフェースの実装クラスである．AclDBImpl クラスでは，Java の Hashtable クラスを利用した実装がされているが，AclDB インタフェースを用いれば，RDB などを利用した実装も可能となる．AclDB インタフェースの内容を図5.3に示す．

AclDB の getAclEntry() メソッドにより得られる AclEntry は，情報家電機器上に送られることで，ユーザの情報家電機器の各機能に対するパーミッションの確認に利用される．AclEntry クラスの内容を，図5.4に示す．

KeyGenerator クラス

KeyGenerator クラスは，特定のユーザクライアントと情報家電機器が共有するセッション鍵を生成する機能を提供する．今回の実装では，Sun JCE(Java Cryptography Extention) が提供する javax.crypto.KeyGenerator クラスを利用し，DES の秘密鍵を生成する．

TicketContainer クラス

TicketContainer クラスは，認証サーバからユーザクライアントに送信されるメッセージに含まれる情報を定義したクラスである．しかし，実際にユーザクライアント側に送信される場合には，ユーザの親鍵による暗号化を行い，byte 列として送信される．従って，TicketContainer クラス内の変数は，特定のユーザクライアントの上でしか参照できない．TicketContainer クラスの内容を，図5.5に示す．

TicketContainer クラス内の Key オブジェクト変数 sesKey は，ユーザクライアント上で利用されるセッション鍵に対応し，byte配列の変数 ticket は，情報家電機器に登録するための認証チケットに対応する．ticket の内容は，認証サーバ上で作成した Ticket クラスのインスタンスオブジェクトを，利用される情報家電機器の親鍵で暗号化したものである．

```
package jp.ac.keio.sfc.ht.kmsf.scna.server;
import  jp.ac.keio.sfc.ht.kmsf.scna.common.*;
public interface AclDB
{
 public AclEntry getAclEntry(String usrName, String appName);
 public synchronized void createAclEntry(String usrName,String appName);
 public synchronized void deleteAclEntry(String usrName,String appName);
 public synchronized void addPermission(String usrName, String appName,
                                        int funcNo);
 public synchronized void deletePermission(String usrName,String appName,
                                           int funcNo);
}
```

図 5.3: AclDB インタフェース

```
package jp.ac.keio.sfc.ht.kmsf.scna.common;
public class AclEntry implements Serializable
{
  boolean[] permission;
  String usrName;
  String appName;
  /* 省略 */
  public boolean checkPermission(int funcNo){/* 省略 */}
}
```

図 5.4: AclEntry クラス

従って，その内容がユーザクライアント上で解読されたり，意図的に改竄されたりすることを防止できる．Ticket クラスの内容を図5.6に示す．

5.3 情報家電機器制御機構の実装

本節では，情報家電機器制御機構の機能を提供するために必要なJavaのクラスの実装について述べる．まず，情報家電機器制御機構のクラス構成を示し，情報家電機器制御機構

```
package jp.ac.keio.sfc.ht.kmsf.scna.common;
import java.security.*;
public class TicketContainer implements Serializable
{
  private Key sesKey;
  private byte[] ticket;
  /* 省略 */
  public Key getSesKey(){/* 省略 */}
  public byte[] getTicket(){/* 省略 */}
}
```

図 5.5: `TicketContainer` クラス

表 5.3: 情報家電機器制御機構の機能と実装クラスの対応

機能	クラス
情報家電オブジェクトの提供	`ApplianceImpl`
チケットの解読	`TicketResolver`, `Ticket`
アクセスの管理	`AuthenticationManagerImpl` , `AclEntry` , `AMReport`
資源の管理	`ResourceManagerImpl` , `RMReport`
ユーザクライアントとの通信	`RemoteApplianceReport`

の各機能ごとに，対応するクラス名を挙げる．その後，各クラスに関する詳しい説明を述べる．

5.3.1 情報家電機器制御機構の機能と実現クラス

図5.7に情報家電機器制御機構のクラス構造を示す．また，情報家電機器制御機構に必要な機能と，それを実現するクラスの対応を，表5.3に示す．

5.3.2 情報家電機器制御機構を構成するクラス

以下に，情報家電機器制御機構を構成するクラスの詳細について述べる．

```
package jp.ac.keio.sfc.ht.kmsf.scna.common;
import java.io.*;
import java.security.*;
public class Ticket implements Serializable
{
  private String usrName;
  private String appName;
  private Key key;
  private AclEntry acl;
  private long timeStamp;
  /* 省略 */
  public String getUsrName(){/* 省略 */}
  public String getAppName(){/* 省略 */}
  public Key getKey(){/* 省略 */}
  public AclEntry getAclEntry(){/* 省略 */}
  public long getTimeStamp(){/* 省略 */}
}
```

図 5.6: `Ticket` クラス

`ApplianceImpl` クラス

`ApplianceImpl` クラスは，情報家電オブジェクトが共通して持つ機能として，チケットの登録，リースの更新，リースの解放といった操作を受け付けるメソッドを持つ．それらのメソッドは，`Appliance` インタフェースで定義されており，`ApplianceImpl` クラスは，`Appliance` インタフェースの実装クラスとなる．`Appliance` インタフェースの内容を図5.8に示す．

そして，実際に個々の情報家電オブジェクトを定義するクラスを作成する場合には，`ApplianceImpl` クラスを継承し，機器固有のインタフェースを実装する．機器固有のインタフェースとは，開発者が新たな情報家電機器を記述する際に，用意するものである．例えば，図5.9に示す `TestAppliance` インタフェースでは，その機器固有の機能として，`func0()` から `func2()` の3つのメソッドが定義されている．

`TicketResolver` クラス

`TicketResolver` クラスは，ユーザクライアントから受け取った認証チケットの内容を解読するためのクラスである．`TicketResolver` クラスでは，まず，情報家電機器の親鍵で暗号化されたbyte配列である認証チケットを復号する．そして，復号されたデータから

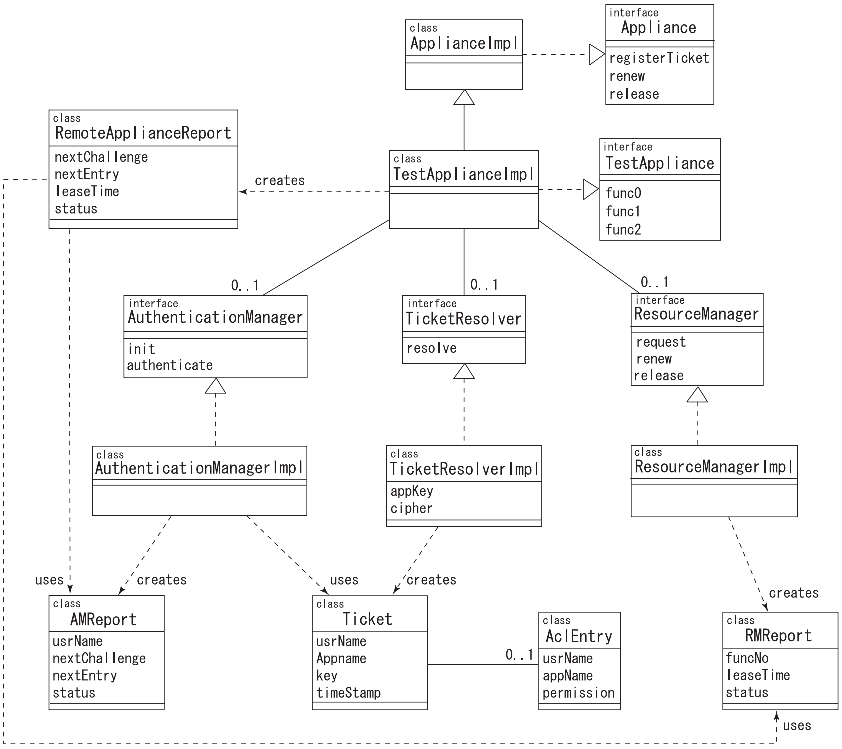


図 5.7: 情報家電機器制御機構のクラス構成図

```
package jp.ac.keio.sfc.ht.kmsf.scna.common;
import java.rmi.*;
public interface Appliance extends Remote
{
  public RemoteApplianceReport registerTicket(byte[] ticket)
                                              throws RemoteException;
  public RemoteApplianceReport renew(int entry,int funcNo,
                                     byte[] challenge)
                                    throws RemoteException;
  public RemoteApplianceReport release(int entry,int funcNo,
                                       byte[] challenge)
                                      throws RemoteException;
}
```

図 5.8: Appliance インタフェース

```
package jp.ac.keio.sfc.ht.kmsf.scna.common;
import java.rmi.*;
public interface TestAppliance extends Remote
{
  static final int funcs = 3;
  public RemoteApplianceReport func0(int entry,byte[] challenge)
         throws RemoteException;
  public RemoteApplianceReport func1(int entry,byte[] challenge)
         throws RemoteException;
  public RemoteApplianceReport func2(int entry,byte[] challenge)
         throws RemoteException;
}
```

図 5.9: `TestAppliane`インタフェースの例

`Ticket`クラスのインスタンスを生成することで，認証に必要な情報を提供する．

`AuthenticationManagerImpl`クラス

`AuthenticationManagerImpl`クラスは，ユーザのアクセス管理をするクラスであり，情報家電機器上のリモートメソッド実行時に参照される．`AuthenticationManagerImpl`クラスが提供する機能としては以下の4つが挙げられる．

- 認証チケットの登録
- ユーザクライアント側で暗号化されたチャレンジの復号，比較によるユーザの認証，対応するACLエントリからアクセスの成否を決定
- 認証エントリの作成
- 認証，アクセス制御の結果の通知

これらの機能は，`AuthenticationManager`インタフェースで定義されており，`AuthenticationManagerImpl`クラスは`AuthenticationManager`インタフェースの実装クラスである．

`AuthenticationManager`インタフェースを用いることで，各情報家電機器上で記憶できる認証エントリの数や，認証エントリを破棄するアルゴリズムに応じた実装が可能となる．`AuthenticationManagerImpl`クラスでは，256個の認証エントリを，FIFO(First-In, First-Out)で管理している．

```
package jp.ac.keio.sfc.ht.kmsf.scna.appliance;
import jp.ac.keio.sfc.ht.kmsf.scna.common.*;
import java.security.*;
public interface AuthenticationManager
{
  public AMReport init(Ticket ticket); /* 認証チケットの登録      */
                                       /* 認証エントリの作成 */
  public AMReport authenticate(int entry, int funcNo,
                               byte[] encrypted);
                                       /* 認証とアクセス権の検証 */
                                       /* 認証エントリの作成 */
}
```

図 5.10: AuthenticationManager インタフェース

表 5.4: AMReport クラスが持つステータスワード

変数名	説明
SESKEY_TOO_OLD	セッション鍵の有効期限切れ
ENTRY_INVALID	認証エントリが無効
PERMISSION_DENIED	アクセス権限が無い
CHALLENGE_NOT_IDENTICAL	チャレンジが間違っている
SESKEY_REGISTERED	セッション鍵の登録完了
AUTHENTICATED	アクセス可能

AuthenticationManager インタフェースの内容を図5.10に示す．

AuthenticationManager インタフェースで定義されるメソッドは，返り値として AMReport クラスのインスタンスを返す．AMReport クラスは，各メソッド内で行われた処理結果を通知するためのクラスであり，ステータスワードと，次回のチャレンジ，認証エントリへのポインタを含んでいる．AMReport クラスが通知するステータスワードとその説明を表5.4に示す．

ResourceManagerImpl クラス

ResourceManagerImpl クラスは，情報家電機器上の資源を管理するクラスであり，情報家電機器上のリモートメソッド実行時に参照される．ResourceManagerImpl クラスが提供

する機能としては以下の3つが挙げられる．

- 資源の利用要求を受け付け，その成否を決定し，競合する可能性のある資源にロックをかける
- リースの更新の受け付け
- 資源の解放の受け付け
- 処理結果の通知

これらの機能は，`ResourceManager`インタフェースで定義されており，`ResourceManagerImpl`クラスは`ResourceManager`インタフェースの実装クラスである．`ResourceManager`インタフェースの内容を図5.10に示す．

```
package jp.ac.keio.sfc.ht.kmsf.scna.appliance;
import jp.ac.keio.sfc.ht.kmsf.scna.common.*;
public interface ResourceManager
{
  public RMReport request(int funcNo, String usrName);
                         /* 資源の利用の可否を決定 */
                         /* 競合資源のロック       */
  public RMReport renew(int funcNo, String usrName);
                         /* リースの更新 */
  public RMReport release(int funcNo, String usrName);
                         /* 資源の解放 */
}
```

図 5.11: `ResourceManager`インタフェース

`ResourceManager`インタフェースで定義されるメソッドは，返り値として`RMReport`クラスのインスタンスを返す．`RMReport`クラスは，各メソッド内で行われた処理結果を通知するためのクラスであり，ステータスワードや，リース時間の情報を含んでいる．`RMReport`クラスが通知するステータスワードとその説明を表5.5に示す．

`RemoteApplianceReport`クラス

`RemoteApplianceReport`クラスは，情報家電機器上で行われた認証やアクセス制御，資源管理などの処理の結果をユーザクライアント側に通知するためのクラスである．具体的には，`AMReport`クラスのインスタンスや`RMReport`クラスのインスタンスを参照し，ユー

表 5.5: RMReport クラスが持つステータスワード

変数名	説明
ALREADY_LOCKED	資源がロックされていて利用不可能
PROPERLY_LOCKED	必要なロックがかけられ，資源が利用可能
PROPERLY_UNLOCKED	資源解放の完了

表 5.6: ユーザクライアントの機能と実装クラスの対応

機能	クラス
認証サーバとの通信	DecodeManagerImpl，TicketContainer
情報家電機器との通信	ChallengeManager，RemoteAppliacneReport
リースの更新	RenewManager
ユーザインタフェース	ClientGUI

ザクライアントに通知すべき内容を決定する．含まれる内容は，情報家電機器上で行われた処理の結果を表すステータスワードであり，必要な場合は，次回のチャレンジとその認証エントリへのポインタ，資源のリース時間の情報をまとめてクライアント側に通知する．

5.4　ユーザクライアントの実装

本節では，ユーザクライアントの機能を提供するために必要なJavaのクラスの実装について述べる．まず，ユーザクライアントのクラス構成を示し，ユーザクライアントの各機能ごとに，対応するクラス名を挙げる．その後，各クラスに関する詳しい説明を述べる．

5.4.1　ユーザクライアントの機能と実現クラス

図5.12にユーザクライアントのクラス構造を示す．また，ユーザクライアントに必要な機能と，それを実現するクラスの対応を，表5.6に示す．

5.4.2　ユーザクライアントを構成するクラス

以下に，ユーザクライアントを構成するクラスの詳細について述べる．

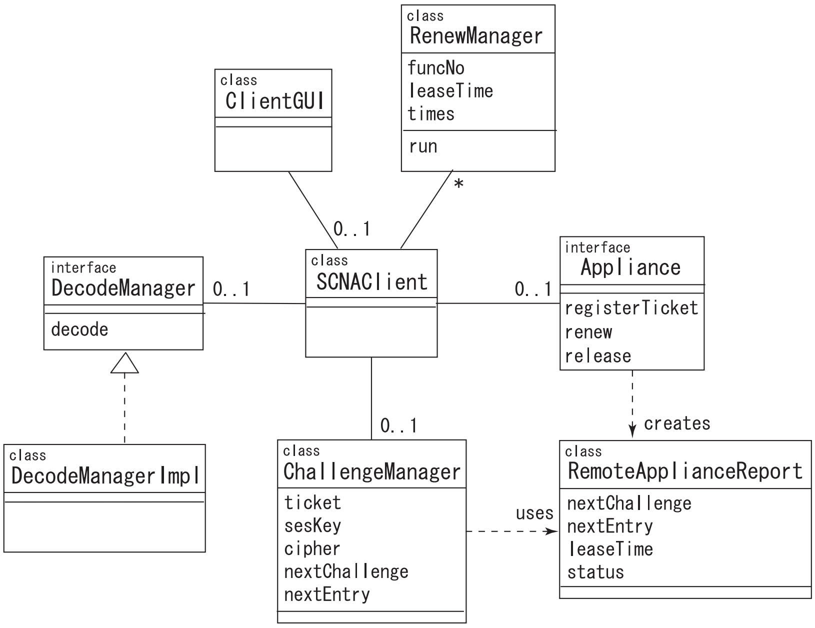


図 5.12: ユーザクライアントのクラス構成図

DecodeManagerImpl クラス

DecodeManagerImpl クラスは，Java Card や Java Ring と通信することで，ユーザの名前を取得し，認証サーバから受け取ったメッセージを復号する機能を提供する．DecodeManagerImpl クラスの持つ機能は，DecodeManager インタフェースで定義されている．DecodeManager インタフェースの内容を図5.13に示す．

```
package jp.ac.keio.sfc.ht.kmsf.scna.client;
public interface DecodeManager
{
  public String getUserName();
  public byte[] decode(byte[] encrypted);
}
```

図 5.13: DecodeManager クラス

ChallengeManagerクラス

ChallengeManagerクラスは，情報家電側から提示されたチャレンジをセッション鍵で暗号化し，認証エントリへのポインタと共に記憶するためのクラスである．情報家電機器へのアクセスの際には，ChallengeManagerクラスのインスタンス内にある暗号化されたチャレンジと認証エントリへのポインタが参照され，情報家電機器からの認証を受けるために使用される．ChallengeManagerクラスの内容を図5.14に示す．

```
package jp.ac.keio.sfc.ht.kmsf.scna.client;
import jp.ac.keio.sfc.ht.kmsf.scna.common.*;
import java.security.*;
import javax.crypto.*;
public class ChallengeManager
{
  public Key sesKey;
  public int nextEntry;
  public byte[] nextChallenge;
  public Cipher cipher;
  /* 省略 */
  public void setNext(RemoteApplianceReport rar)
  {
    /* 次回使用する認証エントリへのポインタの取得 */
    nextEntry = rar.getNextEntry();
    try
      {
        /* 次回使用するチャレンジの暗号化 */
        cipher.init(Cipher.ENCRYPT_MODE,sesKey);
        nextChallenge = cipher.doFinal(rar.getNextChallenge());
      }
    catch(Exception e){e.printStackTrace();}
  }
}
```

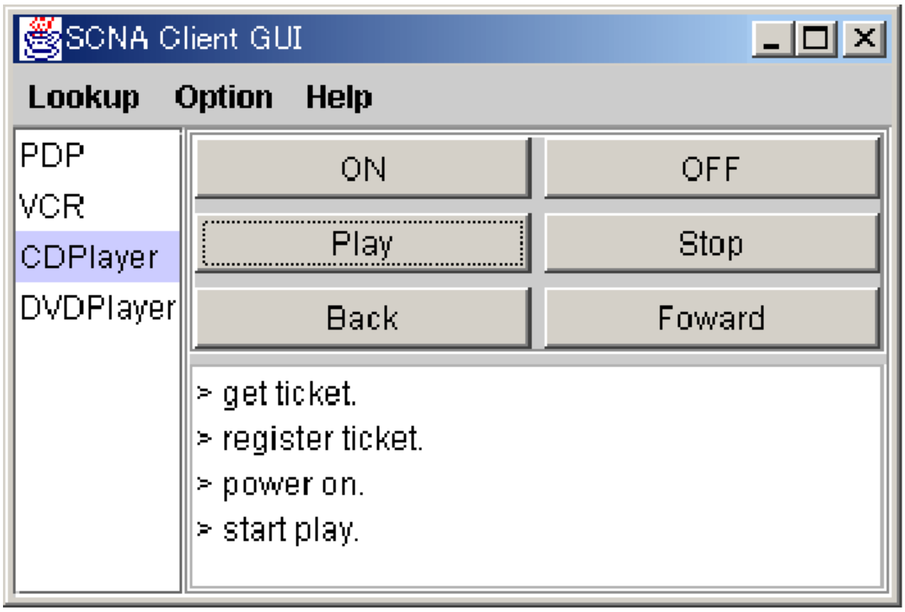
図 5.14: ChallengeManagerクラス

`RenewManager` クラス

`RenewManager` クラスは，情報家電機器から提示されたリース時間を超えて持続的機能をロックして利用する場合に，リースの更新を定期的に行うためのクラスである．

`ClientGUI` クラス

`ClientGUI` クラスは，ユーザが情報家電機器を操作する際に利用しやすいようなGUI(Graphical User Interface)を提供するクラスである．`ClientGUI` クラスによって表示されるGUIの例を図5.15に示す．



```
> get ticket.
> register ticket.
> power on.
> start play.
```

図 5.15: ユーザクライアントGUIの表示例

5.5 API

図5.16に示すように，本システムはパッケージ化されており，情報家電機器の共有システムを開発する際に利用するクラスとして再利用することが可能である．`server` パッケージ，`appliance` パッケージ，`client` パッケージは，それぞれの構成要素を開発する際に利用するクラスやインタフェース群である．`common` パッケージには，各パッケージ内のクラスから共通に参照されるクラスやインターフェース群が含まれ，`util` パッケージには，鍵データベースやACLデータベースを運用する際に利用するクラスが含まれる．

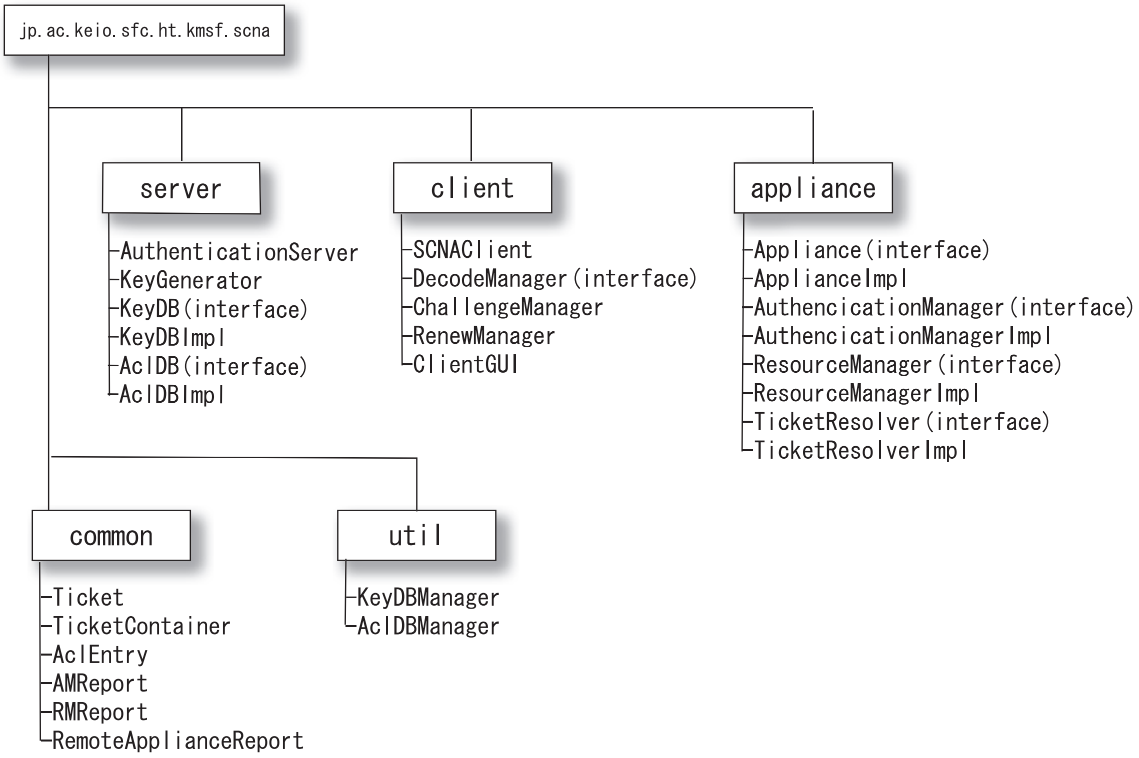


図 5.16: SCNA パッケージの構成

5.6 応用システムの実装

本節では，SCNA を利用した応用システムとして，CD プレーヤを例にとって説明する．まず，CD プレーヤを制御するクラスを作成するためには，図5.17のように実装する．

次に，CD プレーヤを利用する側のクライアント側では，図5.18のようにクラスを作成する．このように，SCNA が提供するクラスを API として利用すると，情報家電機器の共有利用を意識したシステムを容易に構築できる．

5.7 本章のまとめ

本章では，SCNA を構成する認証サーバ，情報家電機器制御機構，ユーザクライアントの実装について個別に詳しい説明を述べた．また，これらの実装を，パッケージ化し，API として再利用しやすい形で提供することにより，情報家電機器の共有システムの開発が容易となった．その例として，CD プレーヤの機能を提供する側と利用する側のプログラムのサンプルコードを取り上げた．

```java
import jp.ac.keio.ht.kmsf.scna.appliance.*;
import jp.ac.keio.ht.kmsf.scna.common.*;
public class CDPlayerImpl extends ApplianceImpl implements CDPlayer
{
  /* 省略 */
  public RemoteApplianceReport play(int entry, byte[] challenge, int track) throw
    String functionID = "play";
    AMReport rar = authenicattionManager.authenticate(entry, challenge);
   RMReport rmr = resourceManager.lock(functionID);
    if(rar.status == AMReport.AUTHENTICATED &&
       rmr.status == RMReport.AVAILABLE)
      {/* 再生の動作内容 */}
    /* 省略 */
  }
  public RemoteApplianceReport stop(int entry, byte[] challenge) throws RemoteExc
  {/ *省略 */}
  /* 省略 */
}
```

図 5.17: CD プレーヤのサンプルコード

```
import jp.ac.keio.ht.kmsf.scna.client.*;
import jp.ac.keio.ht.kmsf.scna.common.*;
import java.rmi.*;

public class CDClient extends SCNAClient
{
  Remote remoteAppliance;

  public CDClient()
  {
    super();
    remoteAppliance = super.lookup("CDPlayer");
    /* 省略 */
  }

  public void playCD(int track)
  {
    int entry = challengeManager.getNextEntry();
    byte[] challenge = challengeManager.getNextChallenge();
    CDPlayer cdp = (CDPlayer) remoteAppliance;
    RemoteApplianceReport rar = cdp.play(entry, challenge, track);
    /* 省略 */
  }

  public void stopCD()
  {
    /* 省略 */
  }

  /* 省略 */
}
```

図 5.18: CD プレーヤを利用するクライアントのサンプルコード

第 6 章

情報家電共有機構SCNAの評価

本章では，情報家電機器の共有機構SCNAの評価について述べる．評価する内容としては，SCNAを用いて構築されたシステムを動作させた時に行なわれる処理に要する時間の測定による定量的な評価と，情報家電機器の共有のために必要となる機能が満たされているのかの検証や，SCNAで対応可能・不可能な攻撃を検証することによる定性的な評価を行なう．

6.1 定量的評価

本節では，SCNA を用いて構築したシステム上で，ユーザクライアントが情報家電機器を操作する場合に，認証サーバおよび情報家電機器制御機構上で行なわれる処理に要する時間を測定し，それらの処理のスケーラビリティや，ボトルネックとなる部分について考察をする．

6.1.1 測定方法

測定には，認証サーバおよび情報家電機器制御機構に定期的に要求を発生するユーザクライアントプログラムを作成し，ユーザクライアントと認証サーバ間での通信のシミュレーションと，ユーザクライアントと情報家電機器制御機構間での通信のシミュレーションを行なった．

認証サーバ上では，ユーザクライアントからの要求を受けてから，それに応答するまでに行なわれる処理に要した時間を記録した．測定の際には，情報家電機器が10台あるとし，登録するユーザの数を10人から100人まで10人ずつ増やし，各段階において100回分のデータを収集した．

情報家電機器制御機構上では，ユーザクライアントがチケットの登録を要求してきた時，それを処理するまでの時間と，ユーザクライアントが情報家電機器の機能の利用を要求してきてから，その機能の実行が許可されるまでの時間を記録した．測定の際には，登録されたユーザが10人の場合と100人の場合について，それぞれ100回分のデータを収集した．

6.1.2 測定環境

測定に利用したマシンの性能は，表6.1に示す通りである．認証サーバおよび情報家電機器制御機構がこのマシン上で実行され，外部に用意された測定用ユーザクライアントプログラムから処理の要求を受ける．

表 6.1: 測定環境

項　目	環 境
CPU	AMD K6-III 450MHz
メモリ	128MB
OS	RedHat Linux 6.2
JDK version	Sun JDK 1.2.2_006 linux i386
暗号化ライブラリ	Sun JCE 1.2.1

6.1.3 測定結果

シミュレーションによって，認証サーバおよび情報家電機器制御機構上での処理に関して，以下のような結果が得られた．

認証サーバ内での処理

認証サーバに関する測定の結果を，図6.1に示す．グラフ中に示されるそれぞれの区間は，

区間A ユーザクライアントとの通信の開始および要求の受け付け

区間B セッション鍵の生成

区間C 鍵データベースの検索

区間D ACLデータベースの検索

区間E 返答メッセージの生成

での処理時間を100回測定した時の平均値を示している．

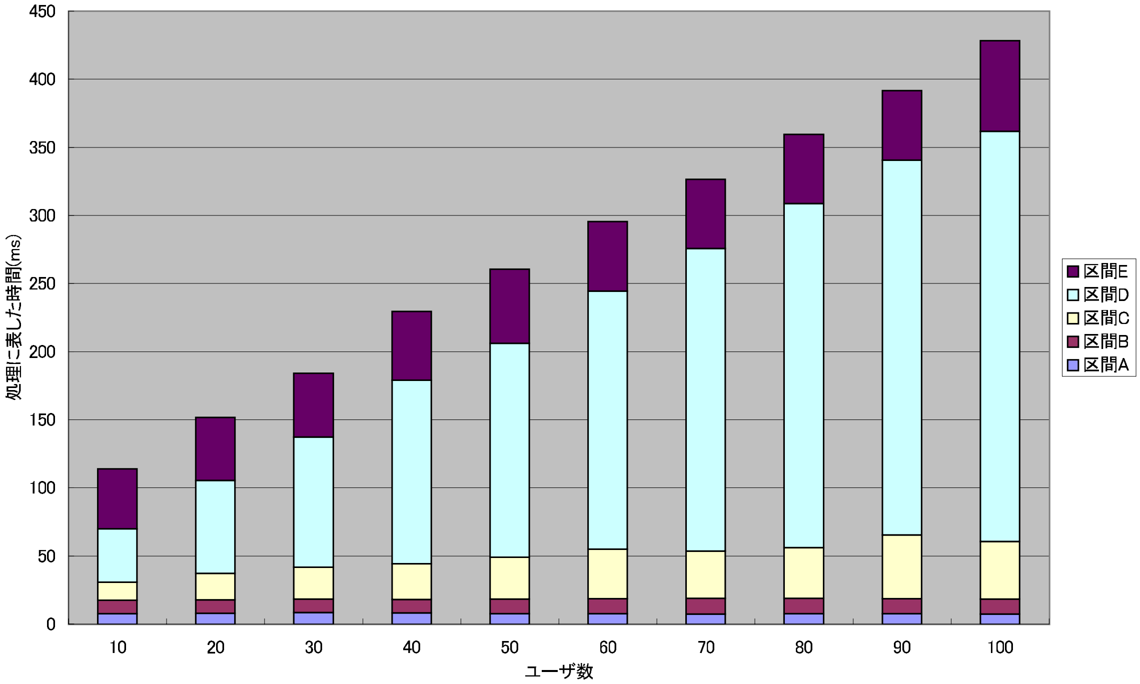


図 6.1: ユーザ数の変化に伴う認証サーバ内での処理時間の変化および内訳

情報家電機器制御機構内での処理

情報家電機器制御機構がチケットを受け付けた時の処理時間の測定結果を，表6.2に示す．グラフ中に示されるそれぞれの区間は，

区間A チケットの復号

区間B セッション鍵およびACLエントリの取得

での処理時間を100回測定した時の平均値を示している．

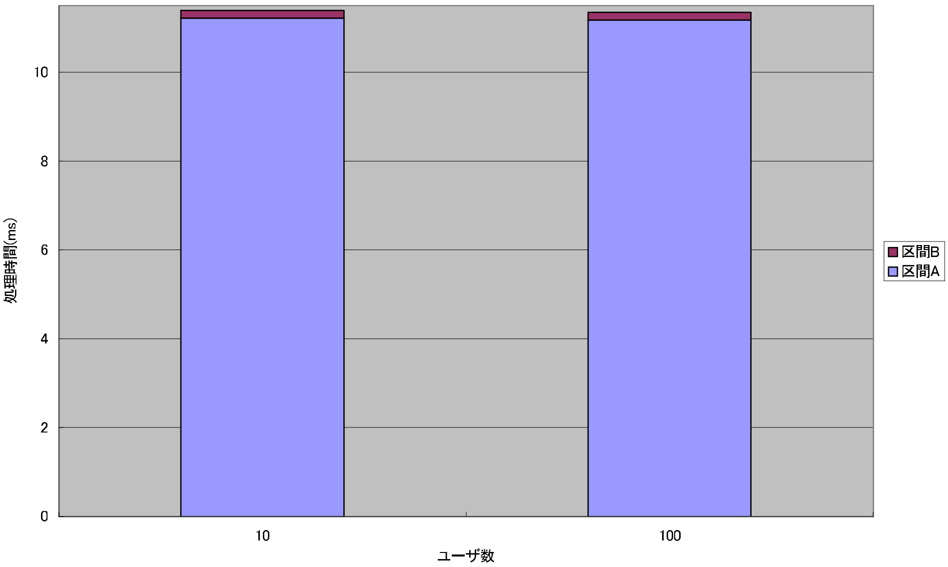


図 6.2: チケットの登録時の処理時間および内訳

情報家電機器制御機構が機能の実行要求を受け付けてから実行するまでの処理時間の測定結果を，表6.3に示す．グラフ中に示されるそれぞれの区間は，

区間A ユーザクライアントが暗号化したチャレンジの復号

区間B 復号したチャレンジの検証

区間C アクセス権の検証および次回チャレンジの生成

での処理時間を100回測定した時の平均値を示している．

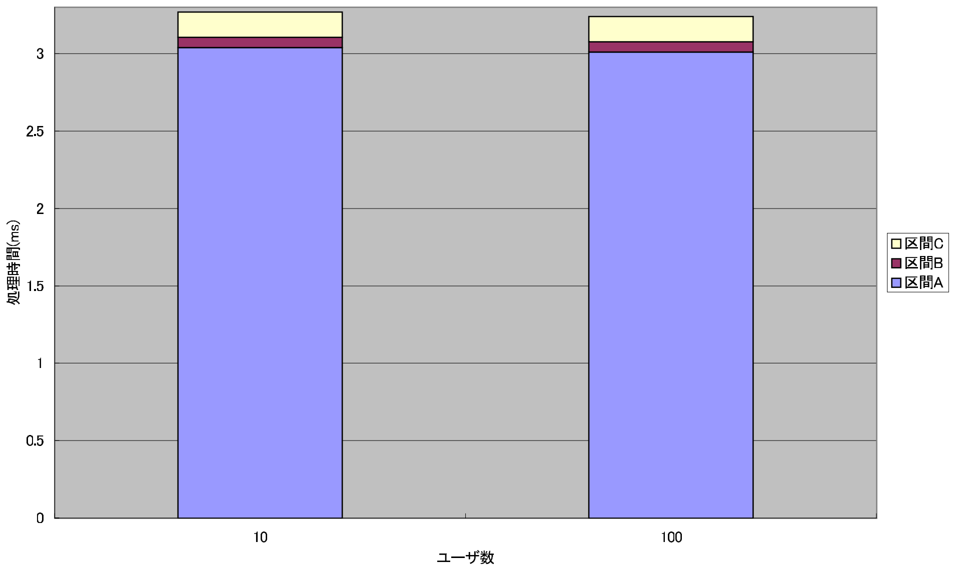


図 6.3: 認証時の処理時間および内訳

6.1.4 測定結果に対する考察

認証サーバ内での処理では，ユーザ数が増えるとともに，区間Cと区間Dでの処理時間が増加している．そして，区間Dでの処理時間は区間Cでの処理時間より増加しやすい．これは，区間Dで扱うACLのデータが，ユーザ数× 情報家電機器数のエントリを保持するデータベースから検索されるのに対し，区間Dで扱う親鍵のデータは，ユーザ数 + 情報家電機器数のエントリを保持するデータベースから検索されるためだと考えられる．

情報家電機器制御機構内でのチケットの受け付け処理では，チケットの復号に時間がかかっている．また，機能実行の検証の処理では，チャレンジの復号に時間がかかっている．これは，相対的にチケットの情報量の方がチャレンジの情報量より大きいからである．しかし，チケットの復号はチケットの登録時のみ行なわれるので，実際のシステムを運用する際には，チャレンジの復号の頻度よりもずっと少ない頻度でしか実行されないと考えられる．

また，認証サーバ内での処理は，ユーザ数の増加とともに負荷が高くなっているのに対し，情報家電機器制御機構内での処理は，登録するユーザの数を変えても特に変化を見せていない．これは，認証のためのACL管理や鍵管理の際に発生する処理の負荷を豊富な計算能力を持つ認証サーバに集中させることで，低い計算能力しか持ち合わせていない情報家電機器の計算負荷を軽減させる効果が現れていることを示している．

6.2 定性的評価

本研究では，情報家電機器の共有に必要な機能として，認証機能，アクセス管理機能，資源管理機能の三つを挙げている．本節では，SCNA がそれらの機能をどこまで満たされたのかを評価する．

また，SCNA によって対応が可能となった攻撃と，今のところ効果的な対応策が講じられていない攻撃について明らかにし，本システムの定性的な評価とする．

6.2.1 満たされた機能

認証機能

認証サーバが発行する認証チケットによって，ユーザクライアントと情報家電機器が安全にセッション鍵を共有し，暗号化を利用することで実現されている．また，セッション鍵を用いることで，情報家電機器上での秘密鍵管理の負荷が軽減されている．

アクセス管理機能

ACL を利用して実現されている．SCNA においては，認証チケットに ACL の情報の一部を含ませることで，情報家電機器上での ACL 管理の負荷を軽減している．

資源管理機能

ロック変数を用いた排他制御を行っている．その際には，ネットワーク上のトラブルに対処するために，リース時間の概念を利用して機能をロックする方法を提供している．

以上の内容から，SCNA が情報家電機器の共有機構としての有意性を持つことが分かる．

6.2.2 各種セキュリティ攻撃への対応

対応が可能な攻撃

- ユーザのなりすまし

  認証サーバに対して偽ったユーザ名を使用しても，ユーザそれぞれに対応する親鍵がない限り，得られた情報を復号して利用することは困難である．従って，他のユーザになりすましたユーザによる情報家電機器の不正利用に対して耐性があると言える．

- アクセス権限の書き換え

  ユーザが情報家電機器に対して持つアクセス権限の内容はチケットに含まれており，ユーザクライアントから情報家電機器上に送られるが，その内容は情報家電機器の親鍵で暗号化されているためユーザクライアント上で知ることは困難であり，故意に書

き換えられる可能性は低い．従って，アクセス権限の書き換えによる情報家電機器上の機能の不正利用に対して耐性があると言える．

- 同一メッセージのリプレイ

  情報家電機器がユーザクライアントを認証する際には毎回乱数を発生し，その暗号化結果をユーザクライアントに返答させ，それを復号した内容を元の乱数を比較することによって行っている．ここで利用される乱数は，一度認証に利用した後は必ず無効化されるので，同一のデータを送っても二度目以降には意味を持たない．従って，同一メッセージのリプレイによる情報家電機器の不正利用に対して耐性があると言える．

対応が不可能な攻撃

- DoS 攻撃

  処理しきれない程の多数の操作を情報家電機器に要求ことで，異常な負荷を意図的に加え，情報家電機器上のサービスが正常に動作するのを妨害するような攻撃として DoS 攻撃が挙げられる．特に，情報家電機器上では小規模なコンピュータが動作している可能性が高いため，この種の攻撃への耐性は低いと考えられる．

6.3　本章のまとめ

定量的な評価によって，認証サーバでは鍵および ACL の検索の負荷が増大しやすく，情報家電機器制御機構内での処理では，チケットおよびチャレンジの復号に要する処理時間が大部分を占めているということが分かった．また，ユーザ数の増加にともなう認証システムへの負荷が認証サーバに集中するということが示されたので，本研究の目的の一つである情報家電機器への計算負荷の軽減が実現されていることが立証された．また，SCNA が情報家電機器の共有機構として必要な機能を満たし，SCNA によって対応可能な攻撃と対応が不可能な攻撃を明らかにした．

第 7 章

結論

本章では，SCNA を情報家電機器上で動作するソフトウェアとして普及させる上で，追加すべき機能，改善すべき点を今後の課題として取り上げる．指摘する課題としては，複数の管理ドメイン間での相互運用性について，デッドロックの回避について，サービスの登録・検索機能の充実についての3点を取り上げる．そして，最後に本論文全体を振り返り，そのまとめとしての結論を述べる．

7.1 今後の課題

本論文執筆時において，SCNA が提供する機能は，情報家電機器を共有して利用するシステムを構築する際に必要となる基本的なものに留まっている．今後，SCNA が情報家電機器を使ったシステム構築の際に広く利用されるミドルウェアとなっていくことを目的とし，以下のような拡張機能や問題点の回避手段を追加していく予定である．

7.1.1 複数の管理ドメイン間での相互運用

現在の SCNA では，情報家電機器がユーザの認証を行なう場合には，その情報家電機器とユーザの双方の情報が，同一の認証サーバで管理されている必要がある．つまり，認証サーバが複数存在した場合，異なる認証サーバ上で管理されている情報家電機器とユーザの間では，認証を行なえず，対応する ACL エントリも存在しない．

このように，異なる認証サーバによって管理されている情報家電機器とユーザの間では通信できないという問題を解決するには，認証に関しては，複数存在する認証サーバ同士が，お互いに秘密鍵を共有する機能を持つという方法が考えられる．

また，アクセス管理に関しては，ユーザを特定せずに，グループ単位でパーミッションを設定できるようにし，不特定ユーザのための権限の少ないユーザグループを設定して利用する方法が考えられる．

7.1.2 デッドロックの回避

SCNA が提供する資源管理機能では，情報家電機器上の資源をロックし排他制御を行なっている．この場合，ユーザクライアント同士で，資源のロックの際の明確なプロトコルを決定しておかないと，デッドロックを引き起こす可能性がある．例えば，あるユーザクライアントが，必要となる全ての資源を確保しロックするまで，確保済みの資源をロックし続けるという設計になっていると，他のユーザクライアントと，お互いに必要とする資源のロックのし合いによって，双方がブロックした状態で止まってしまう．

このような，デッドロックを回避する方法としては，システム上に存在する全ての機器上の機能に対して一意の番号をふり，資源にロックをかける際には，必ずその番号が昇順または降順になるような順番で行なうと決めておく方法がある．本システムでは，認証サーバ上でシステム内の情報家電機器の情報を集中管理しているので，それらの持つ資源に対して一意な番号を管理するのは実現しやすい．

7.1.3 サービスの登録・検索機能の充実

現在の SCNA では，システム内にユーザや情報家電機器を組み込む際の処理として，必要な秘密鍵や ACL を認証サーバに登録する部分や，資源管理に関するポリシを設定する部分しか定義していない．しかし，情報家電機器の共有利用システムを考える際には，他にも有用と考えられる機能が存在する．例えば，システム上に新たな情報家電機器を登録す

る際に，プラグアンドプレイ感覚で作業が自動的に行える機能や，ユーザが要求に対応したサービスの検索できる機能などが挙げられる．

このような，サービスの登録や検索を実現する方法としては，認証サーバにJiniのLookupサービスの機能を導入する方法が考えられる．この場合，ユーザクライアントにとって，自分を認証できる認証サーバは既知なので，サービス利用者によるLookupサービスのDiscovery機能は使用する必要がない．また，情報家電機器の動的な追加や削除を行なうと，認証サーバ上のACLデータベースの管理が複雑になるため，ACL管理部分に改良を加える必要がある．

7.2　まとめ

本論文では，『セキュリティを考慮した情報家電機器の共有機構の開発』について述べた．情報家電機器は，既存の家電機器としての機能に加え，ネットワーク接続性を持つことで，ネットワークを介して共有利用という側面を強めてきた．しかし，その際に発生するユーザの認証，アクセス管理の問題や，資源の管理の問題を解決する手段として，今まで情報家電機器の共有に特化したシステムは存在しなかった．そこで，本論文では，情報家電の利用に特化した認証とアクセス管理のシステム，資源管理のシステムを実現した．認証，アクセス管理の際に重要な要素としては，情報家電機器上での負荷の軽減を指摘し，認証サーバによる認証チケットの発行による認証方法を提案した．また，情報家電機器上の資源管理方法としては，情報家電機器の機能に特化したロックのモデルを提案し，機能の排他制御を実現した．資源のロックの際には，ネットワーク上のトラブルに対処するためにリース時間を設定した資源管理を導入した．さらに本論文では，情報家電機器の共有システムの開発の際に利用できるAPIを提供し，その応用システムの構築例を示し，その処理性能に対する定量的な評価や，既存の関連技術との特徴の比較による定性的な評価を行ない，情報家電機器の共有利用に必要とされる機能が実現されたことを立証した．

謝辞

本研究を進めるにあたり，絶えず懇切丁寧な御指導を賜わりました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝いたします．

慶應義塾大学徳田・村井・楠本・中村研究会の諸先輩方には，お忙しい中，貴重な示唆や御助言、御指導を頂きました．特に慶應義塾大学政策・メディア研究科博士1年の中澤仁氏，大越匡氏，修士2年の藤村浩士氏，牧野聡氏，修士1年の岩井将行氏，楠本晶彦氏，原嶋章介氏，由良淳一氏には，本論文執筆にあたって絶えざる励ましと御指導を賜りました．

そして，KMSF(Keio Media Space Family-Architecture)研究グループの学部4年の青木崇行氏，石井かおり氏，桐原幸彦氏，土田泰徳氏，松宮健太氏および学部2，3年のメンバーには，本研究に関する様々な議論，助言をして頂きました．特に学部3年の柳原正氏には，素晴らしい研究環境の整備に助力して頂き，大変感謝しております．

また，MCng(Mobile Communication Next Generation)研究グループの学部4年の梅染充男氏，村瀬正名氏，若山史郎氏，ECN(Engineering approach to Computer Networks)研究グループの学部4年の間博人氏にも，同輩として大変お世話になりました．

みなさまに，深い感謝の意を表します．

平成13年 2月 19日
慶應義塾大学 環境情報学部 4年
杉田 洋介

参考文献

[1] 経済社会総合研究所: 消費動向調査(四半期) (2000). http://www.esri.cao.go.jp/jp/stat/menu.html.

[2] the Institute of Electrical and Electronics Engineers, Inc.(IEEE).: 802.11 Wireless Standards Package (1999). http://standards.ieee.org/catalog/IEEE802.11.html.

[3] the Institute of Electrical and Electronics Engineers, Inc.(IEEE).: IEEE Std 1394-1995 - Description (1995). http://standards.ieee.org/reading/ieee/std_public/description/busarch/1394-1995_desc.html.

[4] Bluetooth Special Interest Group: Bluetooth Specification, Version 1.0B (1999). http://www.bluetooth.com/developer/specification/specification.asp.

[5] Interfrared Data Association: IrDA Serial Interfrared Data Link Standard Specifications. http://www.irda.org/standards/specifications.asp.

[6] 東日本電信電話株式会社: INSネットサービスのインターフェース. http://www.ntt-east.co.jp/ISDN/tech/spec/topinx.html.

[7] Asymmetric Digital Subscriber Line Forum: ADSL Forum System Reference Model (1996). http://www.adsl.com/Technical_Reports/TR-001.pdf.

[8] 日本CATV技術協会 規格・標準化委員会: デジタルケーブルテレビの標準規格. http://www.catv.or.jp/JCTEA/STD/kikaku616.HTM.

[9] JSAT株式会社: 通信・放送サービス一覧表. http://www.iijnet.or.jp/JSAT/site/corporate/service_list.html.

[10] Sun Microsystems, Inc.: Java 2 Platform, Micro Edition (J2ME) (2000). http://java.sun.com/j2me/index.html.

[11] Sun Microsystems, Inc.: CLDC and the K Virtual Machine (KVM) (2000). http://java.sun.com/products/cldc/.

[12] Sun Microsystems, Inc.: EmbeddedJava Application Environment Specification (1999). http://java.sun.com/products/embeddedjava/spec/ejae-spec.pdf.

[13] Sun Microsystems, Inc.: Java Card 2.1.1 Platform Documentation (2000). http://java.sun.com/products/javacard/javacard21.html.

[14] 横河電機株式会社: Compact Field Server DUONUS (2000). http://www.yokogawa.co.jp/DUONUS/Product/Product.htm.

[15] Dallas Semiconductor Corp.: Introducing TINI: Tiny InterNet Interface (2000). http://www.ibutton.com/TINI/index.html.

[16] Sun Microsystems, Inc.: Jini Architecture Specification Version1.1 (2000). http://www.sun.com/jini/specs/jini1_1.pdf.

[17] OMG (Object Management Group): Common Object Request Broker Architecture and Specification (2000).
http://www.omg.org/technology/documents/formal/corbaiiop.htm.

[18] Sony and Matsushita and Philips and Thomson and Hitachi and Toshiba and Sharp and Grundig: Specification of the Home Audio/Video Interoperability (HAVi) Architecture (2000). http://www.havi.org/techinfo/docs/HAVi10.pdf.

[19] ECHONT CONSORTIUM: The ECHONET Specification Version 1.0, ECHONET 規格書 (2000). http://www.echonet.gr.jp/spec/spec_v10.htm.

[20] Dallas Semiconductor Corp.: Java-Powered Ring Fact Sheet. (2000). http://www.ibutton.com/store/jringfacts.html.

[21] Diffie, W., and Hellman, M .E.: A critique of the proposed Data Encryption Standard (1976). Communication of the ACM.

[22] Lai, X., Massey, J. L., and Murphy, S.: Markov Chiphers and Differential Cryptanalysis (1991). In Advances in Cryptography – EUROCRYPTO '91.

[23] Rivest, R. L., Shamir, A., and Adleman, L.: A method for obtaining digital sigunatures and public-key crypto-system (1978). Communication of the ACM.

[24] Schnorr, C. P.: Comment on DSA: Comparison of the Digital Signature Algorithm and the Signature schemes of ElGamal and Schnorr (1991). Letter to the Director of CSL/NIST.

[25] Kohl, C.Neuman, J.Steiner: The Kerberos Networked Authentication Service. (1992). MIT Ahtena Project.

[26] S.Kent, R.Atkinson: Security Architechture for the Internet Protocol (1994). RFC 2401.

[27] Sun Microsystems, Inc: Java Remote Method Invocation Specification (1999). http://java.sun.com/j2se/1.3/docs/guide/rmi/spec/rmi-title.html.