電子化知的財産・社会基盤 19－6
（２００３．２．２２）

携帯電話向け Java 実行環境を用いた著作権管理システム

仁野 裕一† 中村 暢達† 田口 大悟† 谷 幹也†

携帯電話向けのコンテンツ配信サービスは、利用者／サービス提供者の拡大を続けているが、課金方法の種類は限られており、より柔軟な課金形態が求められている。著者らは、携帯電話向け Java 実行環境上で多様な課金方法を実現する著作権管理システム－モバイル RightsShell®(MRS)を構築した。本システムは、クライアントライブラリサイズを 11Kbyte 程度に軽量化することで、携帯電話 Java 実行環境上で、(1)音楽・待ち受け画像などのデータ型コンテンツおよびゲームなどのプログラム型コンテンツに対する著作権管理、(2)利用回数/利用時間/利用期限を組み合わせたきめ細かな利用制御、(3)プログラム型コンテンツの任意の箇所での利用制御を実現している。

A Digital Rights Management System on Java Application Environment for Cellular Phone

Yuichi NINO†, Nobutatsu NAKAMURA†, Daigo TAGUCHI† and Mikiya TANI†

Digital contents distribution services are widely spreading. However, these services have limited billing methods. We developed a Digital Rights Management (DRM) system called “Mobile RightsShell® (MRS)” that enables various settlement ways. MRS library is small (11Kbyte) enough to run on the Java platforms for cellular phones. The features of this system are as follows: (1) DRM for various contents (games etc.), (2) flexible usage control (pay-per-use/pay-per-second) and (3) easy programming interface for applying usage control.

１．はじめに

近年、デジタルコンテンツの利用が、PC から携帯電話へ拡大されるのに伴い、携帯電話においても著作権管理(DRM: Digital Rights Management)の重要性が高まっている。

現在、日本での携帯電話向けコンテンツ配信サービスは、i モード®[1]端末に見られるように、配信されたコンテンツが携帯電話の外部に出ないことにより、著作権保護を行っているのが一般的である。さらに、管理上の問題から、このようなサービスは月額 100 円～300 円支払う会員制か、ケータイ de ミュージック[2][3]など音楽配信に多く見られる 1 曲あたり 300 円程度の費用を支払うダウンロード課金のいずれかに課金方法が制限されている。

このようなサービスは、利用者･サービス提供者数が増加している一方で、「ダウンロードしたコンテンツがその端末でしか利用できない」・「月に何回も利用するわけではないので会員登録料、高いダウンロード料を払う価値がない」などの理由により、利用者１人あたりのサービス利用回数が伸び悩んでいる[4]。したがって、異なった端末でもコンテンツを利用できるように超流通[5]を可能にし、一回の利用額を少なくするために Pay per View などの機能を提供可能な DRM システムを携帯電話上で実現することで、これらの不満を解消することができる。そして、利用者数のさらなる増加、利用者１人あたりの利用回数の増加による販売増の効果を期待できる。

著者らは、上記の機能を実現する DRM システムとしてモバイル RightsShell®(MRS)システム[6][7][8]を開発した。MRS は、パソコン向けの DRM システム-RightsShell® [9][10]を携帯端末で動作するようにプログラムサイズを縮小し、DRM 機能を携帯端末に最適化したシステムである。

MRS は、携帯電話上のアプリケーション配信サービスを視野に入れ、携帯電話に近いメモリ・CPU 能力を持つ PalmOS で開発を行い[5][6][7]、携帯電話向けの Java アプリケーション配信サービス開始に伴って、これをターゲットとして研究開発を行った。Java アプリケーションの著作権保護には、①Java アプリケーションの実行管理を行う Java Application Manager (JAM) の拡張、②Java API の拡張、③JavaVM より低位の OS の拡張、④JavaVM 上での実装の 4 通りが考えられるが、①②③は携帯電話の仕様を決定しているキャリア・携帯電話メーカによる機能拡張が必要であるため、これを必要としない④の方式での DRM システム構築について検討した。

本稿では、初めに携帯電話向け Java 実行環境とそれを搭載している携帯電話端末の特徴について説明し、メモリ・記憶容量などの制限が厳しい JavaVM 上での DRM クライアント実装方式について検討する。つぎに、この検討をもとに開発した携帯電話 Java 版 MRS システムの詳細、動作例について記す。

† NEC インターネットシステム研究所
Internet Systems Research Laboratories, NEC Corporation

2．デジタル著作権管理(DRM)システム概要

2.1　DRM システムの種類

DRM システムは、それを実現する技術の観点から、①電子透かし応用型、②パッケージコンテンツ保護型、③デジタル権利管理型の 3 つに分類できる(表１)。

著者らは、携帯電話へ格納されるコンテンツの利用回数や利用時間などの制御を実現するシステム開発を目指しており、以下デジタル権利管理型の DRM システムについて説明する。

表 1 DRM システムの種類

種類	特徴	例
電子透かし応用型	コンテンツ内に著作権情報を埋め込み、不正コピーを追跡することによって、コンテンツの不正コピーを抑止する。	CPTWG, MarkAny, エム研, cIDf
ﾊﾟｯｹｰｼﾞｺﾝﾃﾝﾂ保護型	不正アクセスのできない耐タンパデバイスにコンテンツを格納し、コンテンツの不正コピーを防止する。	DVD-RAM/R/RW CSS,CPPM,CPRM, DTCP,SDMI, SDCard(Panasonic), MagicGate™(Sony)*
ﾃﾞｼﾞﾀﾙ権利管理型	マシンのプロセス監視・コンテンツの暗号化により、コンテンツのコピー・改竄を防止し、さらにコンテンツの利用回数・利用時間などの利用制御を行う。	WMT®(Microsoft)* Rights|System™(Intertrust)*, EMMS(IBM), RightsShell®(NEC)*, SecurePackage (LockStream), Helix®(RealNetworks)*

*表中の™,®はそれぞれ括弧内の会社の商標,登録商標である

2.2　デジタル権利管理型 DRM システムの構成

デジタル権利管理型の DRM システムは、一般的に図１に示すように、コンテンツパッケージャ、コンテンツダウンロードサーバ/ライセンスサーバ(以下、合わせて DRM サーバ)、コンテンツプレイヤ(以下、DRM クライアント)から構成される。

コンテンツパッケージャは、コンテンツの著作権情報の埋め込みや暗号化(パッケージ化)を実行する。

コンテンツダウンロードサーバは、パッケージ化したコンテンツを Web サーバに登録するフロントエンドを提供し、単にサーバとしてファイルに置くだけでなく、商品ページの更新や顧客情報との関連付けを自動的に行う。

ライセンスサーバは、ユーザ認証・DRM クライアント認証･DRM クライアントをインストールした端末の認証を行った後、コンテンツの利用条件やコンテンツの復号鍵を格納した情報(以下、チケット)を発行し、DRM クライアントに送信する。このときのライセンスサーバ・DRM クライアント間の通信は、認証情報やチケットの情報が外部に漏洩しないように、SSL などのセキュア通信により行われる。

DRM クライアントは、受信したチケットに含まれる復号鍵でコンテンツを復号し、チケットに記載された利用条件をもとにコンテンツの利用を制御する。また、コンテンツの不正なコピー・改竄を防止するためのアクセス制御を行う。

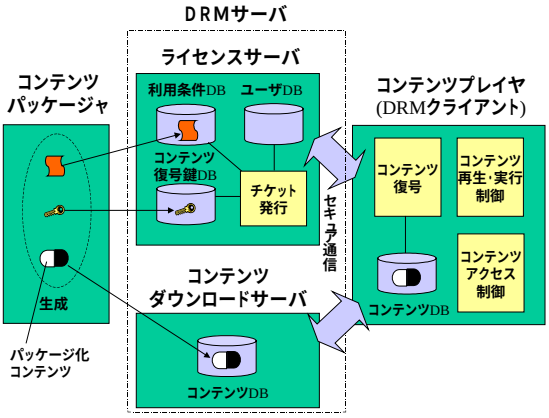


図 1 デジタル権利管理型の DRM システムの一般的な構成

2.3　デジタル権利管理型 DRM システムの機能

デジタル権利管理型 DRM システムでは、表 2 に示すような脅威(表中でビジネス上、著作権管理上必ず防ぐべき脅威を網掛けで示す)からコンテンツの利用を守る必要があり、表 3 ではこれらを解決するために DRM クライアントに実装できる機能をまとめている。これらの機能は、DRM システムの設計により、DRM クライアントあるいは DRM サーバのいずれかのコンポーネント、もしくは両方で実装される可能性がある。また、DRM クライアントを携帯電話に実装する場合、携帯電話の機能制限を利用して、DRM 機能を実現することができる。

3．携帯電話向け Java 端末上の DRM 実装

DRM クライアントの実装には、実装箇所、実装方式、各方式の実装機能の 3 点を検討する必要がある。携帯電話は製品の性質上、仕様を決定しているキャリア・携帯電話メーカに合意を取らない限り拡張が許可されない箇所が多いが、逆に決定している仕様をうまく利用することで、クライアントで実装すべき機能を少なくすることが可能である。もちろん、PC に比べて機能制限が大きいため、DRM クライアントの実装方式が限定される。さらに、携帯電話は処理能力が低く、メモリ容量も少ないため、DRM クライアントの機能とそれを実現するプログラム容量は最小限にすることが望ましい。そのためには、①DRM サーバで実現できる機能は DRM サーバで実装する、②携帯電話の機能制限を利用して DRM クライアント機能を実現する、③携帯電話アプリケーション実行環境の機能を極力活用することが重要である。

本章では、携帯電話向け Java 端末のアーキテクチャの特徴を示した上で、DRM クライアントの実装箇所について論じ、実行環境・携帯電話の機能的制限を示した上で、実現可能な実装方式と各方式の DRM クライアントの実装方法について述べる。

3.1　携帯電話向け Java 端末アーキテクチャの特徴

携帯電話向け Java 端末アーキテクチャは、Sun Microsystems が規定する携帯電話向け Java のプロファイル(UI やアプリケーションを定義する仕様)である Connected Limited Device Configuration (CLDC)[11] +

表 2 デジタル権利管理型 DRM システムの脅威とその対抗手段

利用場面	想定される脅威	対抗手段
ｺﾝﾃﾝﾂ ﾀﾞｳﾝﾛｰﾄﾞ 時	ｺﾝﾃﾝﾂの盗聴、ｺﾝﾃﾝﾂの受取否認、再送要求	ｾｷｭｱ通信(秘匿通信、到達確認)
	利用者のなりすまし	利用者認証
	利用端末のなりすまし	端末認証
ﾁｹｯﾄ 購入時	ﾁｹｯﾄ・ｺﾝﾃﾝﾂ復号鍵の盗聴、ﾁｹｯﾄの受取否認、再送要求、支払い額のごまかし	ｾｷｭｱ通信(秘匿通信、到達確認)、ﾄﾗﾝｽﾞｱｸｼｮﾝﾛｸﾞ管理(*)
	利用者のなりすまし	利用者認証
	利用端末のなりすまし	端末認証
ｺﾝﾃﾝﾂ、 ﾁｹｯﾄ 格納時	ｺﾝﾃﾝﾂの改竄	ﾀﾞｳﾝﾛｰﾄﾞｺﾝﾃﾝﾂへのｱｸｾｽ制限、ｺﾝﾃﾝﾂの暗号化
	ﾁｹｯﾄの改竄、ｺﾋﾟｰ、再利用	ﾁｹｯﾄへのｱｸｾｽ制限、ﾁｹｯﾄの暗号化
	ｺﾝﾃﾝﾂの権利外のｺﾋﾟｰ、二次利用	ｺﾝﾃﾝﾂの権利付与対象(端末、利用者、外部記憶)に応じた暗号化
ｺﾝﾃﾝﾂ 利用時	ｺﾝﾃﾝﾂ利用時に復号したﾃﾞｰﾀを盗み、再配布	再生/実行用ﾃﾞｰﾀへのｱｸｾｽ制限、端末ｼｽﾃﾑ全体の状態管理
	端末の紛失・盗難時の権利外利用者の利用	利用者認証、ﾁｹｯﾄの利用者に応じた暗号化

(*)この処理は、DRM サーバで実装

表 3 DRM クライアントに実装できる機能

利用者認証	利用者本人認証をﾊﾟｽﾜｰﾄﾞ、PKI、ﾊﾞｲｵﾒﾄﾘｸｽ等で行う。
ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時の本人認証。
ﾁｹｯﾄ購入時	ﾁｹｯﾄ購入時の本人認証。
ｺﾝﾃﾝﾂ利用時	ｺﾝﾃﾝﾂ再生/実行時の本人認証。
端末認証	利用者端末認証を証明書等で行う。
ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時の端末認証。
ﾁｹｯﾄ購入時	ﾁｹｯﾄ購入時の端末認証。
ｾｷｭｱ通信	通信ﾃﾞｰﾀを暗号化あるいはｽｸﾗﾝﾌﾞﾙ化し通信経路での保護を実現する。さらにｺﾝﾃﾝﾂ/ﾁｹｯﾄ送付後の送達確認もｻｰﾊﾞに送信する。
ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時のｾｷｭｱ通信。
ﾁｹｯﾄ購入時	ﾁｹｯﾄ購入時のｾｷｭｱ通信。
ｱｸｾｽ制限	ｺﾝﾃﾝﾂを再生するﾌﾟﾚｲﾔ、DRMｼｽﾃﾑ以外のﾌﾟﾛｸﾞﾗﾑが端末上のﾒﾓﾘやﾃﾞﾊﾞｲｽにｱｸｾｽすることを禁止（制限）する。
ﾀﾞｳﾝﾛｰﾄﾞｺﾝﾃﾝﾂ	ﾀﾞｳﾝﾛｰﾄﾞしたｺﾝﾃﾝﾂへのｱｸｾｽを禁止(制限)する。
再生/実行用ﾃﾞｰﾀ	ｺﾝﾃﾝﾂ再生/実行用にﾒﾓﾘ上に一時的に復号されたｺﾝﾃﾝﾂへのｱｸｾｽを禁止（制限）する。
ﾁｹｯﾄ	起動管理[後述]を行う際の利用条件などを記載したﾁｹｯﾄへのｱｸｾｽを禁止（制限）する。
起動管理	ﾌﾟﾛｸﾞﾗﾑが起動時もしくは実行時に、利用回数、利用時間などの条件により、その実行が制御される。ﾀﾞｳﾝﾛｰﾄﾞしたｺﾝﾃﾝﾂの Pay-per-Play、Pay-per-Second などを実現。
状態管理	端末ｼｽﾃﾑ全体（他ﾌﾟﾛｸﾞﾗﾑの実行状態、通信状態）などにより、ｺﾝﾃﾝﾂの再生/実行が制御される。
暗号処理	改竄、不正利用の防止を目的としたｺﾝﾃﾝﾂ、ﾁｹｯﾄの暗号化を意味する。
端末依存	端末に含まれる暗号鍵を使った暗号処理によるｺﾝﾃﾝﾂあるいはﾁｹｯﾄに対するｺﾋﾟｰﾌﾟﾛﾃｸｼｮﾝ。ｺﾝﾃﾝﾂの利用が特定の端末に制限される。
利用者依存	利用者認証を基にした暗号鍵を使った暗号処理によるｺﾝﾃﾝﾂあるいはﾁｹｯﾄに対するｺﾋﾟｰﾌﾟﾛﾃｸｼｮﾝ。ｺﾝﾃﾝﾂの利用が特定の利用者に制限される。
外部記憶依存	外部記憶に含まれる暗号鍵を使った暗号処理によるｺﾝﾃﾝﾂあるいはﾁｹｯﾄに対するｺﾋﾟｰﾌﾟﾛﾃｸｼｮﾝ。
ｺﾝﾃﾝﾂ依存	個々のｺﾝﾃﾝﾂに割り当てられた暗号鍵を使った暗号処理によるｺﾝﾃﾝﾂに対するｺﾋﾟｰﾌﾟﾛﾃｸｼｮﾝ。

Mobile Information Device Profile (MIDP)[12]が標準となっている。このアーキテクチャを図 2 に示す。

Java Application Manager(JAM)は、Java アプリケーションのダウンロード、インストール、検査、起動、およびアンインストールといったアプリケーション管理を行う携帯電話上のコンポーネントである。

CLDC は小型でリソース制約のあるデバイスに対する Java のコア API と Java Virtual Machine(KVM)を定義する。MIDP は、Sun が定めた CLDC 上の携帯電話 UI に関する標準 API(以下、共通 Profile)である。CLDC/MIDP では、この標準 API に対して、キャリア/携帯電話メーカの独自拡張を許している。これを OEM Specification Profile(以下、独自 Profile)と呼ぶ。これは、KDDI(AU)の場合 KDDI-P[13]、J フォンの場合 J-Phone Specific Class Libraries(JSCL)[14]と呼ばれる。

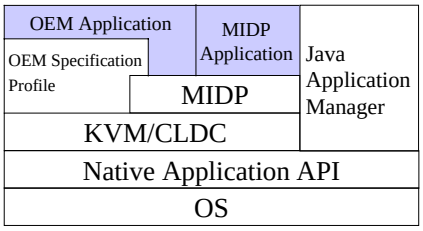


図 2 MIDP 実行環境のアーキテクチャ

一方、NTT ドコモは、CLDC/MIDP フレームワークを採用しておらず、独自の Java 実行環境-Doja™-[15]を提供している。Doja™は、図 3 のように、CLDC 上

に作られた MIDP と異なる機能をもった Doja™プロファイルが実装されている。MIDP と Doja™は互換性がないので、日本において携帯電話向け Java アプリケーションを開発する際には、MIDP と Doja™の２つのプロファイルを考慮に入れなければならない。

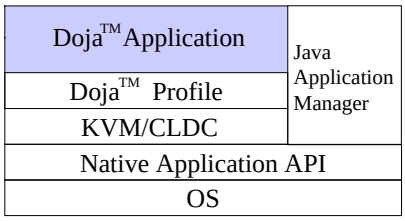


図 3 Doja™**実行環境のアーキテクチャ**

3.2 携帯電話向け Java 端末における DRM クライアントの実装箇所とその実現可能性

携帯電話向け Java 端末が 3.1 節に示したアーキテクチャを持つことを考慮すると、図 4 に示した 4 つの箇所で DRM クライアントの実装が可能である。各箇所での実装方法を以下に示す。

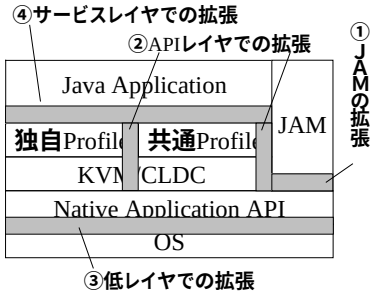


図 4 DRM **クライアントの実装可能箇**

① JAM を拡張し、JAM の中に DRM クライアントの機能を組み込む
② 独自 Profile あるいは共通 Profile と KVM を拡張し、そこに DRM クライアントの機能を組み込む。
③ OS や Native Application API に、コンテンツに関わるファイル処理が行なわれた際にそれをフックし、DRM クライアントの機能を呼び出す。
④ Java アプリケーション(図 2,図 3 の網掛け部分)の中に DRM クライアントに関わる機能を組み込む。

このうち、①③は、キャリアが仕様を決めた後、携帯電話メーカが個別に実装するものであるので、特定の機能を実現するために拡張することは極めて困難である。また、②の共通プロファイルの拡張は、JAVA の標準化の一項目として拡張の提案から仕様化が必要であり、独自プロファイルの拡張に関しては、端末仕様を決定するキャリア、携帯電話メーカの合意が必要であるため、普及活動には時間がかかる。

そこで、著者らは、④のように Java アプリケーションに DRM クライアント機能を組み込む方法を採用した。ただし、この方法を採用したことにより、DRM クライアントの機能が携帯電話向け Java VM 機能で実現できる枠内に制限される。

3.3 携帯電話向け Java API の特徴

2003 年 1 月現在、MIDP は version1.0 の SDK、Doja™は 503i+FOMA®[1]向け、504i 向けの 2 種類の SDK がリリースされている。また、MIDP の version2.0[16]の仕様が 2002 年 11 月にリリースされた。表 4 にこれらの各仕様について DRM クライアント実装時に重要となる機能をまとめた。

この機能制限の上で、携帯電話向け Java VM 上で DRM クライアントを実装する際には、以下に述べる点を留意する必要がある。

表 4 携帯電話向け Java **実行環境の機能比較**

	MIDP		Doja™	
	1.0	2.0	503i	504i
暗号化 Jarﾌｧｲﾙの実行	×	×	×	×
ﾀｽｸ管理	ｼﾝｸﾞﾙﾀｽｸ	ｼﾝｸﾞﾙﾀｽｸ	ｼﾝｸﾞﾙﾀｽｸ	ｼﾝｸﾞﾙﾀｽｸ
HTTP 通信	○	○	○	○
HTTPS 通信	×	○	○	○
HTTPｱｸｾｽ先の制限	無制限	無制限	ﾀﾞｳﾝﾛｰﾄﾞ元	ﾀﾞｳﾝﾛｰﾄﾞ元
ｽﾄﾚｰｼﾞ管理	○	○	○	○
共有ｽﾄﾚｰｼﾞ	×	○	×	×
ﾈｰﾃｨﾌﾞｱﾌﾟﾘ呼出	×	○	×	○
他の Javaｱﾌﾟﾘ呼出	×	×	×	×
外部通信	×	×	×	○
耐ﾀﾝﾊﾟﾃﾞﾊﾞｲｽとの接続	×	×	×	×
認証用ﾃﾞﾊﾞｲｽとの接続	×	×	×	×
端末 ID の取得	×	×	×	×

A. DRM クライアントの実装方式に関わる制限

1. 暗号化 Jar ファイルを Java VM が実行できないため、Java アプリケーションをコンテンツとする場合、コンテンツを暗号化した形で配布できない。
2. シングルタスクであり、なおかつ携帯電話 Java アプリケーションが他の携帯電話 Java アプリケーションを呼び出すことができないため、DRM クライアント機能とコンテンツを再生/実行する機能を別々のアプリケーションに実装しても連携させることが出来ない。したがってこれら２つの機能は、１つのアプリケーションとして実装しなければならない。

B. DRM クライアント機能の実装に関わる制限

1. MIDP version1.0 は HTTPS 通信のライブラリがないため、MIDP での実装を行う場合は暗号化通信のプロトコルを実装しなければならない。
2. Doja™はアプリケーションダウンロード元としか HTTP/HTTPS 通信ができないため、DRM サーバ機能はすべてアプリケーションダウンロード元に実装しておかなければならない。
3. 認証用デバイスとの接続インタフェースがないため、ユーザ認証はパスワードなどの個人しか

[1] FOMA は NTT ドコモ社の登録商標

知らないデータを数字キーなどで入力する必要がある。

4. 耐タンパデバイスとの接続インタフェースがないので、チケットやコンテンツを耐タンパデバイスに格納することはできない。
5. オフラインでアプリケーションから端末 ID を取得するインタフェースがないので、DRM クライアント機能実現の上で必要なデータ（暗号化コンテンツ、チケットなど）を端末個別に暗号化することができない。

C. DRM クライアント機能として利用できる機能

1. ストレージ管理機能により、DRM クライアント機能を有するアプリケーションが管理するデータは、携帯電話 Java 上の他のアプリケーションからアクセスできない。

3.4 携帯電話向け Java 端末の低レイヤ 機能の特徴

上記の携帯電話向け Java VM の仕様以外に DRM クライアント機能を決める重要な要件として、携帯電話の低レイヤ（OS など）機能の特徴がある。携帯電話向け Java VM で機能制限を行っていても、それより低レイヤのアーキテクチャや低レイヤ上で動作するアプリケーションの機能如何では、Java VM の機能制限が有効に働かない場合もあるからである。

このアーキテクチャや低レイヤ上で動作するアプリケーションは、携帯電話端末メーカによって独自に開発されており、その仕様・機能は一般に外部に公開されていない。ところが、一部の機能には公開されているものもあり、その中に DRM クライアント機能の実装に関わる重要な機能がある。それは、携帯電話内部データを取得するインタフェースである。

日本の携帯電話では、携帯電話 Java 実行コードや Java アプリケーションの管理データを取得するインタフェースは公開されていないため、これらのデータの取得を一般ユーザが行うことができない。この場合、携帯電話 Java 実行コードや Java アプリケーションの管理データに関するアクセス制限機能を DRM クライアントに実装する必要はない。

ところが、海外では一般に個人端末と PC 等のデータ交換は許可されている場合が多く、Java のダウンロードを行える Nokia7650 等では、実際に Java アプリケーションや管理データを PC と交換可能である。この場合、実行コードや管理データは外部に取得されることを前提としたアクセス制限機能を DRM クライアントに実装しなければならない。

表 5 に取得制限が端末毎にどの程度行われているかをまとめたものを示す。

表 5 携帯電話の内部データの取得に対する制限

	Java 実行ｺｰﾄﾞ	Javaｱﾌﾟﾘ管理ﾃﾞｰﾀ
ﾄﾞｺﾓ(iﾓｰﾄﾞ®)端末	×	△(*1)
Au(ezplus)端末	×	△(*1)
J-Phone 端末	×	△(*1)
外部取得可能端末 (例 Nokia 7650)	○	○

(*1)Java アプリケーションに外部に送信するコードがない限り、管理ストレージ内のデータの取出は不可。

3.3 携帯電話向け Java アプリケーション組み込み型の DRM クライアント実装方式

携帯電話向け Java アプリケーション組み込み型の DRM クライアント実装方式を決めるにあたり、3.3 節 A.で述べたことを念頭におけば、以下の 2 つの実装方式が考えられる。1 つは音楽データや映像データを再生するプレイヤに DRM クライアント機能を組み込むプレイヤ型、もう 1 つはゲームコンテンツなど配信するプログラムに DRM クライアント機能を組み込むライブラリ型である。

なお以降では、3.3 節の A.1 の制限より、Java アプリケーションは暗号化して配布できないので、Java 実行コードと Java アプリケーションが管理するデータが外部に取得されない携帯電話上で実装することを前提とする。

3.5.1 プレイヤ型実装方式

プレイヤ型とは、映像・音楽などのデータ型コンテンツを再生するプレイヤに全ての DRM クライアント機能を実装する方式である。この実装方式を図 5 に示す。

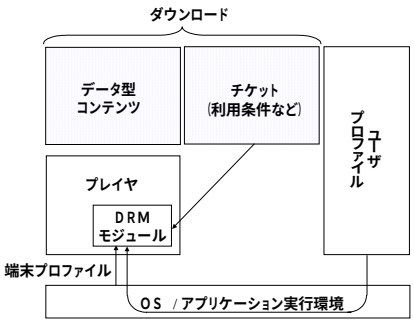


図 5 プレイヤ型実装方式

図中のプレイヤは、データ型コンテンツのプレイヤである。プレイヤは、データ型コンテンツをコンテンツダウンロードサーバから、チケットをライセンスサーバからダウンロードする。これらのデータは、表 4 に記載されているストレージ管理機能によって、他の Java アプリケーションからこれを利用することができないので、暗号化されてなくても構わない。

プレイヤ中の DRM モジュールは、ユーザプロファイルのユーザ認証情報・端末プロファイルの端末情報・チケットの利用条件を必要に応じて参照し、ユーザ・端末・利用条件が有効であるかチェックする。そして、必要な条件が全て有効であれば、プレイヤにコンテンツの再生を許可し、チケットの利用条件に合わせてプレイヤの実行を制御する。

3.5.2 ライブラリ型実装方式

ライブラリ型とは、ゲームなどのプログラム型のコンテンツに対し、プログラムソースを改変し、DRM クライアント機能を実装する方式である。この方式は、コンテンツを Java VM 上でそのまま実行させることを前提とするため、コンテンツの暗号化を行えない。この実装方式を図 6 に示す。

本方式の場合、DRM ライブラリが組み込まれたプログラム型コンテンツは、コンテンツダウンロードサーバから携帯端末自体の機能(通常は JAM)によって、ダウンロードされる。また、チケットは、プログラム内の DRM ライブラリによって、ライセンスサーバからダウンロードされる。

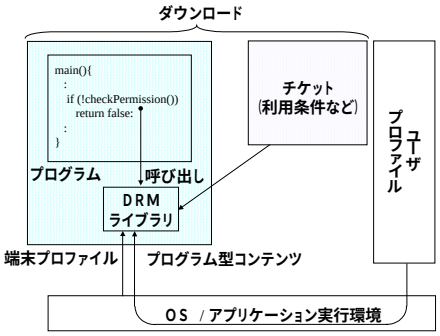


図 6 ライブラリ型実装方式

図中のプログラムは、コンテンツの実行の際に DRM ライブラリを呼び出すコードを含むプログラムである。この DRM ライブラリは、ユーザプロファイルのユーザ認証情報・端末プロファイルの端末情報、チケットの利用条件を必要に応じて参照し、ユーザ・端末・利用条件が有効であるかチェックする。そして、必要な条件が全て有効であれば、プログラムの実行を許可し、チケットの利用条件に合わせてプログラムの実行を制御する。

3.6 各実装方式における DRM クライアント機能の実装

つぎに、携帯電話は内部データを取得できない端末を利用することととし、携帯電話向け Java 実行環境で前節の 2 つの方式で実装した場合の①必須機能(表 2 の網掛けの脅威に対抗)、②実装可能な機能(それ以外の脅威に対抗 or 他の機能で脅威に対抗できているもの)、③Java VM の機能を利用することにより実装不要な機能について説明する。プレイヤ型とライブラリ型の 2 つの実装方式において、表 3 に示した機能が上記①～③のどれにあたるかを表 6 に示す。

表 6 で示したように、アクセス制限機能は端末自体の機能として実現されているので、DRM クライアントに実装する必要がなく、プログラム容量を削減できることが分かる。また、実装可能機能については、セキュリティを高める観点からは実装することが望ましいが、DRM サーバで機能の大半を実装し、DRM クライアントはできる限り少ない実装で済ませることが望ましい。

4．携帯電話 Java 版モバイル RightsShell®

携帯電話 Java 版モバイル RightsShell®(MRS)は、プレイヤ型とライブラリ型の 2 種類の実装方法で開発した。これらは、MIDP と Doja™の 2 種類の Java VM で別々に実装している。ただし、プレイヤ型の実装においては、今後の携帯電話の機能拡張でデータ部分の外部取り出しが可能になった場合を想定し、コンテンツを暗号化して配信している。

表 6 プレイヤ型、ライブラリ型の実装方式の必須機能、実装可能な機能、実装不要な機能

		プレイヤ型		ライブラリ型	
		DRM サーバ	DRM クライアント	DRM サーバ	DRM クライアント
利用者認証	コンテンツダウンロード時	②	—	②	—
	チケット購入時	①	①	①	①
	コンテンツ再生利用時	②	②	②	②
端末認証	コンテンツダウンロード時	①	—	①	—
	チケット購入時	①	—	①	—
セキュア通信	コンテンツダウンロード時	②	②	①	—
	チケット購入時	①	①	①	①
アクセス制限	ダウンロードコンテンツ	—	③	—	③
	再生/実行用データ	—	③	—	③
	チケット	②	③	②	③
起動管理		—	①	—	①
状態管理		—	①	—	①
暗号処理	端末依存	—	—	—	—
	利用者依存	②	②	②	②
	外部記憶依存	—	—	—	—
	コンテンツ依存	②	②	—	—

—：実装不可能

4.1 実装に関する特徴

プレイヤ型とライブラリ型の 2 つの方式に関して、表 3 の機能の実装コンポーネント(サーバ/クライアント)を表 7 に、実装方法の特徴を表 8 に示す。

MRS の実装では、表 7 の検討結果に基づき、必須の全機能と実装可能機能の殆ど全て(オフラインでのコンテンツ利用を実現するため、チケットのサーバ管理に関しては実装せず)を実装した。実装可能機能については、モジュールの共通化を行い、プログラム容量の最適化を行った。その結果、DRM モジュール/DRM ライブラリ(以下、MRS ライブラリと呼ぶ)をともに 11Kbyte 程度に低容量化できた。

表 7 携帯電話 Java 版 MRS のプレイヤ型、ライブラリ型の DRM クライアント機能の実装コンポーネント

		プレイヤ型	ライブラリ型
利用者認証	コンテンツダウンロード時	●	●
	チケット購入時	◎	◎
	コンテンツ再生利用時	○	○
端末認証	コンテンツダウンロード時	●	●
	チケット購入時	●	●
セキュア通信	コンテンツダウンロード時	●	●
	チケット購入時	◎	◎
アクセス制限	ダウンロードコンテンツ	×	×
	再生/実行用データ	×	×
	チケット	×	×
起動管理		○	○
状態管理		○	○
暗号処理	端末依存	×	×
	利用者依存	◎	◎
	外部記憶依存	×	×
	コンテンツ依存	◎	×

◎：DRM サーバ・クライアントでともに実装、○：DRM クライアントのみの実装、●：DRM サーバのみの実装、×：どちらも実装せず

表 8 携帯電話 Java 版 MRS のプレイヤ型、ライブラリ型の実装方法の特徴

		プレイヤ型	ライブラリ型
利用者認証	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	ﾊﾟｽﾜｰﾄﾞ認証	
	ﾁｹｯﾄ購入時	ﾊﾟｽﾜｰﾄﾞ認証	
	ｺﾝﾃﾝﾂ再生利用時	ﾊﾟｽﾜｰﾄﾞ認証	
端末認証	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	ｷｬﾘｱの proxyｻｰﾊﾞが付与する User-Agent により認証	
	ﾁｹｯﾄ購入時	ｷｬﾘｱの proxyｻｰﾊﾞが付与する User-Agent により認証	
ｾｷｭｱ通信	ｺﾝﾃﾝﾂﾀﾞｳﾝﾛｰﾄﾞ時	なし	SSL 通信
		送達確認あり	
	ﾁｹｯﾄ購入時	ﾕｰｻﾞﾊﾟｽﾜｰﾄﾞ、ｺﾝﾃﾝﾂID 等で生成した暗号鍵を用いて暗号化通信 送達確認あり	
起動管理		利用回数、利用時間、利用期限の制限	
状態管理		電話着信に対する実行制御	
暗号処理	利用者依存	ﾁｹｯﾄはﾕｰｻﾞﾊﾟｽﾜｰﾄﾞ等で生成した暗号鍵によって暗号化	
	ｺﾝﾃﾝﾂ依存	ｺﾝﾃﾝﾂはﾁｹｯﾄの内部の鍵で暗号化	なし

4.2 MRS の利用制御

MRS の利用制御は、利用期限・利用回数・利用時間の各制限の論理積あるいは論理和として定義される。これらの定義は、サーバからダウンロードされるチケットに記載されている。各制限の詳細を表 9 に示す。なお、この利用制限は、ODRL[17]などの標準の権利記述言語の利用制限に基づき、必要に応じて適宜拡張可能になっている。

表 9 MRS で設定できる期限

利用期限	有効期限	設定した日時を経過すると利用不可能
	無効期限	設定した日時を経過すると利用可能
利用回数		設定した利用回数に達すると利用不可能
利用時間	利用日数	設定した利用日数に達すると利用不可能
	合計	利用時間の総和が設定時間を超えると利用不可能
	1 回あたり	1 利用ごとに経過時間が設定した時間を超えると利用不可能

4.3 MRS のフレームワーク

MIDP、Doja™の両プラットホームは、特定のクラスファイル(MIDP の場合 MIDlet、Doja™の場合 IApplication－以下、基本クラス)を継承するクラスに処理内容が記述される。MRS は基本クラスを継承したクラス（MRS 継承クラス）を用意し、そのクラスには基本クラスの各メソッド処理に DRM クライアント機能を加えたメソッド（DRM 組み込みメソッド）を追加したフレームワークを用意している。コンテンツ開発者は、このフレームワークを利用して、基本的な DRM クライアント機能をプログラム型コンテンツに付加することが可能である。また、MRS 継承クラスには、コンテンツの利用状態の開始/終了/中断を呼び出し時に行うメソッドも用意されている。コンテンツ開発者は、これをプログラムの必要箇所で呼び出すことにより、ゲームオーバ時のプログラム利用状態の終了など、プログラム実行時の利用制御を任意に行うことができる。

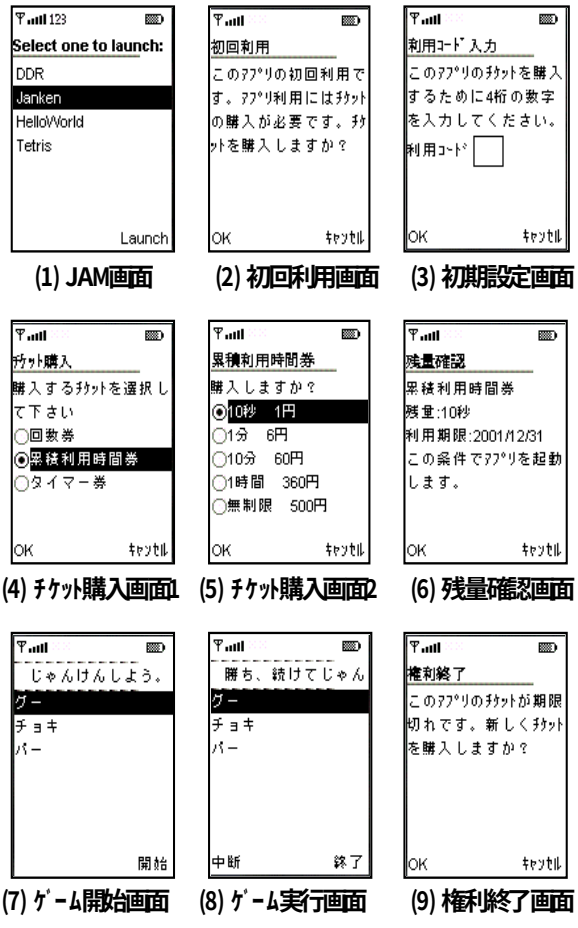
4.3 動作例

MIDP 上で開発したライブラリ型 MRS のシステム動作画面例を図 10 に示す。本例は、簡単なアプリケーション(Janken)に MRS ライブラリを組み込んで、利用制限を行ったものである。

動作時には、まず JAM メニュー画面(1)において、MRS が組み込まれたアプリケーション(Janken)を選択した後、画面(2)から(5)で利用者認証用の利用コードの入力とチケット選択・購入を行う。チケットの購入が終わると、現在のチケット残量が画面(6)に表示され、「OK」を押すと画面(7)のようにゲーム開始画面に移る。

この例では、アプリケーションの起動ではなく、Janken の開始から利用時間が計測されるように、画面(7)の「開始」ボタンが押されたイベントの処理部分で利用状態開始メソッドを呼び出している。このように MRS を使うと、アプリケーション導入画面のようなコンテンツの利用と直接関係のない部分を利用制限の対象から除外することができる。

画面(8)は実際の動作画面である。本例では 10 秒間利用できるチケットなので 10 秒間利用した後は自動的にゲーム実行画面が終了する。このように利用権利が満了になったときは、画面(9)のような権利終了画面が表示される。



(1) JAM画面　(2) 初回利用画面　(3) 初期設定画面

(4) ﾁｹｯﾄ購入画面1　(5) ﾁｹｯﾄ購入画面2　(6) 残量確認画面

(7) ｹﾞｰﾑ開始画面　(8) ｹﾞｰﾑ実行画面　(9) 権利終了画面

図 10 動作例

５．考察

5.1. MRS ライブラリサイズ

MRS ライブラリは、約 11Kbyte のサイズである。ezplus アプリケーション[12]や 504i+FOMA®[15]ではアプリケーションのトータルサイズが 30Kbyte であるため、残り 19Kbyte でアプリケーションを作成する必要がある。しかし、第 1 世代の Doja™の容量が 10Kbyte であったことから、残る 20Kbyte 弱でもビジネス上使い得るコンテンツ作成は可能と判断できる。

5.2 脅威への対抗

MRS は、表 2 に示した脅威のうち、コンテンツの権利外コピー・二次利用の脅威を除き、全ての脅威に対し直接的な対抗手段を有している。上記脅威についても、日本国内で現在使用されている端末ではコンテンツを外部に取り出すことができないので、このコンテンツ保護機能を利用して対抗している。

内部データ取り出し可能な携帯電話では、Java VM のアクセス制限機能が有効に働かなくなり、コンテンツ・チケットの改竄の恐れが出てくるため、アクセス制限機能を実装することが必要になる。また、コンテンツの権利外コピー・二次利用の脅威についても直接的な対抗手段を用意しなければならない。

ダウンロードコンテンツに関するアクセス制限を実行するには、コンテンツの暗号化が有効な手段であるが、3.3 節 A.1 で述べた制限により、Java VM 上では実装できない。したがって、3.2 節で述べた他の３つのどれかの箇所で、暗号化コンテンツを復号し、Java VM に渡す機能を実装する必要がある。

また、再生/実行用データのアクセス制限については、コンテンツ実行時に外部へのデータ取り出しプロセスが起動してないか監視し、起動時には速やかにコンテンツ実行を終了し、再生/実行用データを削除する機能を DRM クライアントに含める必要がある。

チケットのアクセス制限については、単にチケットを暗号化し、改竄を防止するだけでは不十分で、暗号化チケットのコピー・再利用を防止する枠組みが必要である。そのためには、チケットを耐タンパデバイスやサーバに格納したり、チケットがコピーされたものでないか確認する手段を、改竄不可能なサーバ・耐タンパデバイスに実装し、コンテンツ利用時に毎回チェックする機能などが必要である。

コンテンツの権利外コピー・二次利用の脅威の直接的対抗手段としては、コンテンツを特定の端末でしか再生させない機能を実装する場合は、オフラインで端末 ID を取得する機能を DRM クライアントに追加し、端末 ID をもとにコンテンツを暗号化するのが有効である。また、コンテンツを特定の外部記憶でしか再生させない機能を実装する場合は、外部記憶との接続機能を DRM クライアントに追加し、外部記憶にコンテンツを格納する機能を追加すればよい。

６．おわりに

本稿では、機能制限のある Java 環境、外部機器との情報転送制限、少メモリ量・低速度 CPU など実行環境の制限がある携帯電話上で、主として Java アプリケーションを保護するための DRM クライアントを Java VM 上で実装する方式について検討し、その方法に基づき実装したモバイル RightsShell®(MRS)について説明した。実行環境の制限等を利用して、DRM クライアント機能の実装を削減するなど最適化を行い、11Kbyte の低容量な DRM クライアントを実現した。さらに、MRS の特徴であるきめ細かな利用制御方式、プログラム型コンテンツの利用制御を任意の箇所で組み込むためのフレームワークを紹介した。

現在、機能拡張された携帯電話や海外向けの端末を含めて、携帯端末からコンテンツが外部に抽出されても動作するシステムの開発を進めており、携帯電話のコンテンツを他の端末でも安全に再生できる超流通に近づいた DRM システムを構築していく予定である。

参考文献

[1] Enoki K: ”i-mode: The Mobile Internet Service of the 21st Century”, *IEEE International Solid-State Circuits Conference*, pp.12-15, 2001
*i モードは NTT ドコモ社の登録商標
[2]穴沢他：”コンテンツ保護の柔軟化を実現した開放型超流通基盤”, 情処研報, EIP-14-5, 2001
[3]ｹｰﾀｲdeﾐｭｰｼﾞｯｸ, http://www.keitaide-music.org/
[4]Young Laboratory 社：「携帯電話・PHS インターネットサービス利用に関するアンケート」調査結果, http://www.young-lab.co.jp/netscape/report/mobile/mobile_soukatsu.html
[5]Ryoichi Mori, Masaji Kawahara: “Superdistribution: The Concept and the Architecture”, The Trans. of IEICE, vol.E73, No.7, pp.1133-1146, 1990
[6]谷他：コンテンツ利用管理システム：モバイル RightsShell-システム概要-,情処全大 63 回, 5V-2, 2001
[7]中村他：コンテンツ利用管理システム：モバイル RightsShell-コンテンツ配信方式-,情処全大 63 回, 5V-1, 2001 *RightsShell は NEC の登録商標
[8]「ケータイにも著作権管理 NEC がシステムを開発」, 日経エレクトロニクス pp.37-38, 2001/2/26 , http://mim.hml.cl.nec.co.jp/rs/main2.html
[9]細見他：デジタル情報流通アーキテクチャ MediaShell とその利用・課金制御, 情処研報, Vol.98, No.85, EIP-2, pp.49-56, 1998
[10]中江他：ユーザ要求に適合したサービスを提供するカプセル化コンテンツ, 情処研報, EIP-3-11, 1999
[11] CLDC, http://java.sun.com/products/cldc/
[12] MIDP, http://java.sun.com/products/midp/
[13] KDDI 社 Javaｺﾝﾃﾝﾂ(ezplus)に関するﾎｰﾑﾍﾟｰｼﾞ, http://www.au.kddi.com/ezfactory/mm/game01.html
[14] Jﾌｫﾝ社 Javaｺﾝﾃﾝﾂ(Javaｱﾌﾟﾘ)に関するﾎｰﾑﾍﾟｰｼﾞ, http://www.dp.j-phone.com/java/detail. html
[15] NTTﾄﾞｺﾓ社 Javaｺﾝﾃﾝﾂ(iｱﾌﾟﾘ™)に関するﾎｰﾑﾍﾟｰｼﾞ, http://www.nttdocomo.co.jp/p_s/imode/
*i アプリ, Doja は NTTﾄﾞｺﾓ社の商標
[16] JSR-118, MIDP 2.0 Final Release, http://jcp.org/aboutJava/ communityprocess/final/jsr118/index.html
[17] ODRL Initiative, http://www.odrl.net/