マルチメディア通信と分散処理　101－18
コンピュータセキュリティ　12－18
（２００１．２．２１）

情報カプセル流通における利用者システム保護

谷口 展郎，難波 功次，塩野入 理
NTTサイバースペース研究所

Internet経由のディジタル情報流通が現実のものとなりつつあるが，その本格化のためには知的所有権侵害が懸案となる。我々はその一つの解決法として，情報をカプセル化して流通するフレームワークを提案している。情報カプセルとは，流通対象のディジタル情報とその操作のためのプログラムを一体化したもので，このプログラムに内包情報の利用制御機能を持たせることで，ネットワーク上の転送先においても継続的な著作権保護が可能となる。一方，情報カプセルの利用者側からみると，本方式では，ネットワークから受信したプログラムを自らのマシン上で実行することになり，これは大きなセキュリティ懸念につながる。本稿では，情報カプセル流通フレームワークにおいて，利用者の環境を悪意のカプセルから保護する仕組みについて述べる。

User System Protection Mechanism for Capsule-based Information Trans-use Framework

Noburou TANIGUCHI, Koji NAMBA and Osamu SHIONOIRI
NTT Cyber Space Laboratories

While the distribution of digital information such as movies and music over the Internet gets more reality in the recent days, there is still a big problem - piracy/abuse of the information - for the information providers. We have proposed an information trans-use framework based on the 'information capsule' to compose this problem. An information capsule is a file in which digital assets are packaged with a set of programs that plays, edits and protects them. Meanwhile, from the viewpoint of an end user, information capsule may introduce a great security threat to the user's environment because it is an program set loaded from the open insecure network.. In this paper, we present a concept of a end user system protection mechanism against malicious information capsules.

1 はじめに

Internetを利用した，音楽,映画などのディジタル情報の流通がいよいよ現実のものとなりつつある。しかし，ネットワーク情報流通には著作権やプライバシー等の知的所有権に関する懸念がつきまとう。我々は，こうした知的所有権侵害の懸念を緩和し，ディジタル情報の流通を促進するための仕組みとして，情報カプセル流通フレームワークを提案している[1]。

情報カプセルとは，音楽,映画など実際に利用の対象となる情報と，それを操作するためのプログラムを，一つのファイルにパッケージ化したものである。情報カプセルでは，内包された情報は，同じカプセルに内包されたプログラムを介して表示／再生等を行う。このプログラムは，やはり内包されている利用制約条件に沿って動作するよう設計されているので，ネットワーク上の転送先においても，カプセル内の情報を不正な利用から継続的に保護することが可能になる。

一方，このシステムを情報カプセルを利用する側から見ると，カプセル内の情報を利用するには，ネットワーク上から入手したカプセル内のプログラムを，自らの環境上で実行する必要がある。このように，ネットワークからプログラムをダウンロードして実行することは，ローカルシステムに対するセキュリティ上の大きな脅威になる

のは周知の通りであり，その対策なしには情報カプセル流通は成立しない。

本稿では，この「情報カプセルに対する利用者システム保護」のテーマを扱う。まず，現在の情報カプセル及びその流通フレームワークについて説明し，課題を明確化する。次に，この解決方法として「契約に基づくリソース提供」の概念を提案し，そのメカニズムを詳しく述べる。さらに関連技術について考察し，最後に結論を述べる。

2 現状と課題

2.1 情報カプセルとその流通フレームワーク

情報カプセルとは，前述したとおり，音楽,映画など実際に利用の対象となる情報を，それを操作するためのプログラムとともに，一つのファイルにパッケージ化したものである。通常，これらのディジタル情報は独立したデータファイルに格納され，利用の際は該データフォーマットを扱うアプリケーションプログラムを別途起動し，そこから表示／再生等の操作を行う。しかし情報カプセルでは，内包された情報（以下「コンテンツ」）は，同じカプセルに内包されたプログラム（以下「コントロール」）を介して利用される。また，各コントロールは，内包された利用制約条件に沿って動作するよう設計されている。

例えば，ここに音楽データを格納したカプセルがあるとする。このカプセルは，音楽データの他に，この音楽データを再生するためのコントロール（例えばMP3プレイヤー）を内包している。カプセルファイルをダブルクリックすると，まず内包されたMP3プレイヤーが起動され，その操作画面が表示される。ここでユーザーが再生ボタンをクリックすると，利用制約条件（例えば利用期限）のチェックが行われる。利用期限を過ぎている場合，その旨がユーザーに告げられ，楽曲は再生されない。利用期限内であれば，カプセル内に内包されたデータが読み出され，楽曲の再生が始まる。通常，内包データは暗号化されており，カプセル外のMP3再生アプリケーションでそのまま再生することはできない。

このように，情報カプセルにおいては，データとプログラムの関係が同一ファイル内で完結している。喩えるなら，CD一枚毎に一台のCDプレイヤーの中に封入して流通するイメージである。各プレイヤーには，それぞれの利用条件に基づいて操作系が提供されている。ユーザーは，音楽を聴くなど，プレイヤーが提供する正規の操作で音楽を利用することは可能だが，それ以外の不正な利用は困難となる。

このように，情報カプセルを用いることで，ネットワークで情報を流通する際，その転送先においても，ユーザーの情報の利用方法を制御し，不正利用からコンテンツを護ることができる。

現在，我々の実装する情報カプセルでは，コンテンツの種類として静止画,音楽,テキスト,動画などを，また利用制御としてパスワード認証,端末認証,利用回数制限,利用期間制限,利用期限制限などをサポートしている[2,3]。これらの利用制御の一部は，履歴記録機能を用いて実現される。

情報カプセルの流通は以下のように行われる。

まず情報提供者が，コンテンツとなる情報をカプセル化する。この時，コンテンツの利用制約条件も設定し，同封する。カプセル化には，カプセル生成アプリケーション（カプセル ジェネレイター）を用いる，空のカプセルにコンテンツを取り込む，などの方法が考えられる。

このカプセルが，ネットワーク上に公開される。公開の方法は，Webのようなclient-pull型，メイルマガジンのようなserver-push型のどちらでもよい。最終的に，このカプセルはダウンロード／配信されて，情報利用者の手もとに届けられる。

情報利用者は，こうして入手したカプセルを自らのローカルシステム上で起動する。以下，この，利用者がカプセルを実行するローカルシステムを「情報利用者サイト」と呼ぶ。起動されたカプセルは，与えられた利用制約条件の下で，利用者のコンテンツに対する操作を実行する。

現在のところ，カプセルから行えるコンテンツ操作は表示／再生といった読み出し系の操作に限られている。将来的には，カプセル内コンテンツの編集や，編集結果から新たに別のカプセル（オリジナルカプセルの「子」カプセル）を生成する機能なども実装する予定である。これにより，情報の生成／利用／加工／再利用というサイクルを通じてコンテンツ保護を実現し，ネットワーク情報流通を促進する，情報カプセル流通環境を構築していきたいと考えている。

2.2 情報カプセル流通フレームワークの課題 ～情報利用者サイトの保護

上述の流通モデルが，情報提供者側にメリット

をもたらすのは明らかである。しかし，情報利用者の側から見るとき，このモデルには問題が一つある。それは，「コンテンツの利用にあたって，ネットワークから入手したカプセル≒プログラムを，ローカル環境＝情報利用者サイト上で実行する」という点である。

既に"Mellissa","LoveBug"などの事件の例から明らかなように，ネットワーク経由で導入されたプログラムを安易に実行することは，セキュリティ上の大きなリスクを伴う。したがって，カプセルを基本単位とする情報流通を推進するには，このリスクを低減する何らかの方策が必要である。

最も単純なアプローチは，ユーザーがプログラム提供元の「信用」に基づいてプログラム実行の是非を判断することであろう。信用できるブランドの下で提供されるプログラムは実行する，それ以外のプログラムは実行しない - この方針に，ブランド識別のためのディジタル署名技術を組み合わせることで，ある程度妥当なセキュリティリスクの低減が期待できる。

しかし，この仕組みだけではやはり不充分で，以下のような問題がある。

最も大きな問題は，実行の是非の判断（とその責任）がエンドユーザーに委ねられている点であろう。ネットワーク化が進展すれば，情報カプセル以外にも，ネットワーク上の外来プログラムを実行する機会はまちがいなく増える。この時，いちいち実行の是非を判断し，しかもその結果責任まで負わされるというのは，エンドユーザーの負担が大き過ぎる。

また，信用の判断基準が主に提供者のブランドに依存するため，既存の確立されたブランドのネットワーク支配が固定化されやすくなり，結果的に選択の柔軟性などのオープンネットワークならではの利点が失われかねない。

また，この方法では，プログラムの実際の振る舞いのチェックは強制できない。例えば，同じ提供元Pから2つのプログラムA,Bが利用者Uのサイトへ送られたとき，UにとってAの振る舞いは容認できるがBの振る舞いは本当は容認できないような場合でも，Pの信用だけをもとにA,B両方を実行する，という事態が生じうる。

我々は，これらの問題を克服するための仕組みとして，「契約に基づくリソース提供」の概念を提案する。上記の「信用に基づく実行」を，この「契約に基づくリソース提供」に拡張することにより，オープンネットワークの利便性を大きく損なうことなく，妥当なレベルのセキュリティを提供することを可能にしたいと考えている。

3 契約に基づくリソース提供

3.1 概要

「契約に基づくリソース提供」の概念は，実は情報カプセルにおけるコンテンツ保護の仕組みに，その発想の由来がある。

情報カプセルは，与えられた利用制約条件に基づいて，コンテンツの利用制御を行う。これにより，コンテンツを不正な利用から保護する。つまり，情報カプセルは「内部のリソースを外部の不正な利用から保護する」ために「所与の条件に基づいて内部リソースの提供を行う」メカニズムを持つと言える。

利用者サイトのセキュリティ目標も「内部のリソース（CPU,メモリ,ディスク,ファイルなど）を外部の不正な利用から保護する」ことにある。これは，情報カプセル同様，「所与の条件に基づいて内部リソースの提供を行う」メカニズムがあれば，実現可能であろう。

「契約に基づくリソース提供」の概念は，この相称性に基づいている。情報カプセルと利用者サイトは，互いのリソースの利用条件を「契約」として交換し，これに則って実際のリソース供給を行う。こうして，予め契約を取り交わすことによって，不正なリソース利用の検出が容易になり，また問題発生時の責任の所在も明確になる。

技術的には「契約に基づくリソース提供」は，トランザクション毎に動的にロードされ，トランザクション内では（基本的に）静的に適用される，拡張セキュリティポリシーと考えることができる。通常，セキュリティポリシーは，トランザクションをまたがって静的に適用される。一方，通常，プログラムは，実行中＝実行環境とのトランザクション中に，実行環境からメモリやディスクなどのリソースを動的に獲得できる。「契約に基づくリソース提供」はこの中間に位置し，トランザクション単位での柔軟性と，トランザクション内での安全性の両立を目指すものと言える。

「契約に基づくリソース提供」は，概ね以下の3段階から成る。

<1> まず，何らかのタイミングを契機として，情

報カプセルと利用者サイトが，互いのリソースの利用条件に関する契約を取り交わす

<2> 情報カプセルが実際に利用者サイト上で起動されると，相互に認証を行い，正規のリソース利用者であることを確認する

<3> 情報カプセル実行中は，相手の要求に応じ，契約に則って互いにリソースを提供する。場合によっては，提供するリソースについて，その利用方法を継続的に監視し，不正な利用は事前に防止する

これら各々については，3.2節以降でより詳しく説明するが，その前にもう一つ重要な概念である「パブリックリソース」について述べておく。

「パブリックリソース」とは一言で言うと「無契約で利用できるリソース」である。例えば，カプセルが宣伝のため動画コンテンツの一部を対価なしに提供する，などがこれにあたる。一方，このような動画カプセルを試用するには，利用者サイトの側からも，契約無しに使える計算機資源がある程度提供されている必要がある。

このように，各々のカプセルやサイトが，無契約で利用できるリソースを，予めある程度提供しておくことは，オープンネットワークにおける情報流通の柔軟性を大きく高める。このメリットは単にプロモーションや試用のようなものに限られるものではない。あえて言うなら，それは現在のインターネットにおける「自由」を，パブリックリソースを明確に限定することで安全性を向上させつつ，継承／強化するものである。

例えば現在のInternetにおける，WebページやGnutellaのようなP2P検索インターフェイスはパブリックリソースの一例である。またこうした検索インターフェイスを，将来はカプセルが備えることも考えられる。或いは，未来のSETIプロジェクトは，ネットワーク上の全てのサイトのパブリックリソースを少しずつ使って処理を行うようになるかもしれない。

こうしたパブリックリソースの存在を前提として，改めて「契約に基づくリソース提供」を考えてみると，上記の段階<1>の前に，

<0> 情報カプセルが利用者サイトにダウンロード／配信されてくると，利用者はまず，サイトのパブリックリソースで構成されるエリア＝パブリックエリア上でカプセルのパブリックリソースを試用し，契約の是非を判断する

という段階を想定できる。ここで是と判断された場合は<1>の契約が行われ，カプセルはサイトの非パブリックなリソース（プライベートリソース）で構成されるエリア（プライベートエリア）上に移動して，以後そこで起動<2>,実行<3>される。この場合，<1>の契約における「何らかのタイミング」とは，即ち「カプセルがサイトのパブリックソースからプライベートソースへ移動するタイミング」と考えることができる。

以下，このモデルに基づいて，「契約」「認証」「リソース提供」について述べていく。

3.2 契約

情報カプセルが利用者サイトのパブリックエリアからプライベートエリアへと移行する時，両者の間で，互いに提供するリソースについて契約が取り交わされる。

この契約は，自らのリソースの利用をどこまで相手に許可するかという「セキュリティポリシー」であると同時に，自らのリソースの利用をどこまで相手に保証するかという「サービスレベル記述」の側面を併せ持つと考えることができる。例えば，ある情報カプセルの利用期限が「2001年2月20日」である場合，そのカプセルは2001年2月20日までは正常に動作しなければならない。

契約で扱われるリソースの種類としては，さまざまなものが考えられる。カプセルから提供されるもは，主として「情報」系コンテンツ（映画,音楽,小説,ニュースなど），あるいは「ツール」系コンテンツ（オフィス生産性ツールから前述の検索エンジンまで種々のプログラム類）などであろう。一方，利用者サイトから提供されるの」は，およそネットワーク上でリソースとして扱える全てのもの（CPUから通信帯域まで，種々のデバイスやサービス）と考えてよい。この他，オフラインリソースである「人」や「金銭」なども記述できればさらに便利であろう。

この他，契約の記述には，リソースに対する操作（許可される操作，禁止される操作）の種類，操作の主体，リソースの量／質，などが必要となるであろう。

現在のところ，実際にこれらの多種多様なリソースに関するあらゆる形態の契約を表現しうる，機械可読な（単一の）言語は存在しない。そうした言語を策定することは，法律を記述可能なコンピューター言語を作り上げることにほぼ等しく，仮に仕様化できたとしても，その仕様は巨大すぎ

て現在の技術水準での実装は無理だろう。

現時点で実用性のある契約の記述法として，我々は，制約の表現としてはCORBAやJava等のセキュリティポリシー記述を，リソースの表現としてはDMTF (Distributed Management Task Force) の規定するCIM (Common Information Model) などを用い，対象を限定してある程度有効に機能するものを考えている。

3.3 認証

利用者サイトのプライベートエリアで情報カプセルが起動されると，相互に認証を行い，ロードすべき契約を同定する。

認証の最も基本的な仕組みは，ディジタル署名である。しかし，単にディジタル署名だけでは「なりすまし」等のクラッキングを防止することは難しい。また電子署名を利用する場合，署名後にデータが更新されると，そのデータは電子署名的には「不正」となるため，更新後に改めて署名を行い正当性を再認する必要がある。我々の情報カプセル流通フレームワークでは，カプセルもサイトも更新によって変化する可能性があるため，こうした再署名の手間はできるだけ軽減したい。

そこで我々は，電子署名に加えて，カプセルとサイトが共有するトランザクション履歴を利用して，相互に認証を行う仕組みを導入することを考えている。

この「履歴に基づく認証」は，「情報カプセル流通のような分散オブジェクトの世界では，履歴は，オブジェクトのネイティブな特性と同じかそれ以上に重要な，オブジェクトを識別するための要素である」という考えに基づくものである。

例えば，Z氏と電話で話している2人の人物が，ともに同一の人物Xを名乗っているとする。顔は見えず，声もそっくりなので，外形的な情報からだけでは，どちらが本物なのかを判断することはできない。この時，どちらが本当にZ氏の知り合いのX氏なのかを見分ける，最も単純で有効性が期待できる方法は，Z氏とX氏の共通の記憶について質問することであろう。

分散オブジェクトの世界では，オブジェクトのネイティブな特性，つまり，クラスや属性といった外形的な固有値は，簡単に複製されうる。もちろん，記憶に相当する履歴も複製されうるが，複製後一度でもそのうち一つのオブジェクトとトランザクションを持っていれば，その最新のトランザクションに関する履歴を持っているのは，当の相手のオブジェクトしかありえない。また更新情報そのものである履歴を認証に利用することで，再署名を行って改めて正当性を保証する手間をある程度省略することもできる。

このように，電子署名に加えて，履歴を用いることで，より安全かつ柔軟な認証が実現できるのではないかと考えている。

3.4 リソース提供及びその監視,制御

起動時の認証が正常に行われると，情報カプセルの実行が開始され，契約に記載されたリソースの提供が開始される。

リソースへのアクセスは，仮想的なビューを介して一元的に行うことを考えている。提供元は，契約の記述に基づき，提供先がアクセス可能なリソースをこのビューに配置する。提供先はこのビューを受け取り，必要に応じてビュー上のリソースへのアクセス要求を提供元に依頼する。提供元は，要求に応じて該リソースの実体もしくは参照を提供元に返す。

この方法をとると，アクセスが許可されないリソースはそもそも提供先からは「見えない」ため，安全性が高い。また，トランザクション開始時にリソースビューを共有するので，リソースへのアクセスを抽象化して実行環境の実装から切り離すことができ，設計が容易になる。

ビューのデータ構造は，ディレクトリ構造を想定している。ビュー内のリソース数が多い場合は，その中から特定の条件に合うリソースを探しだす機能＝検索機能など，将来的にはLDAP的な資源探索プロトコルのサポートも考えている。

ビューを介してアクセスを提供したリソースについては，その利用方法が契約どおりであるかどうか監視／制御を行うことで，より契約の強制力を高めることができる。

既に，情報カプセルについては，カプセル内のコントロールを介する方法で，ある程度の利用制御を実現している。

一方，利用者サイトについては，想定されるあらゆるリソースの利用監視／制御を行うには，実行環境であるOSやVM（Virtual Machine）そのものに手を入れる必要がある。そのため，3.2節で述べたのと同様，当初は対象リソースを限定してその範囲内で有効に機能するものを実装することを目指している。現在は，Java環境上で，Javaセ

キュリティアーキテクチャを用い，情報カプセルの実行に特化したセキュリティポリシーを定義，これを情報カプセル実行時に実施するライブラリーを作成することで，利用者サイトのリソース利用監視／制御を実現できないかを探っている。

4 関連技術

分散オブジェクト環境のセキュリティについては，非常に多くの研究や実装が既に存在する。

CORBAセキュリティサービス仕様[4]はその集大成の一つで，セキュリティに関連する非常に広範な問題を扱っている。CORBAではドメインベースのセキュリティモデルを採用しており，各オブジェクトはドメインによってグループ化され，アクセスの可否はドメインによって判定される。

Javaは，CORBAに似ているが独自のドメインベースのセキュリティアーキテクチャ[5]を持っている。Javaセキュリティアーキテクチャとしては，よくアプレットのsandboxモデルが取り上げられるが，現在はより柔軟なモデルに拡張され，アプレットだけでなく様々なクラスローダー毎にセキュリティポリシーを設定できる。実際にこの機構を利用しているかどうかは不明だが，iモードのJavaサービスであるiアプリ[6]は，Javaアプリケーションでありながら，ローカルリソース（携帯電話機上のメモリ等）へのアクセスが，各アプリケーション毎に提供されるScratchPadという記憶領域に限定されている。

モバイルエージェント環境Telescript[7]では，'place'という一種の仮想ビューを介してローカルリソースがグループ化され，エージェントに提供される。エージェントがplace内のリソースにアクセスしようとすると，placeは，エージェントのアクセス権'permit'をチェックし，可否の判定を行う。

より一般的な分散OSとしては，例えばPlan 9 [8]がある。Plan 9は，キーボードやディスプレイ等を含め，基本的に全てのリソースをファイルとして扱い，しかもその名前空間を，リモートシステムに対して個別かつ動的に提供することを特徴とする分散OSである。リモートに提供されるリソースは個別の名前空間を持つので，仮にある名前空間のリソースがクラックされてもその影響は該名前空間内に限定される。

WebOS[9]は，現在のGnutellaのようなP2Pシステムや，Microsoft.Netなどのwebサービス環境の前身ともいえる分散ファイル／プログラミングシステムである。WebOSでは，「セキュリティ機構の冗長化」や「権利の細流度化」などを導入することで，Internet-wideの分散システムに必要なセキュリティを実現するとしている。

5 おわりに

本稿では，情報カプセルを利用したネットワーク情報流通システムにおける利用者サイト保護の必要性について述べ，それを解決するための方法として，「契約に基づくリソース提供」の概念を提案した。この概念は情報カプセル及び利用者サイトについて相称的であり，したがって情報提供者と情報利用者の双方にとって受け入れ易い情報流通体系を実現できるものと考えている。

今後は，提案した概念を実現するシステムの実装を行う。当面は，対象リソースや対象操作を限定し，Javaカプセルと連携して動作する利用者サイト管理システムを，Javaプラットフォーム上で実現することを目指している。

参考文献（URLは2001年1月下旬時点）

[1]谷口, 森賀, 久松, 櫻井: "マルチメディア情報ベースとその格納単位Matryoshka", 情処 DICOMOシンポジウム, 1999

[2]加賀美, 森賀, 塩野入, 櫻井: "コンテンツ流通における自律管理を目的としたカプセル化コンテンツMatryoshka", 情処 DPS 97-18, 2000

[3]阿部, 谷口, 塩野入: "Javaを用いた動画配信カプセルの実装", 情処 DPSワークショップ, 2000

[4]ftp://ftp.omg.org/pub/docs/formal/00-06-25.pdf

[5]http://java.sun.com/j2se/1.3/ja/docs/ja/guide/security/spec/security-spec.doc.html

[6]http://www.nttdocomo.co.jp/i/java.html

[7]General Magic Inc.: Telescript Language Reference, 1995

[8]http://plan9.bell-labs.com/plan9dist/

[9]http://www.cs.duke.edu/ari/issg/webos/