オープンソフトウェアにおけるソフトウェア保護

Software Protection in Open Software Environment

橋本　幹生　■ HASHIMOTO Mikio
山口　健作　■ YAMAGUCHI Kensaku
磯崎　宏　■ ISOZAKI Hiroshi

パソコン（PC）や携帯情報端末（PDA）のマルチメディア化に伴い，オープンシステムにおいてコンテンツ保護が必要とされる事例が増えている。オープンシステムを用いたコンシューマ機器では，ソフトウェアを解析してコンテンツ保護の仕組みを破ろうとする一部ユーザーの行為が避けられない。そのため，ソフトウェアに含まれる秘密情報などを解析から防止する手段が必要となる。とりわけ，異なる保護方式の伝送媒体，記録媒体間のコンテンツ転送には，機器ソフトウェアの担う部分が大きい。TRS（Tamper Resistant Software）と呼ばれるソフトウェア保護技術は，このような保護の連鎖を守るうえで重要な役割を果たしている。

With multimedia applications having become popular for personal computers and personal digital assistants, some content protection mechanisms are necessary for their open software. Since an open software is inevitably vulnerable to the possibility of attack, a software protection mechanism also serves to protect its embedded secret information for content protection. Moreover, the chain of secure transmission and recording of copyrighted content depends on the software of an appliance.

A software protection technology called tamper resistant software (TRS) is playing an important role in appliance software protection.

1　まえがき

現在のコンテンツ保護方式は，基本的に暗号技術に基づいている。ユーザーが，入手したコンテンツを機器で再生して楽しむことができることと，コンテンツの不正コピーなどの防止を両立するために，ユーザーがアクセスできないような秘密情報（鍵情報）が機器にあることが前提となる。

ソフトウェアにこれらの秘密情報が含まれているとき，それらの保護が必要である。とりわけ，ハードウェアやソフトウェアの仕様が公開されたオープンシステムにおいてこの問題は深刻となる。ここでは，コンテンツ保護とオープンソフトウェアにおけるソフトウェア保護との関係と課題を示し，この課題の解決手段としてのTRS技術について述べる。

なお，ここではオープンシステムを，広い意味において公開された仕様に基づくハードウェア及びソフトウェアインタフェースを持つシステムと定義し，そこで実行されるソフトウェアをオープンソフトウェアと呼ぶこととする。オープンシステムの代表にはPCベースのWindows®（注1）や，Linuxなどが挙げられる。

2　オープンシステムと著作権保護

コンテンツ保護にかかわるオープンシステム導入は，二つの方向から広まりつつある（**図1**）。一つはPCやPDAのようなオープンシステムの機器がマルチメディア処理能力を備えるものであり，もう一つは家電に分類される組込み機器にオープンシステムが導入されるものである。

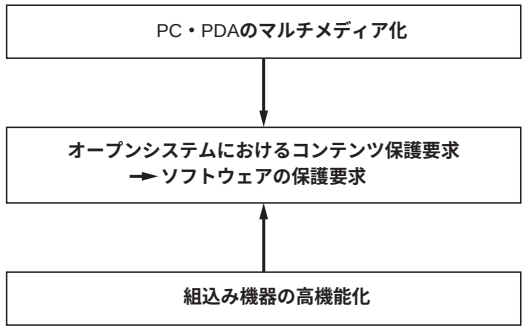


図1．オープンソフトウェア導入の背景 ― 二つの方向からオープンシステムの導入が進みつつある。

Background of open software introduction

オープンシステムを使わない従来の組込み機器では，製品がソフトウェアのデバッグ機能を持たないこと，ソフトウェアや筐体（きょうたい）内部のコネクタが独自仕様であること，筐体に物理的保護が行われていることなどにより，内部のソフトウェアの保護は必要とされていなかった（**図2**（a））。物理的な障壁と仕様情報の非公開による二重の障壁がソフト

（注1）　Windowsは，米国Microsoft Corporationの米国及びその他の国における登録商標。

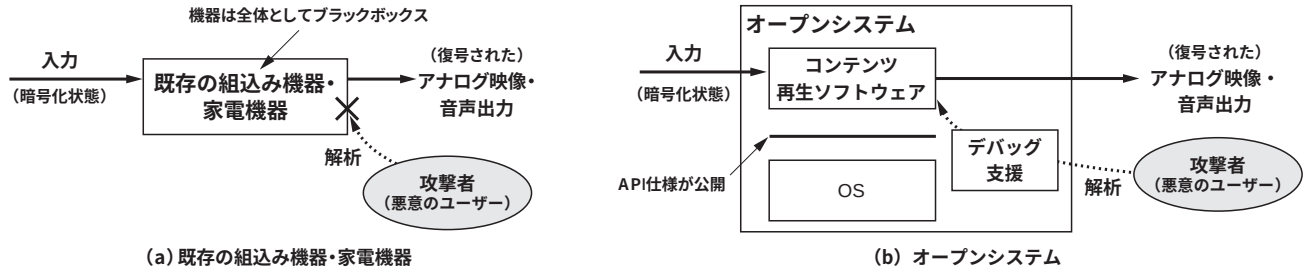


図2．既存システムとオープンシステムの違い― オープンシステムでは内部情報が公開されていることから，アプリケーションのブラックボックス化が要求される。

Difference between existing system and open system

ウェアを守っていたと言える。コンテンツ保護の仕組みを破ろうとする攻撃者からは，筐体内部はいわばブラックボックスとして見えていた。

ところが，PCのようなオープンシステムでは，ハードウェア情報はもとより基本ソフトウェア（OS）とアプリケーションプログラムのインタフェース（API）が開示されていることに加えて，OSにデバッグ支援機能まで備えられている場合も多い（図2（b））。これらの情報は一般ユーザーにも入手可能であり，ソフトウェアの解析は既存の組込み機器に比べて容易になっている。ソフトウェアの保護を怠ったことにより，秘密情報が解析されてしまった残念な事例が既に知られており[(1)]，もはやPCの内部はユーザーにとってブラックボックスとは言いがたい。

このような条件下で，保護対象のソフトウェア自身の内部構成を解析困難なものとする手法がTRSと呼ばれる技術である[(2)]。TRSとは広い意味でプログラム難読化一般を意味するが，多くの場合自己改変型プログラムと呼ばれる手法が基本となっている。この手法は，記録媒体上では暗号化されていたソフトウェアが，その実行と同時に自分自身を復号しながら実行を進めるものである。暗号化の方法やパラメータなどは，ソフトウェアのベンダーが任意に選択する。従来機器では，仕様の非公開性に基づいていたソフトウェアの保護の土台を，ソフトウェア内部の暗号パラメータ情報などの非公開性に置き換えたものと言える。理想的には，オープンシステムにおいても保護対象のソフトウェアをブラックボックスとすることができる。

ただし，後述のようにTRS技術にはいくつかのトレードオフがあり，ソフトウェア保護の安全性を厳密に追及することは，現実のシステムにおいては高コストとなる。エンターテインメントコンテンツ保護技術の目的は，ホビーユーザー（日曜プログラマー）によるいたずら半分の不正複製（Casual Copy）を妥当なコストにおいて防止することに限定して，以下の説明を進める。

3 TRS技術の原理

TRS技術の目的はプログラム解析の防止であり，その中心はプログラム難読化である。

実際に使われるプログラム難読化手法の多くは自己改変型プログラムと呼ばれるもので，あらかじめプログラムを暗号化しておき，実行時にプログラムが自分自身を復号しながら実行する手法である。

図3は，単純化した自己改変型プログラムの実行経過である。メモリに配置された機械語プログラムは三つの領域（a）～（c）に分割されている。実行開始時の状態（1）では先頭の領域（a）のみが平文で，残りの領域（b），（c）は暗号化状態にある。領域（a）が実行されると領域（b）の復号を行い，復号が完了した領域（b）に制御を移す（状態（2））。領域（b）では呼出し元の領域（a）を再暗号化して，次に実行する領域（c）を復号する。これら一連の自己改変処理の中に，秘密にしたい処理や値が埋め込まれているのである。

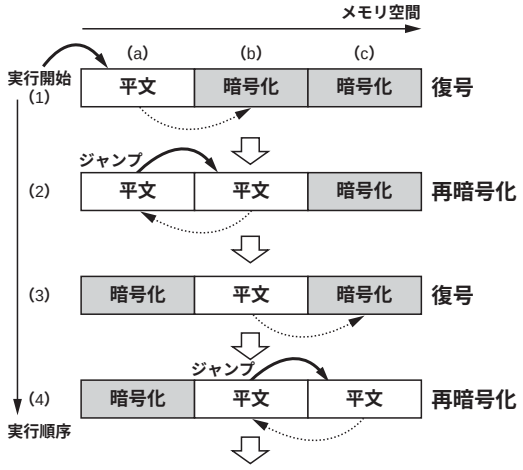


図3．自己改変型プログラムの実行― 各領域は復号―実行―再暗号化のサイクルをたどる。

Execution of self-modification program

攻撃者がプログラムを解析する方法には大きく2通りの方法がある。ファイル上のプログラムを解析する静的解析と，実システム上のプログラムの動作を解析する動的解析である。プログラム暗号化はどちらの解析方法にも対策となる。ファイル上ではプログラムの大部分が暗号化されているため，単純な静的解析だけでは秘密処理の部分を見ることはできない。動的解析についても，実行中に領域の再暗号化が行われるため，すべての領域が同時に復号状態にある瞬間はない。したがって，プログラム全体の解析のためには，攻撃者は領域がどの時点で復号状態にあるかを知らなければならず，実行の過程を逐一追跡することを強制される。

プログラム難読化に暗号化を使うことで，様々な暗号アルゴリズム（暗号化というよりは，むしろスクランブルというほうが適切である）やパラメータの選択が可能となり，攻撃側の探索領域が広くなる。図3の例では，プログラム領域は3分割されていただけだったが，実際のプログラムでは解析を困難にするため，メモリ領域の分割は更に細分化され，領域の配置や呼出順序も解析が困難になるよう変更される。これら任意選択されたパラメータで構成される自己改変型プログラムの全体を，人手だけで解析することは実際上不可能に近い。

一方，TRSの適用は性能や開発効率に影響する。まず，自己改変処理のオーバヘッドにより，処理性能はTRS化なしのプログラムに比べて低下し，プログラムのサイズも大きくなる。また，TRS技術により解析が難しくなっていることは直接開発効率の低下にもつながってしまう。

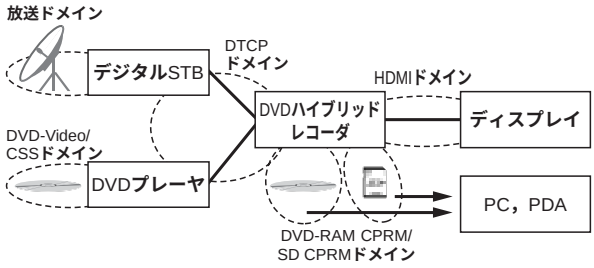
いくつかのデメリットはあるものの，後述のようにオープンシステムでコンテンツ保護をソフトウェア実装する際には，なんらかのTRS技術の導入がライセンス条件で義務づけられており，広く利用されている。

4 システムの中のソフトウェア保護

4.1 セキュアドメイン間の連鎖

コンテンツ保護は単体の装置で閉じたものではなく，様々な機器の連携・連鎖によって完全なものとなる[(3)]。コンテンツ保護ドメインの連鎖の模式を**図4**に示す。

デジタル放送やDVD-Videoの形式で暗号化されて配布されたコンテンツは，セットトップボックス（STB）やDVDプレーヤで一度復号された後，再度DTCP（Digital Transmission Content Protection）形式で暗号化されてDVDハイブリッドレコーダに伝送される。コンテンツには複製，再生，出力の許可条件を表す制御情報が付加されている。DVDハイブリッドレコーダでは，制御情報に定められた許可条件の範囲内で，コンテンツをDVD-RAMやSDカードに複製することができる。同様に，メディアに複製されたコンテンツは，制御情報の範囲内でハイブリッドレコーダ自身で再生して，HDMI（High Definition Multimedia Interface）を通じてディスプレイに表示することもできるし，別の再生装置，例えばPCやPDAで再生することもできる。このように，コンテンツ供給側から再生装置まですべての伝送・記録メディアにおいて，一貫して暗号化状態でコンテンツを扱う経路が現在確立されつつある。



CSS　：Content Scrambling System
CPRM：Content Protection for Recordable Media

図4．セキュアドメイン間の連鎖 ― 各機器のソフトウェアがドメインの変換を担う。

Chain of secure content domain

この連鎖の安全性は，各ドメインの伝送・記録媒体上の暗号化方式に加えて，ドメインの連鎖の結節点にあたる機器の安全性にも依存する。同時にドメイン結節点での処理は，制御情報の高度化に伴って複雑化する傾向にある。例えば，複製制御では従来の単なるコピーの可・不可から，回数を制御する仕様の導入が進みつつある。HDD（ハードディスクドライブ）ビデオレコーダにおけるタイムシフト視聴を対象とした制御では，蓄積画像の再生回数，保存時間の制限が要求されている。

保護ドメイン結節点の処理では，物理メディア依存のフォーマット変換に加えて，複雑な制御情報の処理を正しく行うことが要求される。このため，大部分の機器ではなんらかのソフトウェア処理が制御情報管理に関与している。ソフトウェア保護には，秘密情報の保護だけでなく，攻撃者の干渉があってもこのような管理動作が正しく行われることが要求されるのである。

4.2 コンテンツ保護ルールとソフトウェア保護

コンテンツ保護における保護対象は，秘密情報やコンテンツそのものだけでなく，コピーの作成回数などの機器のふるまいも含まれる。これらは保護仕様のライセンスルールによって定められており，コンプライアンスルールとロバストネスルールの二つの部分で構成される。

コンプライアンスルールでは複製，再生，出力の各制御情報の定義と規則が定められており，機器はこの定義に従って制御情報を解釈し，規則に従って正しく動作しなければならない。ロバストネスルールでは，ライセンスされる機密情報

(鍵情報)の機密性の定義や機密情報を実装する際の規則が定められている。ソフトウェア実装における規則もここで定められている。多くのライセンス条件ではコンプライアンスルールで定められたルールが,一般ユーザーによって回避されないことが要求されている。

コンテンツ保護を実装するソフトウェアは機密情報を格納していると同時に,コンプライアンスルールを実現する制御プログラムも含んでいる。ソフトウェアに埋め込まれた鍵情報が暴かれてはならないのと同時に,コピー制御や再生制御のルールがユーザーによって勝手に変更されることも防がれなければならない。

4.3 ソフトウェア実装とソフトウェア保護の必要性

現状では,複雑なコピー制御などのコンテンツ保護機能をすべてハードウェアにより実装するのは困難であり,多くの部分をソフトウェアに頼らざるをえないが,コンテンツ保護をソフトウェア実装することの積極的な利点もある。

一つは,追加コストなしに機器にコンテンツ保護機能を追加できることである。PC, PDAなどの汎用機器では,必ずしも著作権保護されたコンテンツを扱うとは限らない。このような機器では,コンテンツ保護のソフトウェア実装によってハードウェアのコストアップを回避できる。

また,ソフトウェアによる実装はハードウェア実装と比較して更新が容易であり,コンテンツ保護システムの更新が容易になる利点もある。

5 あとがき

オープンシステムの普及とコンテンツ保護ルールの複雑化を背景として,コンテンツ保護システムにおけるソフトウェア保護技術の必要性が高まっている。コンテンツ保護におけるソフトウェア保護技術では,安全性と同時に開発効率,性能といった広い意味でのコスト要因とのバランスが重要となる。

TRSは,既存環境に対してバランスの良いコンテンツ保護手段を提供できる技術だが,安全性の評価手段などはいまだ確立していない。一方でプロセッサアーキクテチャの変更やハードウェアの追加によって,より安全なソフトウェア保護を実現する提案がされている[4]。これらの提案は,追加ハードウェアを必要とするものの普及や量産によってコストは低下する可能性があり,今後注目が必要である。

文 献

(1) Andy Patrizio.“ The DVD Hack ”, Wired News. <http://www.wired.com/news/technology/0,1282,32265,00.html>, (accessed 2003-02-17)
(2) Auschmith, D.; Graunke G.“ 逆解析や改変からソフトを守る ”. 日経エレクトロニクス. 706, 1998, p.209－220.
(3) 山田尚志.“ DVDを起点に著作権保護空間を広げる ”. 日経エレクトロニクス. 2001-8-13, p.143－153.
(4) Lie, D. et.al.“ Architectural Support for Copy and Tamper Resistant Software ”. ACM Computer Architecture News. 28, 5, 2000, p.16－177.

橋本 幹生 HASHIMOTO Mikio

研究開発センター コンピュータ・ネットワークラボラトリー研究主務。ホームネットワーク・セキュリティ技術に関する研究・開発に従事。電子情報通信学会, ACM会員。
Computer Network Lab.

山口 健作 YAMAGUCHI Kensaku

研究開発センター 通信プラットホームラボラトリー。ホームネットワーク技術に関する研究・開発に従事。
Communication Platform Lab.

磯崎 宏 ISOZAKI Hiroshi

研究開発センター コンピュータ・ネットワークラボラトリー。ホームネットワーク・セキュリティ技術に関する研究・開発に従事。
Computer Network Lab.