簡便なコンテンツ配布システムの提案とその安全性の検討

大道健広† 多々内允晴†
豊田工業大学†

1. はじめに

インターネットでは，動画，音楽，文章等のコンテンツをデジタルデータとして普及させることは容易で，誰でも手軽に扱えるものとなっている．そこでは不正コピーによる著作権の侵害が容易で，コンテンツを扱う上で大きな障害である．著作権保護のための DRM という技術が注目を集めているが，十分な機能・性能を持った方式はまだ存在しない．DRM にはハードウェアで保護するものとソフトウェアで保護するものがある．特に，後者において問題となるのは，著作権保護を行うプログラムのリバースエンジニアリングに対する耐タンパ性である．また，現状の DRM は課金の有無に関係なく，コンテンツ作成者とライセンスサーバとのやりとりが必要となり，授業資料の配布あるいはグループの会員間で著作物を配布するといった課金はしないものの著作権の保護が必要な場合においても煩雑な作業が必要である．このような場合でも安全性を保ちつつ利便性を高めた DRM の使い方が求められる．

ここでは，コンテンツの無料配布でも簡易に使用できるソフトウェア DRM による配布システムの提案とその安全性を検討する．

2. 従来技術

動画や音声を保護する従来技術である Windows Media DRM では，ライセンスキーは「シード」と「キーID」と言う二つの要素から生成される．「シード」はコンテンツ所有者とライセンスサーバのみが知り，「キーID」はコンテンツ暗号化時にコンテンツのヘッダに埋め込まれる．

ユーザからのコンテンツの再生要求があった場合，ライセンスサーバは固有の ID(GUID)を再生要求したユーザに割り当てる．次に「シード」とコンテンツから抽出した「キーID」を組み合わせてライセンスキーを再生成する．そして，サーバは GUID でライセンスキーを暗号化してユーザに送付する．ユーザは GUID を用いてライセンスキーを使用，コンテンツを再生する．

この方式では無料配信においても有料配信と同様に，コンテンツ所有者とライセンスサーバの間でライセンスキーの「シード」の共有のために，予めコンテンツ所有者はライセンスサーバとやりとりを行う必要がある．

3. 提案システム

ここではコンテンツ毎のライセンスキー管理を各ユーザの PC にインストールして用いるコンテンツベースソフトウェア(以下，CBS と呼称する)と名付けたソフトウェアに任せることで，サーバでのライセンスキー管理と，無料配信において作成者がサーバとのやりとりを必要としない簡便なコンテンツ配布システムを提案する．

3.1. システムの概要

提案システムはサーバの公開鍵で暗号化したライセンスキーをコンテンツと共に配布し，ユーザが閲覧・購入時にサーバに問い合わせてライセンスキーを復号してもらい閲覧するものである．そのため，サーバで管理する鍵はサーバに唯一の公開鍵とペアの秘密鍵１つのみである．

課金処理を行わない場合，作成者はサーバとのやり取りは不要である．

また，教材配布等の際に生徒あるいは会員に閲覧を限定するという場合，コンテンツのヘッダに許可対象リストを埋め込み，サーバがユーザをそのリストと照合することで認証するため，事前のサーバとのやりとりは不要である．

著作権保護はサーバとユーザがそれぞれ公開鍵と秘密鍵のペアを持ち，それによる暗号化と共有鍵暗号とを組み合わせることで実現する．

また，安全性を守るために，CBS そのものの不正コピーと，秘密鍵と共有鍵の流出を防止しなければならない．これは従来のソフトウェア DRM でも同等である．

3.2. 提案システムの構成・機能

図 1 に示すように提案システムはコンテンツ作成者とユーザが用いる CBS とサーバで構成される．

CBS はライセンスキーとコンテンツの管理，暗号・復号の処理，サーバとのやり取りをする．

サーバはコンテンツ作成者認証，CBS 認証，ユーザ認証，入金処理，ライセンスキーの暗号・復号処理等の機能を持つ．

3.3. 提案システムの動作

ユーザが使用する CBS 自体を不正コピーさせないため，CBS は起動の際にレジストリの PC 構成情報を参照し，それがインストールされた PC のものと異なっていた場合，起動しない．また，CBS 初回起動時に公開鍵 Kcp と秘密鍵 Kcs のペア

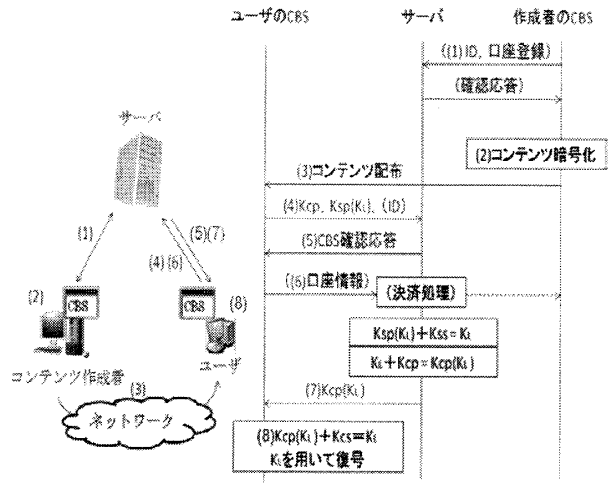

図 1. 提案システムの構成・動作

サンプル	
認証サーバの情報	
(作成者のID)	(閲覧許可するユーザの公開鍵リスト)
Kspで暗号化されたライセンスキーKL	
KLで暗号化されたコンテンツ	

図 2. 配布ファイルのフォーマット

を作成する.

次に図 1 で示した CBS-サーバ間のやりとりについて述べる. カッコ内の数字は図 1 内の数字と対応する.

(1) コンテンツ作成者は課金が必要な場合, サーバにコンテンツ作成者の ID と入金に用いる口座情報を登録. 無料配布では不要.

(2) コンテンツをライセンスキー K_L で暗号化. K_L をサーバの公開鍵 Ksp で暗号化. コンテンツのサンプル, サーバの情報を添付. 課金する場合は ID を, 閲覧許可を与える相手を制限する場合は, 該当ユーザの公開鍵リストを更に添付.
(配布ファイルのフォーマットは図 2)

(3) 配布ファイルの配布方法はダウンロード形式でも P2P 形式でも可.

(4) ユーザは配布ファイルに含まれる暗号化されたライセンスキー $K_{sp}(K_L)$ と作成者の ID, 許可ユーザのリスト, CBS の公開鍵 Kcp をサーバに送信.

(5) サーバはユーザが CBS を用いていることと許可ユーザであることを確認.

(6) 課金する場合は ID とユーザの情報を用いて決済処理を実行.

(7) 秘密鍵 Kss で K_L を復号した後, 受信した Kcp で K_L を再暗号化して返送.

(8) ユーザはライセンスキーを Kcs で復号した後, コンテンツを K_L で復号して閲覧.

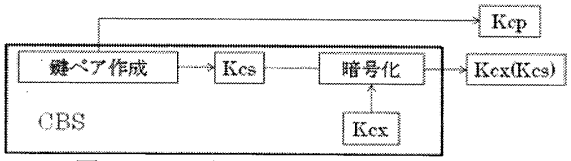

図 3. CBS による公開鍵ペアの作成

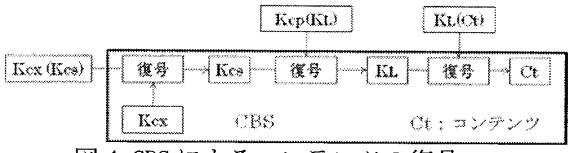

図 4. CBS によるコンテンツの復号

3.4. CBS による暗号化

CBS 内のソフトウェア耐タンパ性を高めるため, CBS は初回起動時の鍵ペア作成とコンテンツの復号を行う際に秘密鍵 Kcs の暗号化・復号化処理をする.

図 3 に示すように鍵ペア作成において公開鍵 Kcp はそのまま出力されるが, 秘密鍵 Kcs は CBS 内の共有鍵 Kcx で暗号化されてから出力される.

図 4 に示すようにコンテンツ復号時は Kcx により Kcx(Kcs)を Kcs に復号, Kcs により Kcp(K_L)を K_L に復号, K_L を用いてコンテンツを復号という 3 段階の復号処理をする.

4. CBS の安全性

CBS のソフトウェア耐タンパ性を保つために CBS に埋め込まれている Kcx とそれを用いる暗号アルゴリズムの抽出防止が必要である.

そのためにプログラムの難読化を実施する. これにより, 攻撃者によるプログラムの動作や定数の解析を妨げ, 安全性を保つのである.

5. CBS の安全性の評価

ここでは, 逆コンパイラを用いてプログラム内の Kcx, 及び暗号アルゴリズムを解析できるかを確かめた.

C++で作成したプログラムを難読化ソフトで難読化し, 逆コンパイルを行った結果, Kcx, 暗号アルゴリズムともに推測が難しいということが確かめられた.

難読化ソフトと逆コンパイラは既存ソフトの Dotfuscator と Reflector for .NET をそれぞれ用いた.

6. おわりに

ここでは, コンテンツ作成者にとって扱い易いシステムを提案し, コンテンツ保護が可能なことを確かめた. ソフトウェアの耐タンパ性確保には従来技術を用いたため, 完全な安全性を保っているとは言い難い. また, 動的解析に対する耐タンパ性の評価, CBS に復号したコンテンツを保護するビューアの実装が求められる.