ユーザーのプライバシーを保護するデジタル著作権管理システム

A Digital Rights Management System Protecting Users' Privacy

荒 川 淳 平☆ 小 嶋 徹 也☆ 青 野 正 宏☆

Jumpei Arakawa, Tetsuya Kojima and Masahiro Aono

1. はじめに

近年，音楽や映像など様々なコンテンツがデジタル化されたことにより，それらのデジタルコンテンツに対する改ざんや不正コピーといった問題が生じている．これに対処するものとして，デジタル著作権管理（*Digital Right Management; DRM*）と呼ばれる技術が注目を集めている．しかし，現在提案されている多くのDRMシステムは，公開鍵暗号を用いることで正規ユーザー以外のコンテンツの利用を防ぐ方法を採っており，コンテンツの購入や利用に際して，ユーザーは識別情報となりうる公開鍵をコンテンツの販売者に送信しなければならない．このため，ユーザーの購入履歴や利用情報などのプライバシーが侵害される恐れがある．そこで本稿では，ブラインド署名に代表されるプライバシーを考慮に入れた暗号技術を応用することによって，コンテンツ販売者と公開鍵の認証機関が結託したとしても，ユーザーのプライバシーが露顕することなく，安全性と汎用性にも優れた，現実的なデジタル著作権管理システムを提案する．

2. システム概要

2.1 構成要素

本稿で提案するシステム（以下，本システム）は，コンテンツに関する権利を持つ権利者（*Right Holder; RH*），コンテンツの利用者（*User*），証明書の認証局（*Certificate Authority; CA*），不正発覚時に対処する司法局（*Judicial Authority; JA*）によって構成される（**図 1**）．ただし，本システムにおいて，権利者と利用者は，本質的に同等であり，その意味で両者をまとめてピア（*Peer*）と呼ぶ．ピアはセキュアトークンと呼ぶ耐タンパー装置を持つものとする． また，セキュアトークンは識別子 *ID* を持ち，認証局は署名の検証用の公開鍵 K_{P-CA} および安全パラメータ N，証明書破棄リスト，不正 ID リストを，司法局は識別子暗号化用の公開鍵 K_{P-JA} をそれぞれ公開しているものとする．

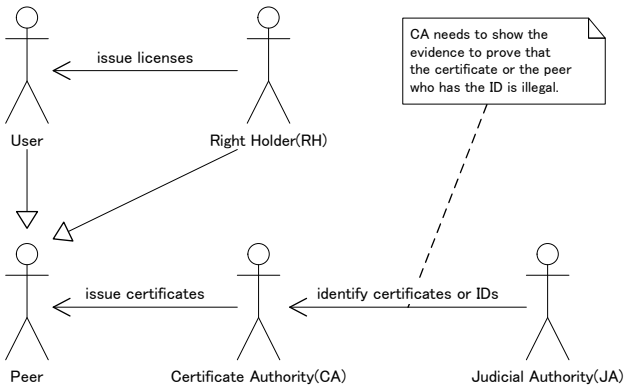

図 1 システム構成

2.2 暗号技術

本システムで利用する暗号技術の記号とそれらの満すべき性質を以下に定義する．ただし，実際の構成方法や性質の数学的証明は省略する．

公開鍵暗号: K_S を秘密鍵（署名鍵），K_P を公開鍵（検証鍵），m を平文，c を暗号文，σ を署名とする．

- 暗号化関数 $c = \mathcal{E}(K_P, m)$
- 署名関数 $\sigma = \mathcal{S}(K_S, m)$
- 復号化関数 $m = \mathcal{D}(K_S, c)$
- 検証関数 $\mathcal{V}(K_P, \sigma) = 1$（when σ is valid.）

ブラインド署名: K_S を署名鍵（秘密鍵），K_P を検証鍵（公開鍵），m を署名対象，σ を署名，r を乱数とする．

- ブラインド化関数 $m' = Blind(K_P, m, r)$
- 署名関数 $\sigma' = Sign(K_S, m')$
- ブラインド化解除関数 $\sigma = Unblind(K_P, \sigma', r)$
- 検証関数 $Verify(K_P, m, \sigma) = 1$（when σ is valid.）

ただし，r を知らずに m' から m を，σ' から σ を推測することは困難である．

確率暗号: K_S を秘密鍵，K_P を公開鍵，m を平文，c を暗号文，r を乱数，t を変換パラメータとする．

- 暗号化関数関数 $c = Encrypt(K_P, m, r)$
- 変換関数 $c' = Transform(K_P, c, t)$
- 復号化関数 $m = Decrypt(K_S, c) = Decrypt(K_S, c')$

ただし，対の平文 (m_0, m_1) とそのいずれかの暗号文（c_0 または c_1）が与えられた場合，それがどちらの平文を暗号化したものかを判断することは困難である．

検証符号: x を元（符号化対象），c を符号，r を乱数，t を変換パラメータとする．

- 符号化関数 $c = Encode(x, r)$
- 変換関数 $c' = Convert(c, t)$
- 比較関数 $Compare(y, c) = Compare(y, c') = 1$（when $y = x$）

ただし，c や c' から x を求めることや，t を知らずに，c と c' が同一の元 x の符号であることを推測することは困難である．

段階的秘密交換プロトコル: 段階的秘密交換プロトコルは，段階的にお互いの秘密を交換しあうことを可能にする．各段階で交換する情報は秘密全体を漏らすことなく，その秘密の一部であることを証明できる．部分的な秘密をすべて交換した時点で始めて，相手の完全な秘密を得ることができる．

3. システム仕様

表 1 と**表 2** にシステムで使用する証明書と契約書の定義を示す．

3.1 証明書の発行

セキュアトークンがベースとなる証明書 $C_0 = (V, K_{P-0}, T, E_0, D_0, s_0, c_0)$ と $e_0 = Encrypt(K_{P-JA}, ID, r)$ を保有しているとして，新しい証明書を発行する手順を以下に示す．（ただし，f と g は一方向ハッシュ関数．）

☆ 国立東京工業高等専門学校
Tokyo National College of Technology

表 1 証明書の定義

項目	説明
規格情報 V	証明書の規格を示す情報
公開鍵 K_P	対象となる公開鍵
有効期間 T	証明書の有効期間
暗号化識別子 E	$E = \mathcal{E}(K_{P-CA}, e)$，ただし $e = Encrypt(K_{P-JA}, ID, r)$
識別検証符号 D	$D = Encode(ID, r')$
署名 σ	内容を証明する署名
検証補助情報 c	証明書の検証に必要な情報

表 2 契約書の定義

項目	説明
契約値 α	秘密の値 a を契約関数 f に入力したときの出力
契約関数 f	契約に使用する一方向関数
公開鍵 K_P	契約対象者の公開鍵
宣言文 M	“公開鍵 K_P と対応する秘密鍵を有するものが，$f(a) = \alpha$ なる a を提示すれば，データ D を所有することを認める．”
データ D	契約対象となるデータ

step1: セキュアトークン内で新しい鍵ペア (K_{S-new}, K_{P-new}) を作成する．

step2: セキュアトークンは乱数 $(r_k, u_k, v_k, w_k, x_k)$ $(1 \le k \le N)$ を選び，次の計算を行う．

$$e_k = Transform(K_{P-JA}, e_0, u_k)$$
$$E_k = \mathcal{E}(K_{P-CA}, e_k)$$
$$D_k = Convert(D_0, v_k)$$
$$S_k = f(K_{P-new} \| w_k) \| g(V \| T \| E_k \| D_k)$$

これらをまとめて次を計算し，

$$\alpha_k = Blind(K_{P-CA}, S_k, r_k)$$
$$\beta_k = Encrypt(K_{P-JA}, r_k \| S_k, x_k)$$

以上をすべてまとめて次を算出する．

$$\sigma_{Token} = \mathcal{S}(K_{S-0}, (\alpha_1 \| \beta_1) \| ... \| (\alpha_N \| \beta_N) \| E_0 \| D_0 \| M)$$

step3: ピアはすべての (a_k, b_k) と σ_{Token} および C_0 を CA に送る．

step4: CA は C_0 を検証し，署名 σ_{Token} を確認する．不正が認められなければ，CA は $k'(1 \le k' \le N)$ を選び，ピアに送る．

step5: ピアは $k \ne k'$ となるすべての k に対して，$(r_k, u_k, v_k, x_k, f(K_{P-new} \| w_k))$ を送る．

step6: CA は step2 に従い，α_k と β_k を計算し，step3 でピアが送ってきたものを一致することを確認する．ただし，e_0 は E_0 を秘密鍵 K_{S-CA} で復号化して得る．不正がなければ，$\sigma' = Sign(K_{S-CA}, \alpha_{k'})$ を計算し，$(\sigma', E_0, \alpha_{k'}, \beta_{k'})$ を保存する．

step7: セキュアトークンは新しい証明書 $C_{new} = (K_{P-new}, V, T, E, D, \sigma, c)$ を次のように計算し，$e_{k'}$ を保存する．

$$E = E_{k'}$$
$$D = D_{k'}$$
$$\sigma = Unblind(K_{P-CA}, \sigma', r_{k'})$$
$$c = w_{k'}$$

3.2 証明書の検証

証明書の有効性を検証する手順を以下に示す．

step1: 規格情報 V や有効期間 T が有効かどうか確認する．

step2: 署名 σ が証明書破棄リストに含まれていないか確認する．

step3: $Verfy(K_{P-CA}, f(K_P \| c) \| g(V \| T \| E \| D)) = 1$ を確認する．

step4: 不正 ID リストに含まれるすべての ID について，$Compare(ID, D) = 1$ が成り立たないことを確認する．

3.3 証明書また ID の特定

認証局が不正である十分な証拠を示した場合，司法局が証明書または ID を特定する手順を以下に示す．

証明書の特定: CA は保存していた α と β を JA に送る．JA は β を符号化して r を得て，α のブラインド化を解除し，σ を計算し，CA に送る．

***ID* の特定:** CA は保存していた E を復号化して e を得て JA に送る．JA は e を復号化して ID を得て，CA に送る．

3.4 ライセンスの発行

利用者の利用する公開鍵を K_{P-User}，権利者の利用する公開鍵を K_{P-RH} として，ライセンスを発行する手順を以下に示す．

step1: 利用者と権利者は公開鍵の証明書を交換し，相手に不正がないことを確認する．

step2: 利用者は，データ D にコンテンツの対価 C を入れ，契約書 $\mathcal{X}_{User} = (\alpha = f(a), f, K_{P-RH}, M, C)$ を作成し，それに署名をし，K_{P-RH} で暗号化して権利者に送る．

step3: 権利者のセキュアトークンは契約書 $\mathcal{X}_{User}$ に対する署名を確認すると，$\mathcal{X}_{User}$ を一時的な記憶領域に格納する．

step4: 権利者は，データ D にコンテンツのライセンス $\mathcal{L}$ を入れ，契約書 $\mathcal{X}_{RH} = (\beta = g(b), g, K_{P-User}, M, \mathcal{L})$ を作成し，それに署名をし，K_{P-User} で暗号化して利用者に送る．

step5: 利用者のセキュアトークンは契約書 $\mathcal{X}_{RH}$ に対する署名を確認すると，$\mathcal{X}_{RH}$ を一時的な記憶領域に格納する．

step6: 利用者と権利者は秘密の値 a と b を段階的秘密交換プロトコルを利用して交換する．

step7: セキュアトークンは，秘密の値を確認すると，一時的な記憶領域のデータを通常の記憶領域へと移動させる．

4. 考　　察

4.1 安　全　性

- 不正なピアの証明書や ID は，認証局と司法局が協力することで特定し，無効化することができる．
- 段階的秘密交換プロトコルを用いることで，ライセンスまたは対価の持ち逃げを防ぐことができる．

4.2 プライバシー

- ピアは自由に新しい証明書を入手できるので，公開鍵を識別情報とすることはできない．
- ブラインド署名を利用しているので，認証局であっても証明書を特定できない．
- 暗号化識別子 E は認証局と司法局の鍵で二重に暗号化されているため，司法局が検閲し ID を入手することはできない．

5. ま　と　め

本稿では，安全性を確保しつつもユーザーのプライバシーを保護することが可能なデジタル著作権管理システムを提案した．

今後は，現実的なシステムを実装し，処理時間や通信効率などの測定・検証を進める予定である．

参　考　文　献

1) 佐藤直之, 鈴木英明: 耐タンパ個人端末を利用し個人情報の保護を可能とした認証方式, 情報処理, Vol. 41, No. 8, pp. 2129–2137 (2000).
2) 岡本龍明, 山本博資: 現代暗号, 産業図書 (1997).
3) 情報処理振興事業協会, 通信・放送機構: 暗号技術評価報告書（2002 年度）, http://www.ipa.go.jp/security/enc/CRYPTREC/ にて配布 (2003).