RSA暗号の公開鍵への所有者情報埋め込み手法とその著作権管理システムへの応用

北原, 基貴
九州大学大学院システム情報科学府情報学専攻

穴田, 啓晃
九州先端科学技術研究所

川本, 淳平
九州大学大学院システム情報科学研究院

櫻井, 幸一
九州大学大学院システム情報科学研究院 | 九州先端科学技術研究所

出版情報：情報処理学会CSEC研究発表会. 第65回, 2014-05-22
バージョン：
権利関係：

所有者情報の公開鍵への埋め込み手法とその著作権管理システムへの応用

北原 基貴[1,a)]　穴田 啓晃[2,b)]　川本 淳平[3,c)]　櫻井 幸一[3,2,d)]

概要：公開鍵暗号では，ユーザの公開した公開鍵に対するなりすましを防ぐため，公開鍵の正当性を保証する証明書が信頼できる認証局によって添付される．この仕組みは公開鍵基盤と呼ばれる．公開鍵と証明書は誰もが参照可能な公開鍵ディレクトリに保存される．送信者はこれらの情報を用いて認証・暗号化を行う．この証明書を必要としない暗号として，ID ベース暗号がある．ID ベース暗号ではユーザの ID が公開鍵として扱われる．秘密鍵は，秘密情報を持つ鍵配付センターから，自身の証明を行うことで受け取る．これまでの ID ベース暗号には鍵配付センターがユーザの使う秘密鍵を知ってしまうという鍵供託問題が存在する．本論文では，所有者情報と証明書を公開鍵に埋め込むことで，鍵供託問題のない ID ベース暗号に相当するシステムを提案する．提案システムでは，正当な ID を持つユーザ以外は ID を埋め込み不可にするため，公開鍵への所有者情報の改ざんが行われた場合にそれを検知できる．また，証明書添付の必要がない．更に，提案システムの著作権管理システムへの適用を提案する．コンテンツ提供者の公開鍵に証明書を埋め込むことにより，著作権管理システムを，コンテンツ提供者による不正売買や分配を検証可能なものにすることができる．

1. はじめに

本稿で我々は公開鍵暗号の公開鍵にユーザの ID 情報を埋め込むことを論じる．

公開鍵暗号の利用の際には公開鍵の正当性を検証する必要があるため，通常，その鍵が誰のものであるかを保証する証明書が信頼できる認証局によって添付される．公開鍵を受け取ったユーザはこの証明書の署名を検証することで，その正当性を検証できる．このような機構は公開鍵基盤と呼ばれる．現在用いられている公開鍵基盤では，多くの場合公開鍵ディレクトリと呼ばれる，誰もが参照できる場所に公開鍵と証明書を保存する．証明書なしで正当性を検証できる暗号として，ユーザの ID を公開鍵として用いる ID ベース暗号がある．ここでの ID は，ユーザを特定できる一意の値とする．しかし現在知られている実現法では，鍵配付センターの秘密情報を使わなければ秘密鍵を作ることができないため，鍵配布センターがすべてのユーザの秘密鍵を知ってしまうという鍵供託問題がある．

[1] 九州大学大学院システム情報科学府情報学専攻
[2] 公益財団法人九州先端科学技術研究所
[3] 九州大学大学院システム情報科学研究院
[a)] kitahara@itslab.inf.kyushu-u.ac.jp
[b)] anada@isit.or.jp
[c)] kawamoto@inf.kyushu-u.ac.jp
[d)] sakurai@csce.kyushu-u.ac.jp

ID を公開鍵に埋め込むことで，ID ベースとした公開鍵も考えられている．RSA 暗号では公開鍵に情報を埋め込む手法が提案されており [12][10]，ID を埋め込むことで鍵供託問題のない ID ベース暗号を目指している．しかし，ID を埋め込むという作業自体は誰でも可能であり，なりすましの問題を回避できていない．本研究では既存の埋め込み手法を拡張し，公開鍵に対するなりすましを防ぐ手法を提案する．

2. 先行研究

本節では，本稿に係わる先行研究を振り返る．公開鍵基盤，ID ベース暗号，Self-certified PKC，そして RSA 暗号の公開鍵の法への情報の埋め込みについてまとめる．最後に Lenstra のアルゴリズムの改良版を説明する．

2.1 公開鍵基盤

公開鍵暗号では，公開鍵は誰でも取得できる公開鍵ディレクトリに預けられ，保存される．例えば Alice へ暗号文を送りたいユーザは公開鍵ディレクトリから Alice の公開鍵を取得し，暗号化を行うこととなる．しかし基本的に公開鍵は秘密鍵をもとに計算されるため，公開鍵を見ただけでは誰のものなのかを知ることはできない．ユーザが公開鍵ディレクトリに公開鍵を預ける際に名前を一緒に登録し

たとしても，その名前を偽造することにより，他人に送られたはずの暗号文を復号できるという問題が発生してしまう．この問題を解決するため，実際の運用では公開鍵基盤が用いられている．[*1]

2.2 ID ベース暗号

ID ベース暗号は，Shamir が 1984 年に初めてその概念を提案した．ID ベース暗号は ID を公開鍵とみなす暗号方式である．メッセージの送信者は受信者の ID を用いて暗号化を行い，暗号文を受信者に送信する．受信者は任意のタイミングで自分の ID に対応する秘密鍵を信頼できるセンターから受け取っておき，受け取った秘密鍵を用いて暗号文を復号する．これが ID ベース暗号の考え方である．ID ベース暗号には，鍵配付センターがユーザ全員の秘密鍵を知ってしまうという鍵供託問題が存在する．これは，ユーザに対して送られた暗号文を鍵配付センターが解読可能ということを意味するため，セキュリティの観点，プライバシーの観点からは好ましくない．[*2]

2.3 Self-certified PKC

上記 ID ベース暗号に似た機能を持つ Self-certified PKC を説明する．1991 年に Giraut が self-certified public keys という暗号を提案した [5]．この暗号では公開鍵自体が検証されており，別途証明書を用意する必要がないという利点がある．この暗号では秘密鍵はユーザが作り，その情報をもとに公開鍵管理センターが公開鍵を作成する．公開鍵管理センターだけが持つ情報を用いないと公開鍵を作成することはできないため，攻撃者は有効な公開鍵を作ることができない．よって認証を行わずに公開鍵を使うことができる．もし攻撃者によって公開鍵を入れ替えられていたとしても，攻撃者が暗号文を復号できないため問題ないという考え方である．

2.4 RSA 暗号の公開鍵の法への情報の埋め込み

RSA 暗号の variation として，公開鍵 N の一部に予め決めておいた既定値を埋めこませるというものがある．これを利用することで様々な情報を埋め込むことができる．Canstone と Zuccherato が最初にこの考え方を実際のアルゴリズムを用いて提案した [12]．埋め込むビットの長さにより生成効率が変化し，N の半分のビット長の情報を埋め込む方式では既定値の因数分解が必要で，このため速度が遅くなってしまう問題があった．N のビット長のうち 1/4 までの情報を埋め込む条件ならば高速に埋め込むことができており，埋め込む長さと鍵生成速度はトレードオフの関係にあった．その後，Lenstra によって N の半分のビット長の値を指定でき，実行時間の観点からも効率的なものが提案された [10]．この手法では鍵生成の速度を RSA 暗号での鍵生成の速度とほぼ等しくすることができる．

2.4.1 Lenstra のアルゴリズム

Lenstra は 2 素数の積からなる RSA 公開鍵 N に効率的に既定値を埋めこませる事のできるアルゴリズムを提案した?．この論文中では公開鍵の中の様々な部分に予め決めておいた値を埋め込む手法を提案しているが，ここではそれらのアルゴリズムのうち，公開鍵 N の上位ビット部分に埋め込む手法を紹介する．

公開鍵に埋め込む情報を I とする．$g \in \mathbb{Z}$ s.t. $|g| = c,\ c|(\lambda - |I|)$，を固定する．$L = (\lambda - |I|)/c$ と置く．

(1) $N' = Ig^L$ を計算する．
(2) 素数 p s.t. $|p| < L$，をランダムに選択する．
(3) pq' が N' に最も近くなるような正整数 q' を計算する．
(4) $q'+t$ の値が素数となる最小の正整数 t を探し，$q = q'+t$ とおく．
(5) $N = pq$ を計算する．
(6) N の上位ビットが I であるならば p, q, N を返す．そうでないならば (2) に戻る．

2.4.2 改良 Lenstra アルゴリズム

第 2.4.1 節の Lenstra のアルゴリズムに対し，我々は $|p| = |q| = \lambda/2,\ |N| = \lambda$ と出来る改良版を提案する．なお，λ は偶数とする．

公開鍵に埋め込む情報を I とする．

(1) $N' = I \parallel 00\cdots0$ s.t. $|N'| = \lambda$ を計算する．
(2) 素数 p s.t. $|p| = \lambda/2$，をランダムに選択する．
(3) $q' = \lceil N'/p \rceil$ を計算する．
(4) $q'+t$ の値が素数となる最小の正整数 t を探し，$q = q'+t$ とおく．
(5) $N = pq$ を計算する．
(6) N の上位ビットが I であるならば p, q, N を返す．そうでないならば (2) に戻る．

3. 攻撃モデルと安全性の定義

本節では，本稿で問題とする，RSA 暗号の公開鍵に対するなりすまし攻撃を説明する．当該の攻撃モデルは中間者攻撃の一種である．次いで，ID 情報を法 N へ埋め込み可能な RSA 暗号の，上記の攻撃に対する安全性を定義する．

3.1 記法の準備

セキュリティパラメータを λ とする．ビット列 I のビット長を $|I|$ で表す．ハッシュ関数を $H(\cdot)$ で表す．

本稿で論じる，公開鍵 PK に対する ID 情報の埋め込み

[*1] この機構を最初に考えたのは MIT の学部 4 年生で，発表したのも卒業論文という形だった．しかしこの機構は現実世界で使うには必須であり，その後急速に広まった．
[*2] しかし，全てのユーザを管理するスーパーユーザのような存在が仮定出来る状況においては，鍵供託問題を許容できると考えられる．例えば企業等，全体を管理する必要がある世界においてはアドミニストレータは会社の機密を保持するという観点から全社員の秘密鍵を知っておく必要がある．それゆえ ID ベース暗号は都合がよく，鍵供託を利点として用いることもできる．

に関し，PK から取り出された ID を ID(PK) で表す．

3.2 攻撃モデル

攻撃モデル（中間者攻撃モデル）

(1) $\mathcal{X}$ は，第一フェーズとして $\mathcal{A}$ と通信する．$\mathcal{X}$ は公開鍵 $\mathrm{PK}_{\mathcal{B}}$ を用い，$\mathcal{A}$ と正当な通信を行う．

(2) $\mathcal{X}$ は，第二フェーズとして $\mathcal{B}$ と通信する．$\mathcal{X}$ は公開鍵 $\mathrm{PK}^*_{\mathcal{A}}$ s.t. $\mathrm{ID}(\mathrm{PK}^*_{\mathcal{A}}) = \mathrm{ID}(\mathrm{PK}_{\mathcal{A}}) = \mathrm{ID}_{\mathcal{A}}$，を用い，$\mathcal{B}$ と正当な通信を行う．

(3) $\mathcal{X}$ が，$\mathcal{A}$ に対しては「$\mathcal{B}$ と正当な通信した」と確信させることが出来，かつ，$\mathcal{B}$ に対しては「$\mathcal{A}$ と正当な通信した」と確信させることが出来たとき，$\mathcal{X}$ は勝ちであるものとする．

3.3 安全性の定義

ランダムに選んだユーザ $\mathcal{A}$ とユーザ $\mathcal{B}$ に対し，$\mathcal{X}$ が勝ちとなる確率が，セキュリティパラメータ λ に関し negligible であるとき，RSA 暗号の法への ID 情報の埋め込み手法は安全である，と呼ぶものとする．

4. 提案方式とその安全性

本節では，第 4 節で述べた，RSA 暗号の公開鍵に対するなりすまし攻撃に対し安全な，我々の提案方式を説明する．次いで，我々の提案方式を，第 2 節で説明した先行研究と比較する．

4.1 提案方式のアルゴリズム

自身の公開鍵に認証情報を埋め込む手続き 以下で，guarantor は，既に公開鍵ネットワークに参加しているユーザのうち，これから参加するユーザ $\mathcal{A}$ の公開鍵を発行する保証人のこととする．guarantor は公開鍵ネットワークに参加している任意のユーザがなることが出来るものとする．

(1) ユーザ $\mathcal{A}$ は素数 $p_{\mathcal{A}}$ s.t. $|p_{\mathcal{A}}| = \lambda/2$，をランダムに選択する．

(2) ユーザ $\mathcal{A}$ は $m_{\mathcal{A}} := \mathrm{ID}_{\mathcal{A}} \parallel H(p_{\mathcal{A}})$ を guarantor へ送る．

(3) guarantor は $m_{\mathcal{A}}$ に署名する：
$\sigma_{\mathcal{A}} \leftarrow \mathbf{Sign}(\mathrm{PK}_{\mathrm{gurnt}}, \mathrm{SK}_{\mathrm{gurnt}}, m_{\mathcal{A}})$.

(4) ユーザ $\mathcal{A}$ は，DRM の RSA 公開鍵 $\mathrm{PK}_{\mathrm{gurnt}} = (N_{\mathrm{gurnt}}, e_{\mathrm{gurnt}})$ を用い，素数 $p_{\mathcal{A}}$ の RSA 暗号文を生成する：
rsa-c($p_{\mathcal{A}}$) ←
$\mathbf{Enc}(\mathrm{PK}_{\mathrm{gurnt}}, p_{\mathcal{A}}) = (p_{\mathcal{A}})^{e_{\mathrm{gurnt}}} \bmod N_{\mathrm{gurnt}}$.

(5) ユーザ $\mathcal{A}$ は，公開鍵 $\mathrm{PK}_{\mathcal{A}}$ に埋め込む情報 I として次のストリングを取る：$I := \mathrm{ID}_{\mathcal{A}} \parallel \sigma_{\mathcal{A}} \parallel$ rsa-c($p_{\mathcal{A}}$).

(6) ユーザ $\mathcal{A}$ は，素数 $p_{\mathcal{A}}$ を用い，I を埋め込んだ公開鍵 $\mathrm{PK}_{\mathcal{A}}$ 及び対応する秘密鍵 $\mathrm{SK}_{\mathcal{A}}$ を，改良 Lenstra アルゴリズムで生成する．

閲覧した公開鍵の正当性を検証する手続き 以下で，ユーザ $\mathcal{B}$ は，ユーザ $\mathcal{A}$ の公開鍵の正当性を検証する．

(1) ユーザ $\mathcal{B}$ は，ユーザ $\mathcal{A}$ の公開鍵 $\mathrm{PK}_{\mathcal{A}}$ から埋め込み情報 I を取り出す．

(2) ユーザ $\mathcal{B}$ は，RSA 公開鍵 $\mathrm{PK}_{\mathrm{gurnt}} = (N_{\mathrm{gurnt}}, e_{\mathrm{gurnt}})$ を用い，ユーザ $\mathcal{A}$ の公開鍵 $\mathrm{PK}_{\mathcal{A}}$ の RSA 法 $N_{\mathcal{A}}$ の暗号文を生成する：
rsa-c($N_{\mathcal{A}}$) ←
$\mathbf{Enc}(\mathrm{PK}_{\mathrm{gurnt}}, N_{\mathcal{A}}) = (N_{\mathcal{A}})^{e_{\mathrm{gurnt}}} \bmod N_{\mathrm{gurnt}}$.

(3) ユーザ $\mathcal{B}$ は，rsa-c($N_{\mathcal{A}}$) が rsa-c($p_{\mathcal{A}}$) で割り切れるか否かを検証する．
割り切れるならば，True を返す．
割り切れないならば False を返す．

4.2 提案方式の安全性

定理 4.1 第 5.1 節のアルゴリズムで記述された提案方式は，第 4 節の攻撃モデルに対し，安全である．

証明 第 4 節の攻撃モデルの定義から，攻撃者 $\mathcal{X}$ は，次の手順で偽の公開鍵 $\mathrm{PK}^*_{\mathcal{A}}$ を作らざるを得ない．：
攻撃者 $\mathcal{X}$ は，ハッシュ関数 $H(\cdot)$ の終域から，ストリング r.string を一様ランダムに選び，メッセージ $m^*_{\mathcal{A}} := \mathrm{ID}_{\mathcal{A}} \parallel$ r.string を guarantor へ送る．guarantor は $m^*_{\mathcal{A}}$ に署名する:
$\sigma^*_{\mathcal{A}} \leftarrow \mathbf{Sign}(\mathrm{PK}_{\mathrm{gurnt}}, \mathrm{SK}_{\mathrm{gurnt}}, m^*_{\mathcal{A}})$.
攻撃者 $\mathcal{X}$ は, $\mathrm{ID}_{\mathcal{A}}$ 及び rsa-c($p_{\mathcal{A}}$) をユーザ $\mathcal{A}$ の公開鍵 $\mathrm{PK}_{\mathcal{A}}$ から取り出し，下記の情報 I^* が埋め込まれた公開鍵 $\mathrm{PK}^*_{\mathcal{A}}$ の法 $N^*_{\mathcal{A}}$ を，第 2.4.2 節の改良版 Lenstra アルゴリズムで生成する．

$$I^* := \mathrm{ID}_{\mathcal{A}} \parallel \sigma^*_{\mathcal{A}} \parallel \text{rsa-c}(p_{\mathcal{A}}). \tag{1}$$

攻撃者 $\mathcal{X}$ は，ユーザ $\mathcal{B}$ に対し $\mathrm{PK}^*_{\mathcal{A}}$ がユーザ $\mathcal{A}$ の正当な公開鍵であることを確信させようとする．ところが, アルゴリズム **Sign** のランダムネスにより，rsa-c($N^*_{\mathcal{A}}$) が rsa-c($p_{\mathcal{A}}$) で割り切れる確率は λ に関し negligible である． □

4.3 提案方式と他の方式との比較

表 1 に我々の提案方式と他の方式との比較をまとめる．

5. 著作権管理システムへの応用

本節では, 第 5 節の提案方式の, 著作権管理システム（Digital Rights Management System．以下，DRM-System と呼ぶ）への応用例を示す．

5.1 著作権管理システム

DRM を使用する場合，利用者が購入できるコンテンツは暗号化されたものとなる．この暗号化されたコンテンツを利用する際の手順は以下となる．1. 正規の購入者は直接コンテンツ配布者のサーバと通信を行い，自身が正規の購入者であると認証を行う．2. 正規の購入者であると証

	公開鍵基盤	ID ベース暗号	Self-certified	提案手法
鍵供託問題	なし	あり	なし	なし
送信者の公開鍵取得	必須	不要	必須	必須
公開鍵証明書作成	必須	不要	不要	必須
公開鍵証明書添付	必須	不要	ID が必要	不要
公開鍵の検証	必須	不要	不要	必須
公開鍵の検証者	誰でも可能	誰でも可能	誰でも可能	センターのみ
公開鍵の秘匿性	なし	なし	あり	あり

表 1 我々の提案方式と他の方式との比較
Table 1 Comparison with Our Proposal Scheme and Previous Schemes

明できたら，コンテンツ配布者のサーバは購入者が持つパソコンのコンテンツ再生用ソフトウェアに対して復号鍵を送る．この際，正規の購入者であっても復号鍵を自由に使うことはできない．3. コンテンツを利用する際，利用者はコンテンツ再生用ソフトウェアのみを用いてコンテンツを利用する．コンテンツ再生用ソフトウェアは自身が持つ復号鍵を利用し，コンテンツを復号しつつ再生する．4. コンテンツの再生が終了したら，コンテンツ再生用ソフトウェアは復号鍵を廃棄する．以上の通り，正規の利用者でもコンテンツを復号しつつ再生するため，暗号化されていないコンテンツを得ることはできない，このことにより，暗号化されていないコンテンツの二次配布を防ぐ事ができるといえる．この DRM の問題点として，コンテンツ配布者のサーバに掛かる負担が大きいという点が存在する．正規の利用者がコンテンツを利用したいと思った場合，復号鍵を得るためにこのコンテンツ配布者のサーバと通信することが必要となる．利用者の数が増加すればするほどこの通信は増え，認証にかかる負担も増加する．もしこのサーバが負担に負け，落ちてしまった場合，全ての正規の利用者はコンテンツを再生することができなくなる．この点は DRM を利用しない場合と比べて正規の利用者に負担をかける，利便性を低下させるということであり，大きな問題といえる．1.3 関連研究 DRM には，通信方式を元にするとサーバ・クライアント型方式と P2P 型方式の二つが提案されている．本章では，これらの型での実現手法と違いについて検討する．DRM では，以下の二段階の暗号化を行うことで，コンテンツの暗号化を行う．

コンテンツを暗号化する．

暗号化されたコンテンツに対し，コンテンツを利用するための条件とコンテンツを扱うプログラムを付加し，カプセル化する．

P2P を利用した DRM 方式では，コンテンツ自体の配布は各ユーザがそれぞれで行うことができるため，管理サーバに依存する必要がない．DRM 管理サーバはそれぞれの利用者が持っているコンテンツに対して，金銭を受け取ることと引き換えに利用許可を与えることのみを行う．また，P2P 型 DRM システムでは，DRM 管理サーバに対して自分が著作権を持つコンテンツを登録することで，任意のユーザがコンテンツ提供者になることも出来るという利点がある．コンテンツ提供が個人でも容易に可能になるだけでなく，信頼できる大きなシステムが一つあれば，各企業がコストをかけて自前のシステムを用意する必要もなくなる．DRM の実装の一例として，ユーザの ID を用いて検証を行うという手法が考えられている [2]．コンテンツ再生用ソフトウェアを，コンテンツの初回利用時に自分の ID を管理サーバに送るような仕様にし，管理サーバは登録されている ID に対して後日代金を請求するというものである．この手法では，利用条件は ID を送ることとなる．このような仕様にすることで，自由に配布することが可能となる．コンテンツを希望するユーザが一度に大量に出たとしても，コンテンツ保持者から P2P を利用してダウンロードが可能であり，サーバに負担をかけることもなくなる．予め金銭を送る必要がないため利用条件はないが，コンテンツの不正利用を防ぐことに関しては，コンテンツの初回起動時に自分の ID を送る，という点にかかっているため，この部分を迂回してコンテンツが起動できてしまわないよう暗号化・カプセル化を施している．また，この手法では P2P を利用した DRM に関して，サーバ・クライアント型に基づいた方式と，認証 DB を個人が管理する分散 P2P ベースの DRM システム，そして P2P をベースとして DRM 管理サーバに認証 DB を任せた半分散 P2P システムの三種類が提案されている．また，それぞれで第三者に売買するシステムについても言及されている．どの方式においてもコンテンツをカプセル化するための DRM 管理サーバの導入が必要となる．コンテンツ提供者はこの DRM 管理サーバに対してコンテンツと利用するための条件を送り，カプセル化されたコンテンツを受け取ることになる．このカプセル化されたコンテンツを別のユーザに配り，別ユーザは利用権限を得ることでこのカプセル化されたコンテンツを利用できるようになるという流れである．それぞれの手法の違いについては，以下となる．サーバ・クライアント型に基づいた方式では，認証済みユーザの DB を DRM 管理サーバが管理する．カプセル化されたコンテンツを得たユーザはこの DRM 管理サーバと通信を

行い，利用許可を得ることになる．この手法と従来のサーバ・クライアント型の違いは，カプセル化されたコンテンツはユーザ間で自由に受け渡しができるという点にある．このため，コンテンツの配布自体は一極集中することなく行うことができるが，利用者が増えるに従い利用許可を得る点で DRM 管理サーバにアクセスが集中してしまうという問題点や，コンテンツ提供者による配布状況の認識の困難性といった問題点がある．分散型 P2P ベースの DRM システムでは，認証済みユーザの DB をコンテンツ配布者が管理する．カプセル化されたコンテンツを得たユーザはコンテンツ配布者と通信を行い，利用許可を得ることになる．コンテンツ配布者は好きな DRM 管理サーバを用いることが出来るため，コンテンツ配布者が増えたとしても問題はなく，完全な P2P 型での DRM システムが構築できているといえる．この方式では，利用許可をユーザが与えることになるため，金銭との取引なども個人で行う必要がある．また，全体を見渡すことの出来るユーザがいないため，システムの使用状況などが不鮮明になるという欠点もある．半分散 P2P ベースの DRM システムでは，認証済みユーザの DB を DRM 管理サーバが管理する．カプセル化されたコンテンツを得たユーザはコンテンツ配布者と通信を行うが，その際にコンテンツ配布者は DRM 管理サーバと通信を行い，認証と課金に関するリクエストを送る．コンテンツ配布者がそれぞれ DRM 管理サーバと通信を行うため，利用者の増加に対する耐性も高く，課金管理を管理サーバに任せることが出来るというのが利点である．サーバ・クライアント型の利点を継承しつつ，可用性をできるだけ高めた手法といえる．このようにある程度人数が増えても対処できるような P2P システムである岩田らの手法であるが，完全な P2P システムという点から利用状況の把握が難しく，またコンテンツ配布に対する利点の薄さから，実現性に関してはあまり考えられていないという問題が存在する．1.4 貢献本論文では，P2P をベースとした新たな DRM システムの考案を行う．既存の手法に対し，認証の方式においてユーザ間通信のみでセキュリティを保つ Bitcoin に注目し，同様のシステムを取り入れる．Bitcoin は電子現金の一種であり，自分の持つコインを二回使用することができないよう，常にチェックが行われる．この方式を利用することで，誰がどう配布したかを明確にでき，二次配布者に利点を与えることや全体の流通状況の把握といったことが可能となる．1.5 先行研究との比較ここでは，先行研究である P2P ベースの手法と比較を行う．先にあげた岩田らの手法では，利用したユーザの ID を収集し，この ID をもとに認証を行う [2]．誰がどう配布したかに関わらず，コンテンツを利用したユーザの利用のみ可能な手法である．よってシンプルかつ最低限であり，それ以上の拡張性はないものとなっている．本提案手法では，利用券をコインとして受け渡す．このため，コンテンツの購入者によるコンテンツの売却が可能である他，どういった経路でコンテンツが受け渡されたのかのデータの収集も可能である．P2P 通信において，ユーザがコンテンツを配布する上で，利点となることは基本的には存在しない．そのため，アップロードを嫌うユーザが多い現状となってしまっている．よって，コンテンツ配布の経路が明らかであれば，コンテンツをより多く配布したユーザに利点をつけることも可能となり，より実用性が高まると考えられる．提案方式と比較したものを表 1 に表す．提案手法ではサーバに対する負荷という点では P2P 型に準じたものとなっており，流通状況を可視化が可能であるという利点がある

5.2 P2P 型 DRM system

DRM を使用する場合，利用者が購入できるコンテンツは暗号化されたものとなる．この暗号化されたコンテンツを利用する際の手順は以下となる．1. 正規の購入者は直接コンテンツ配布者のサーバと通信を行い，自身が正規の購入者であると認証を行う．2. 正規の購入者であると証明できたら，コンテンツ配布者のサーバは購入者が持つパソコンのコンテンツ再生用ソフトウェアに対して復号鍵を送る．この際，正規の購入者であっても復号鍵を自由に使うことはできない．3. コンテンツを利用する際，利用者はコンテンツ再生用ソフトウェアのみを用いてコンテンツを利用する．コンテンツ再生用ソフトウェアは自身が持つ復号鍵を利用し，コンテンツを復号しつつ再生する．4. コンテンツの再生が終了したら，コンテンツ再生用ソフトウェアは復号鍵を廃棄する．以上の通り，正規の利用者でもコンテンツを復号しつつ再生するため，暗号化されていないコンテンツを得ることはできない，このことにより，暗号化されていないコンテンツの二次配布を防ぐ事ができるといえる．この DRM の問題点として，コンテンツ配布者のサーバに掛かる負担が大きいという点が存在する．正規の利用者がコンテンツを利用したいと思った場合，復号鍵を得るためにこのコンテンツ配布者のサーバと通信することが必要となる．利用者の数が増加すればするほどこの通信は増え，認証にかかる負担も増加する．もしこのサーバが負担に負け，落ちてしまった場合，全ての正規の利用者はコンテンツを再生することができなくなる．この点は DRM を利用しない場合と比べて正規の利用者に負担をかける，利便性を低下させるということであり，大きな問題といえる．1.3 関連研究 DRM には，通信方式を元にするとサーバ・クライアント型方式と P2P 型方式の二つが提案されている．本章では，これらの型での実現手法と違いについて検討する．DRM では，以下の二段階の暗号化を行うことで，コンテンツの暗号化を行う．

コンテンツを暗号化する．

暗号化されたコンテンツに対し，コンテンツを利用するための条件とコンテンツを扱うプログラムを付加し，カプ

セル化する．

P2P を利用した DRM 方式では，コンテンツ自体の配布は各ユーザがそれぞれで行うことができるため，管理サーバに依存する必要がない．DRM 管理サーバはそれぞれの利用者が持っているコンテンツに対して，金銭を受け取ることと引き換えに利用許可を与えることのみを行う．また，P2P 型 DRM システムでは，DRM 管理サーバに対して自分が著作権を持つコンテンツを登録することで，任意のユーザがコンテンツ提供者になることも出来るという利点がある．コンテンツ提供が個人でも容易に可能になるだけでなく，信頼できる大きなシステムが一つあれば，各企業がコストをかけて自前のシステムを用意する必要もなくなる．DRM の実装の一例として，ユーザの ID を用いて検証を行うという手法が考えられている [2]．コンテンツ再生用ソフトウェアを，コンテンツの初回利用時に自分の ID を管理サーバに送るような仕様にし，管理サーバは登録されている ID に対して後日代金を請求するというものである．この手法では，利用条件は ID を送ることとなる．このような仕様にすることで，自由に配布することが可能となる．コンテンツを希望するユーザが一度に大量に出たとしても，コンテンツ保持者から P2P を利用してダウンロードが可能であり，サーバに負担をかけることもなくなる．予め金銭を送る必要がないため利用条件はないが，コンテンツの不正利用を防ぐことに関しては，コンテンツの初回起動時に自分の ID を送る，という点にかかっているため，この部分を迂回してコンテンツが起動できてしまわないよう暗号化・カプセル化を施している．また，この手法では P2P を利用した DRM に関して，サーバ・クライアント型に基づいた方式と，認証 DB を個人が管理する分散 P2P ベースの DRM システム，そして P2P をベースとして DRM 管理サーバに認証 DB を任せた半分散 P2P システムの三種類が提案されている．また，それぞれで第三者に売買するシステムについても言及されている．どの方式においてもコンテンツをカプセル化するための DRM 管理サーバの導入が必要となる．コンテンツ提供者はこの DRM 管理サーバに対してコンテンツと利用するための条件を送り，カプセル化されたコンテンツを受け取ることになる．このカプセル化されたコンテンツを別のユーザに配り，別ユーザは利用権限を得ることでこのカプセル化されたコンテンツを利用できるようになるという流れである．それぞれの手法の違いについては，以下となる．サーバ・クライアント型に基づいた方式では，認証済みユーザの DB を DRM 管理サーバが管理する．カプセル化されたコンテンツを得たユーザはこの DRM 管理サーバと通信を行い，利用許可を得ることになる．この手法と従来のサーバ・クライアント型の違いは，カプセル化されたコンテンツはユーザ間で自由に受け渡しができるという点にある．このため，コンテンツの配布自体は一極集中することなく行うことができるが，利用者が増えるに従い利用許可を得る点で DRM 管理サーバにアクセスが集中してしまうという問題点や，コンテンツ提供者による配布状況の認識の困難性といった問題点がある．分散型 P2P ベースの DRM システムでは，認証済みユーザの DB をコンテンツ配布者が管理する．カプセル化されたコンテンツを得たユーザはコンテンツ配布者と通信を行い，利用許可を得ることになる．コンテンツ配布者は好きな DRM 管理サーバを用いることが出来るため，コンテンツ配布者が増えたとしても問題はなく，完全な P2P 型での DRM システムが構築できているといえる．この方式では，利用許可をユーザが与えることになるため，金銭との取引なども個人で行う必要がある．また，全体を見渡すことの出来るユーザがいないため，システムの使用状況などが不鮮明になるという欠点もある．半分散 P2P ベースの DRM システムでは，認証済みユーザの DB を DRM 管理サーバが管理する．カプセル化されたコンテンツを得たユーザはコンテンツ配布者と通信を行うが，その際にコンテンツ配布者は DRM 管理サーバと通信を行い，認証と課金に関するリクエストを送る．コンテンツ配布者がそれぞれ DRM 管理サーバと通信を行うため，利用者の増加に対する耐性も高く，課金管理を管理サーバに任せることが出来るというのが利点である．サーバ・クライアント型の利点を継承しつつ，可用性をできるだけ高めた手法といえる．このようにある程度人数が増えても対処できるような P2P システムである岩田らの手法であるが，完全な P2P システムという点から利用状況の把握が難しく，またコンテンツ配布に対する利点の薄さから，実現性に関してはあまり考えられていないという問題が存在する．1.4 貢献本論文では，P2P をベースとした新たな DRM システムの考案を行う．既存の手法に対し，認証の方式においてユーザ間通信のみでセキュリティを保つ Bitcoin に注目し，同様のシステムを取り入れる．Bitcoin は電子現金の一種であり，自分の持つコインを二回使用することができないよう，常にチェックが行われる．この方式を利用することで，誰がどう配布したかを明確にでき，二次配布者に利点を与えることや全体の流通状況の把握といったことが可能となる．1.5 先行研究との比較ここでは，先行研究である P2P ベースの手法と比較を行う．先にあげた岩田らの手法では，利用したユーザの ID を収集し，この ID をもとに認証を行う [2]．誰がどう配布したかに関わらず，コンテンツを利用したユーザの利用のみ可能な手法である．よってシンプルかつ最低限であり，それ以上の拡張性はないものとなっている．本提案手法では，利用券をコインとして受け渡す．このため，コンテンツの購入者によるコンテンツの売却が可能である他，どういった経路でコンテンツが受け渡されたのかのデータの収集も可能である．P2P 通信において，ユーザがコンテンツを配布する上で，利点となることは基本的には存在しない．そのた

め，アップロードを嫌うユーザが多い現状となってしまっている．よって，コンテンツ配布の経路が明らかであれば，コンテンツをより多く配布したユーザに利点をつけることも可能となり，より実用性が高まると考えられる．提案方式と比較したものを表 1 に表す．提案手法ではサーバに対する負荷という点では P2P 型に準じたものとなっており，流通状況を可視化が可能であるという利点がある．

5.3 考察対象とする P2P-著作権管理システム

考察対象とする DRM-System は，Peer-to-Peer（以降，P2P と略す）モデルの一種であり，サーバー-クライアントモデルの対照概念である．特に，P2P モデルの中でもフラットモデルと呼ぶべきものである．以下説明する．

新規ユーザの DRM-System への登録は，新規ユーザへの公開鍵の発行，及び公開鍵の公開鍵リストへの追記によってなされる．初め，DRM-System を生成した 1 人がユーザとなり，自身の公開鍵を自身で発行し，公開鍵リストに追記する．（以降，公開鍵リストへの追記は既に参加しているユーザのみが可能とする．）次いで，新規ユーザ $\mathcal{A}$ は，既に DRM-System に参加しているユーザの内の任意の 1 人に guarantor となるよう依頼する．Guarantor を引き受けた既ユーザは，新規ユーザの公開鍵を発行し，公開鍵リストに追記する．

DRM-System において，コンテンツを作成したユーザは，コンテンツを他のユーザに提供することが出来る．ただしコンテンツの著作権を管理するため，コンテンツリスト *List* が DRM-System において維持される．*List* は DRM-System のどのユーザも閲覧可能なものとし，またどのユーザもコンテンツを登録可能なものとする．

ユーザ $\mathcal{A}$ がタイトル *title* のコンテンツ *content* を作成し，その著作権を登録したい状況を考える．登録するには，まず $\mathrm{ID}_{\mathcal{A}}$，*title* 及び *content* の連接のハッシュ値を計算する：

$$h \leftarrow H(\mathrm{ID}_{\mathcal{A}} \parallel title \parallel content).$$

$\mathcal{A}$ は三つ組 $(\mathrm{ID}_{\mathcal{A}}, title, h)$ をコンテンツリスト *List* に登録する．このため，コンテンツリスト *List* は次のベクトルの集合となる：

$$List = \{(\mathrm{ID}_{\mathcal{A}}, title, h)\}.$$

ユーザ $\mathcal{B}$ が *List* を見，タイトル *title* の *content* を入手したいときは，$\mathcal{B}$ は公開鍵リストから，法 N に $\mathrm{ID}_{\mathcal{A}}$ を含む公開鍵 PK を探索する．$\mathcal{B}$ は，第 5.1 節に示した検証方法で，正しい公開鍵 $\mathrm{PK}_{\mathcal{A}}$ を探し当てることが出来る．$\mathcal{B}$ は $\mathrm{PK}_{\mathcal{A}}$ に対応する唯一のユーザ $\mathcal{A}$ と通信を開始する．$\mathcal{A}$ は *content* を暗号化するための対称鍵を $\mathrm{PK}_{\mathcal{B}}$ で暗号化することが出来る．$\mathcal{B}$ は $\mathrm{SK}_{\mathcal{B}}$ で対称鍵を復号化することが出来る．

6. おわりに

本稿では，所有者情報と証明書を公開鍵に埋め込むことで，鍵供託問題のない ID ベース暗号に相当するシステムを提案した．提案システムでは，正当な ID を持つユーザ以外は ID を埋め込み不可にするため，公開鍵に対する所有者のなりすましを検知できる．また，証明書添付の必要がない．提案システムは，P2P ベースの DRM-System に適用出来る．この適用により，コンテンツ提供者のなりすましを排除することが出来る．

謝辞 第四著者は本研究の推進に関し次の研究費の支援を部分的に受けております．ここに深謝申し上げます．科研費基盤研究 (B)，研究課題番号：23300027,「サイバーシステムにおける内部攻撃脅威に対する評価指標確立と体系的対策研究」.