L-004

二次利用コンテンツの構成集合を識別可能とするデジタル署名方式
E-signature scheme that allows configuration identifies a set of secondary use of content

伊佐 仁　　小出 雅史　　稲村 勝樹　　岩村 恵市
Hitoshi Isa　Masashi Koide　Masaki Inamura　Keiichi Iwamura

１．はじめに

近年，コンピュータおよびネットワーク環境の発達により，You Tube[1]などのように，一般のユーザがコンテンツを作成し，インターネット上でそのコンテンツの流通を行うことができる消費者生成メディア(CGM:Consumer Generated Media)という概念が発生している．このCGMによってインターネット上には様々なコンテンツがあふれている．また，このCGMにおいては，マッシュアップと呼ばれるコンテンツ作成が行われている．マッシュアップとは，複数の異なる提供元のコンテンツを複合させて新しいコンテンツを形作ることである． 1つのコンテンツはマッシュアップにより新たな多くのコンテンツを生み出していくことになる．よって，マッシュアップによるコンテンツでは，ある著作物の二次利用によってコンテンツを生成し，その生成されたコンテンツをさらに二次利用するような過程を経る．この過程を経ることで，引用されたコンテンツに対しては，一次利用・二次利用・・・という階層を与えることができる．そしてここでは，コンテンツ同士で形成された階層を“コンテンツの構造”と呼ぶことにする．また、近年ではマッシュアップのための表記法を規定したクリエイティブ・コモンズ[2]のような活動も始まっている．

デジタルコンテンツの流通に伴い，その著作権保護のためにデジタル著作権管理(DRM:Digital Rights Management)技術が発達してきた．しかし，CGMサービスが拡大するに従ってDRMに対する考え方の変革が起きている．従来のコンテンツ提供では，放送局やDVD制作者のような特定のコンテンツ提供者が存在し，その提供者によるコンテンツの著作権を保護する目的によってDRM技術が存在していた．そのため従来のDRM技術に関する研究ではデジタルコンテンツの再生，一次流通を制限することが念頭に置かれており，暗号化方式や電子透かしを用いた技術が提案されてきた．しかし，CGMサービス上でのコンテンツは，既にネット上で流通しており，著作物であるコンテンツの引用あるいは編集，加工によって新たに生成されるという特徴がある．そこで，従来のコンテンツの二次利用を制限するというDRM技術ではなく，CGMコンテンツに対するマッシュアップを考慮したDRM技術が必要となってきている．

この課題に対し，デジタル署名を用いてマッシュアップを考慮した DRM 技術が検討されている．デジタル署名は電子商取引などで既に実用化されている方式であり，数学的に安全であるということが証明されたアルゴリズムによって実現されている．デジタル署名は，紙に書かれた文章に対する拇印と印鑑が有する機能をデジタル上のデータに対して実現する技術である．また，それを拡張し，複数の署名者によるデジタル署名の作成，およびその署名の検証を効率的に行うことが可能となる多重署名方式の研究も進んでいる．

多重署名方式を用いたコンテンツの二次利用に対する著作権保護方式に関しては既にいくつかの研究報告[3][4][5][8][9]されおり，マッシュアップによるコンテンツの構造が規定できている．しかし，これらの方式では署名者が順次にコンテンツに署名を施していくため，マッシュアップされたコンテンツが再編集される場合には，署名を最初から付け直す必要があり，署名処理に手間が生じてしまう．例に挙げた You

Tube[1]などでは，動画の編集や削除などが頻繁に行われているため，署名処理に手間が生じることは非常に不都合なことである．

それらを踏まえたうえで，本稿ではグループ化を用いた木構造表記型多重署名方式について提案する．この方式では，コンテンツの二次利用により新たに作成されたコンテンツが，オリジナルコンテンツを正しい内容で引用していること，コンテンツの構造が完成した後に，正当な編集者が署名を施すことによってコンテンツの構造を変更することが可能である．そしてマッシュアップされたコンテンツの再編集に対しては，一次利用，二次利用といった階層毎のコンテンツに対して，そのグループを規定することが有効である．本稿では，ハッシュ関数の操作からマッシュアップ過程のグループを規定できる識別子を生成し，再編集に有効である署名方式を実現した．また，この操作を加えたことで，新たな公開鍵を発行する手間を必要とせずにコンテンツを何度他のコンテンツと組みこんでも署名の正当性を検証できることが可能となった．

本稿の構成を以下に示す．第 2 章において従来の方式を説明し，その問題点を整理する．その後，第 3 章において提案方式を紹介し，第 4 章でまとめとする．

2．先行研究

稲村らによって提案された方式[1]では，BLS 署名[6]の多重化によって階層関係を規定している．その表現によって二次利用への著作権保護を展開している．また，白川らによって提案された方式では，コンテンツの再構成を想定した方式が提案されている．この章では，それらの方式を説明し，またそれらの問題点を示す．

2．1　稲村らによる方式

稲村らの方式の署名生成アルゴリズムを以下に示す．$u_{q,r(q)}$ を Aggregate 署名[7]作成に参加した署名者とし，$x_{l_i,q(l_i)}, v_{l_i,q(l_i)}$ をそのユーザの BLS 署名における署名鍵，検証鍵とする．各ユーザはマッシュアップコンテンツの構造を展開した木構造において，それぞれのノードに配置されるものとし，q はルートノードを階層 1 とした時の，その署名者が位置するノードの階層を示し $(1 \le q \le n), r(q)$ は階層 q に位置するその署名者の識別符号とする．この時，

$$L_{q,r(q)} = \{(u_{q,r(q)}, \{u_{q+1,r(q+1)}, \ldots\})$$

$$(u_{q+1,r(q+1)_1}, \{u_{q+2,r(q+2)_1}, \ldots\}), \ldots$$

$$, (u_{i-1,r(i-1)_\alpha}, \{u_{i,r(i)_\beta}, \ldots\}), \ldots\}$$

を部分的な木構造で表現される $u_{q,r(q)}$ 以下の署名者の関係性を示した付随情報とする．

但し，親ノードに位置する署名者が異なると，子ノードに位置する署名者の識別符号も異なるため，便宜上識別符号の箇所に添字を付与することでこの差異を表現している．さらに $m_{q,r(q)}$ を署名対象となる平文とし，$H : \{0,1\}^* \to G$ を一方向性ハッシュ関数と定義し，$h_{q,r(q)} = H(m_{q,r(q)})$ とする．

2．1．1　署名生成

以下の手順で，体系を保証する Aggregate 署名が生成される．

1．リーフノードに位置する署名者 $u_{l_{\alpha 1},r(l_{\alpha 1})}$ は，平文 $m_{l_{\alpha 1},r(l_{\alpha 1})}$ から $h_{l_{a1},r(l_{\alpha 1})} = H(m_{l_{\alpha 1},r(l_{\alpha 1})})$ を求める．そしてデジタル署名 $\sigma_{l_{\alpha 1},r(l_{a1})} \leftarrow x_{l_{a1},r(l_{a1})} H(m_{l_{a1},r(l_{a1})})$ を計算する．これらのデジタル署名を自分の親ノードに位置する署名者 $u_{l_{\alpha 1-1},r(l_{a1-1})\beta 1}$ に送信する．

2．中間ノードに位置する(リーフノードとルート以外の)署名者 $u_{s,r(s)_\beta}$ は，自分の子ノードに位置する署名者 $u_{l_{s+1},r(S+1)_1}, \ldots u_{l_{s+1},r(S+1)k'}$ から受信した全ての $m_{l_{\alpha 2},r(l_{\alpha 2})_i}$ (ただし $1 \le i \le k'$)を用いて $h_{s+1,r(s+1)_i} = H(m_{s+1,r(s+1)_i})$ を求める．さらに署名者 $u_{s,r(s)_\beta}$ は，自分が本来署名したいメッセージ $m_{s,r(s)_\beta}$ から $h_{s,r(s)_\beta} = H(m_{s,r(s)_\beta})$ を求める．これと全

ての子ノードに位置する署名者$u_{s+1,r(s+1)_i}$から受信した$\sigma_{s+1,r(s+1)_i}$を用いて

$\sigma_{s,r(s)_\beta}$

$=\sum_{i=1}^{k'}(\sigma_{s+1,r(s+1)_i}x_{s,r(s)_\beta}h_{s+1,r(s+1)_i}+x_{s,r(s)_\beta}h_{s+1,r(s+1)_i})$

を計算する．また，

$L_{s,r(s)_\beta}=\sum_{i=1}^{k'}L_{s+1,r(s+1)_i}+\{(u_{s,r(s)_\beta},\{u_{s+1,r(s+1)_1},\ldots u_{s+1,r(s+1)_{k'}}\})\}$

を作成する．

この$\sigma_{s,r(s)_\beta},L_{s,r(s)_\beta},m_{s,r(s)_\beta}$を自分の親ノードに位置する署名者$u_{s-1,r(s-1)_{\beta'}}$に送信する．この手順をルートの子ノードに位置する署名者まで再帰的に行う．

3．ルートに位置する署名者$u_{1.1}$は，自分の子ノードに位置する署名者$u_{2.1},\ldots u_{2.k}$から受信した全ての$m_{2,i}$(ただし$1\le i\le k$)を用いて，$h_{2,i}\leftarrow H(m_{2,i})$を求める．さらに署名者$u_{1.1}$は自分が本来署名したいメッセージ$m_{2,i}$から$h_{1,1}\leftarrow H(m_{1,1})$を求める．これと子ノードの署名者$u_{2.i.}$から受信した$\sigma_{2.i}$を用いて$\sigma_{1.1.}=\sum_{i=1}^{k'}(\sigma_{1.i}+x_{1.1}h_{1.i})+x_{1.1}h_{1.1}$を計算する．

また，$L_{1.1}=\sum_{i=1}^{k}L_{2.i}+\{(u_{1.1},\{u_{2.1}\ldots u_{2.k}\})\}$を作成する．この$\sigma_{1.1}$および$L_{1.1}$を全署名者$u_{q,r(q)}$の全署名対象$m_{q,r(q)}$に対する Aggregate 署名として公開する．

2．1．2　署名検証

以下の手順で Aggregate 署名の検証を行う．

1．検証者は$L_{1.1}$に示されているすべての署名者の検証鍵$v_{q,r(q)}$および署名対象となる全ての平文$m_{q,r(q)}$を集める．

2．検証者は，集めた平文から$h_{q,r(q)}=H(m_{q,r(q)})$を求める．

3．検証者は，

$\prod_{AllLeaf}e(v_{l_i,q(l_i)},h_{l_i,q(l_i)})$と

$\prod_{AllexceptLeaf}e(v_{l_i,q(l_i)_i},h_{q+1,r(q+1)_j}+h_{q,r(q)_i})$

の積を計算し，この値と$e(g,\sigma_{1,1})$の値が一致することを確認する．

2．2　白川らによる方式

白川らの方式では，稲村らが提案した方式と同様のシナリオに対する署名方式を展開すると共に，すでにマッシュアップされたコンテンツを一次のコンテンツ(稲村らのシナリオでは木構造のリーフに属すコンテンツ)として扱うことを可能としている．それぞれの方式を示す．方式の説明に際しては，すでに編集者に対しては BLS 署名での鍵生成と同じアルゴリズムによって生成された鍵ペアが発行されているとする．また白川らの方式においては，マッシュアップ階層の深さを 2 とし，署名方式を展開している．そして，一次コンテンツを編集してコンテンツを生成する編集者を上位者と呼び，一次コンテンツ作成者を下位者と規定している．白川らの手法では以下で定義されるハッシュ関数H_1,H_2を利用する．

$H_1:\{0,1\}^n\rightarrow\{0,1\}^n$

$H_2:\{0,1\}^n\rightarrow G$

2．2．1　一次コンテンツのみでマッシュアップされたコンテンツへの署名方式

以下の手順で署名生成を行う．

1．　下位者iは自身のコンテンツm_iから

$\sigma_i\leftarrow H_2(m)^{x_i}$によって署名を生成する．この署名$\sigma_i$上位者へと送信する．また$m_i$をコンテンツに対するメタデータとする．

2．　上位者kは自身のコンテンツm_kから

$h_k\leftarrow H_1(m_k)$を算出する．自身の秘密鍵x_kとh_kを用いて$\sigma_k=\prod_i(\sigma_i{}^{h_k})^{x_k}=\prod_i(h_i{}^{x_i})^{x_kh_k}$によって署名を算出する．この$\sigma_k$を上位者の署名とする．

ここで上位者kは引用した編集者の公開鍵に対して新たな公開鍵Vを生成する．

$V_k=\{V_1,V_2,V_2\}=\{v_1{}^{x_k},v_2{}^{x_k},v_2{}^{x_k}\}$を生成する．

署名検証は以下のようにして行う．

1. 全ての下位者の検証鍵v_i，上位者が作成したV_kについて上位者の検証鍵v_iを用いて以下の検証を行う．

$e(V_k,g)=\prod e(v_i,v_k)$

検証が成功すれば新たな公開鍵が正当であるとみなす.

2. 検証者は上位者，下位者全てのコンテンツを用いて $e(\sigma_k, g) = \prod_i (h_i^{h_k}, v_i^{x_k})$ を計算し，成功すれば署名の正当性を承諾する.

2. 2. 2　一次コンテンツと n 次コンテンツでの組み合わせに対する署名方式

上記における処理との違いは会社にマッシュアップされたコンテンツが組み込まれていることである. これを踏まえた上で，アルゴリズムの説明を示す.

以下の手順で署名生成を行う.

1.　下位者 i (一次コンテンツの作成者)は自身のコンテンツ m_i から，$\sigma_i \leftarrow H_2(m)^{x_i}$　によって署名を生成する. この署名 σ_i を上位者に送信する. また m_i をコンテンツに対するメタデータとする.

2.　下位者 n (n 次コンテンツの作成者)は新たな署名鍵 x_n を手に入れる. それ以前に所有している鍵ペア (x_n', v_n') から $v = v_n'^{x_n}$ を計算し，v を新たな公開鍵とする. そして，すでに作成された署名 σ' と x_n から $\sigma = \sigma'^{x_n}$ を計算し，σ_i を新たな署名として上位へ送信する. また，σ_i 構成する直下のコンテンツの署名を集め，$\delta = \prod_l \sigma_l$ を新たに計算する.

3. 上位者 k は一次コンテンツの組み合わせの時と同様にして署名を生成する.

署名検証は以下のようにして行う

1. 全ての下位者の検証鍵 v_i , 上位者が作成した V_k について，上位者の検証鍵 v_i を使って以下の検証を行う.

2. 検証者は上位者，下位者全てのコンテンツを用いる. 下位者に属する n 次コンテンツに対してはハッシュ値は $h_i \leftarrow \delta^{H_i(m_i)}$ として計算する. これらの値を用いて $e(\sigma_k, g) = \prod_i (h_i^{h_k}, v_i^{x_k})$ を計算し, 成功すれば署名の正当性を承諾する.

2. 2. 3　問題点

稲村らの方式では，署名の多重化を繰り返すことによって階層関係を規定している. しかし，マッシュアップの体系が完成した後に任意のコンテンツの権利を体系に組み込むことを可能とする署名方式のアルゴリズムの検討において，中間編集者は $\sigma_{s,r(s)_\beta}$

$= \sum_{i=1}^{k'} (\sigma_{s+1,r(s+1)_i} + x_{s,r(s)_{\beta i}} h_{s+1,r(s+1)_i} + x_{s,r(s)_{\beta i}} h_{s+1,r(s+1)_i})$

の計算を必要とする. また，体系が完成し，$\sigma_{1.1}$ が生成された段階において体系には存在しなかったコンテンツ $m_{s+1,r(s+1)_{malicious}}$ を中間編集者 $u_{s,r(s)_\beta}$ によって加えることが以下の式でわかる. ただし $u_{s+1,r(s+1)_{malicious}}$ が存在し，$m_{s+1,r(s+1)_{malicious}}$ に対する署名 $\sigma_{s+1,r(s+1)_{malicious}}$ は計算されているものとする.

$\sigma_{s,r(s)_{\beta-malicious}}$

$= \sigma_{s,r(s)_\beta} + \sigma_{s+1,r(s+1)_{malicious}} + x_{s,r(s)_\beta} h_{s+1,r(s+1)_{malicious}}$

この操作によって $\sigma_{1.1}$ が生成された後に新たな署名が作成される. よって，この方式では階層を隔てた関係性が弱い方式であるといえる.

次に，白川らの方式では新たな公開鍵がコンテンツ毎に必要であることが大きな負担となると考えられている. 公開鍵は一般的に PKI によって正当性を保証する必要がある. しかし白川らの方式では，マッシュアップコンテンツの階層が深くなるにつれて公開鍵のサイズが大きくなり，PKI の負荷は多大である. この問題点は白川らも示唆している.

3．提案方式

3. 1　二次利用コンテンツに対する著作権保護

2 章においてマッシュアップコンテンツによるコンテンツに対する著作権保護対象について言及した. この節において，コンテンツの構成集合の定義も行い，階層を隔てた関係性を強固に保障でき，各マッシュアップコンテンツに対して新たな公開鍵を必要としない署名方式による二次利用を考慮した著作権保護方

式の提案について説明する．

初めにその要件を以下に示す．

1．オリジナルコンテンツ作成者や編集者の関係性とそれらの著作権の主張・保証が可能であること．

2．署名作成・検証において，特定の箇所に負荷が集中しない．

3．マッシュアップのコンテンツ引用順，あるいは素材提供者や編集者といったグループ毎の区別が必要となる．

4．二次利用されたコンテンツの二次利用など，利用推移が異なるコンテンツを複数組み合わせて新たなコンテンツを作り上げることが想定される．

以上の要件から提案方式では要件 3．4 に対して，ハッシュ関数の操作によってグループ毎の識別子を生成して引用されているコンテンツをグループとして区別すること，利用推移が異なるコンテンツによる編集から生成されたコンテンツの検証負荷を軽減することの 2 つを実現した．

3．2　マッシュアップによるコンテンツの構成集合

コンテンツの一次利用・二次利用・・・といったマッシュアップコンテンツへの構成集合を定義する．(図 3．1)あるコンテンツが複数の著作物を引用して生成されている場合，そのコンテンツを引用されているコンテンツを 1 つの集合として定義する．またその概念によって，その生成されたコンテンツを引用して作られた二次利用コンテンツに対しても同様にして，それを 1 つの集合として定義することができる．したがって，図 3．1 によって示されるマッシュアップコンテンツに対しては，集合 1，集合 2，集合 3 の内容を引用したコンテンツ集合 4 としてさらに定義することが可能となる．

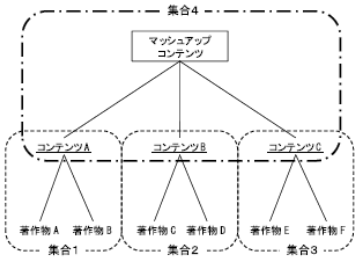

図 3．1 マッシュアップによるコンテンツの構成集合の定義

このようにして定義した時，オリジナルコンテンツ作成者(図中における著作物の作成者)や編集者の関係性の検証について整理する．初めに，3．1 節で述べた要件 1 で求められる機能を図 3．1 の場合において振り返る．要件 1 が示すことは，コンテンツ A，B，C それぞれが各著作物を正しい内容で引用して生成されていることがわかり，さらにマッシュアップコンテンツがコンテンツ A，B，C を正しい内容で引用して生成されていることが検証できるということである．つまり，マッシュアップされたコンテンツから生成された電子署名の検証によって，それぞれのコンテンツの著作権とその関係性を検証できることが必要とされる．(図 3．2)よってマッシュアップコンテンツの構成集合を定義してこの要件を満たすには，集合 1・集合 2・集合 3 のそれぞれの集合内で著作権とその関係性を検証し，さらに集合 4 におけるコンテンツを検証すればよい．

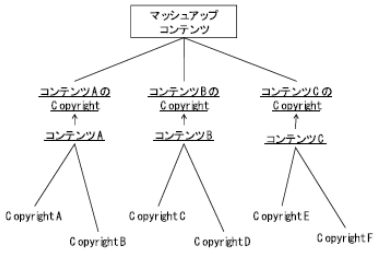

図 3．2 マッシュアップコンテンツの権利推移

3．3　二次利用コンテンツへの構成集合を識別可能とする署名方式

前節で定義したコンテンツの構成集合を利用し，コンテンツの二次利用を考慮した著作権保護方式を電

子署名法しいによって実現する．また，マッシュアップされるそれぞれのコンテンツ M を，著作権メタデータとコンテンツデータの 2 つを含めたコンテンツ全体のデータとして定義する．そして小出らの方式では，引用したコンテンツ M を入力とするハッシュ関数から出力された結果を，生成された二次利用コンテンツデータと関連づけて電子署名を生成する．方式の一般性を失うものではないが，簡単な説明のためにコンテンツの引用推移を完全 3 分木の 3 階層の木構造として表現する．(図 3．3)図 3．3 のようにコンテンツの利用推移を定義するとき，提案方式では 2．1．1 節での定義にしたがって 4 つの集合を構成することができる．その集合をそれぞれ $C_1, C_2, C_3 C_4$ と定義する．そして以下のように各集合の要素を定義する．

$\{M_{1-1}, M_{1-2}, M_{1-3}, M_{2-1}\} \in C_1$
$\{M_{1-4}, M_{1-5}, M_{1-6}, M_{2-2}\} \in C_2$
$\{M_{1-7}, M_{1-8}, M_{1-9}, M_{2-3}\} \in C_3$
$\{M_{2-1}, M_{2-2}, M_{2-3}, M_3\} \in C_4$

またそれぞれのコンテンツ作成者を $u_{i-j} (1 \le i \le 3, 1 \le j \le 9)$ とする．次に方式の前提となる条件を以下に示す．

1．編集・加工によって生成されていなコンテンツ $M_{1-j} (1 \le j \le 9)$ はそのことを証明する情報 α を共通に持つとする．また情報 α を持つコンテンツをオリジナルコンテンツとし，編集・加工によるコンテンツを二次利用コンテンツと定義する．さらに，最終的な編集によって生成されたコンテンツをマッシュアップコンテンツと呼ぶ．

2．コンテンツの作成者・編集者は第三者によって正当であると証明されている秘密鍵・公開鍵の鍵ペアを持つとする．

3．コンテンツ作成時におけるすべてのオリジナルコンテンツ作成者・編集者は正当に署名作成を行う．

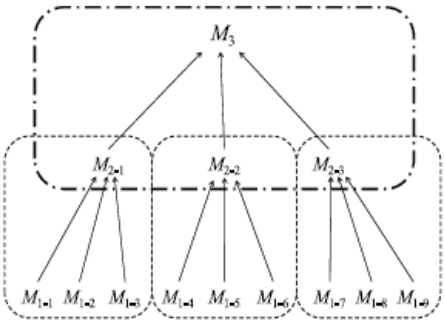

図 3．3 マッシュアップコンテンツの構成要素

提案方式では， BLS 署名と Aggregate 署名によって，編集により生成されたコンテンツの著作権とその引用コンテンツとの関係性を検証する．また，署名生成では引用コンテンツとの関係性の証明を行うために，グループ毎の識別子を生成する処理を加えた．方式の説明では図 3．3 におけるマッシュアップによるコンテンツへの署名生成過程を示す．なお，グループ毎の識別子の生成には以下のハッシュ関数 H と変換関数 f を使用する．

$H_1 : \{0,1\}^n \to \{0,1\}^n$
$H_2 : \{0,1\}^n \times \{0,1\}^n \to Z / pZ$
$H_3 : \{0,1\}^n \to G_1$
$f : Z / pZ \to \{0,1\}^n$

3．4　署名生成

方式の BLS 署名方式と同様にして行う．図 3．3 におけるそれぞれのコンテンツ作成者(署名者) $u_{i-j} (1 \le i \le 3, 1 \le j \le 9)$ について $x_{u_{i-j}} \in Z_p$ を選択し，$v_{u_{i-j}} \leftarrow g_2^{x_{u_{i-j}}}$ を計算する． $x_{u_{i-j}}$ を秘密鍵， $v_{u_{i-j}}$ をその公開鍵とする．

1．　オリジナルコンテンツ作成者による署名生成

初めにオリジナルコンテンツに対しての署名方法を以下に示す． u_{1-j} は自身の秘密鍵でコンテンツに対して署名を施す． $\sigma_{1-j} \leftarrow H_3(M_{i-j})^{x_{u_{1-j}}} (1 \le j \le 9)$

そして σ_{1-j} と M_{1-j} をオリジナルコンテンツ作成者 u_{1-j} は出力する．

2．二次利用コンテンツへの署名生成

集合C_1において以下の処理を行う．まず，各著作物からハッシュ値を生成する．

$h_{i-j} \leftarrow H_1(M_{1-j})(1 \le j \le 3)$

それらのハッシュ値に対して，

$h \leftarrow h_{1-1} \oplus h_{1-2} \oplus h_{1-3}$

を計算する．次にそのハッシュ値$h \in \{0,1\}^n$と引用元が持つコンテンツ$\alpha \in \{0,1\}^n$から集合識別子$\lambda \in Z / pZ$を生成する．

$\lambda_{2-1} \leftarrow H_2(\alpha, h)$

生成されたλ_{2-1}と秘密鍵を用いて，編集者u_{2-1}による二次利用コンテンツへの署名を生成する．

$\sigma_{2-1} \leftarrow H_3(M_{2-1})^{\lambda_{2-1} x_{u2-1}}$

また，編集で引用したコンテンツの作成者情報U_{2-1}と集合情報C_1を出力する．

$U_{2-1} =< u_{1-1}, u_{1-2}, u_{1-3}, u_{2-1} >$
$C_1 =< M_{1-1}, M_{1-2}, M_{1-3}, M_{2-1} >$

編集者u_{2-1}はコンテンツM_{2-1}と

$I_{2-1} =< \sigma_{2-1}, \lambda_{2-1}, U_{2-1}, C_1 >$を公開する．

集合C_2, C_3においても同様にして署名処理を行う．

3. 最終編集者による署名生成

$I_{2-1}, I_{2-2}, I_{2-3}$とそれに伴うコンテンツ$M_{2-j}(1 \le j \le 3)$を集める．次に，各コンテンツからハッシュ値を生成する．

$h_{2-j} \leftarrow H_1(M_{2-j})(1 \le j \le 3)$

生成されたハッシュ値$h_{2-1}, h_{2-2}, h_{2-3}$に対して

$h \leftarrow h_{2-1} \oplus h_{2-2} \oplus h_{2-3}$

を計算する．また，二次利用コンテンツを引用していることを示す情報として，

$\beta \leftarrow f(\lambda_{2-1}) \oplus f(\lambda_{2-2}) \oplus f(\lambda_{2-3})$

を計算する．算出された$\beta \in \{0,1\}^n$とハッシュ値$h \in \{0,1\}^n$から集合識別子$\lambda_3 \in Z / pZ$を導出する．

$\lambda_3 \leftarrow H_2(\beta, h)$

生成されたλ_3と秘密鍵を用いて，編集者u_3によって編集されたコンテンツへの署名を生成する．

$\sigma \leftarrow H_3(M_3)^{\lambda_3 x_{u3}}$

また，

$U_3 =< u_{2-1}, u_{2-2}, u_{2-3}, u_3 >$
$C_4 =< M_{2-1}, M_{2-2}, M_{2-3}, M_3 >$

を出力する．さらに最終編集者u_3はそれぞれの署名を集約する．

$\sigma \leftarrow \sigma_3 \times \prod_{i=1}^{3} \sigma_{2-i} \times \prod_{i=1}^{9} \sigma_{1-i}$

最後に，最終編集者u_3はコンテンツM_3と

$I_3 =< \sigma, \sigma_3, \lambda_3, U_3, C_4 >$を出力する．

3. 5　検証処理

検証者は最終編集者からの情報U_3から$U_{2-1}, U_{2-2}, U_{2-3}$へアクセスし，コンテンツの利用推移を復元する．次に，オリジナルコンテンツの署名を検証する．すなわち，それぞれのコンテンツに対して，$e(\sigma_{1-i}, g_2) = e(H_3(M_{1-i}), v)$が成り立つかを検証する．それらの署名検証が成功したことを条件に以下の処理を行う．

1.　オリジナルコンテンツの署名を検証者が集約し，集約署名σ_{veri}を算出する．

$\sigma_{veri} \leftarrow \prod_{i=1}^{9} \sigma_{1-i}$

2.　署名生成の過程で公開されている情報Uを基にして，各集合内でそれぞれのλを計算する．

3.　$\sigma^* \leftarrow \sigma / \sigma_{ver}$によって，最終編集者によって生成された$\sigma$からオリジナルコンテンツの署名集合を削除する．そして検証式

$e(\sigma^*, g_2) = e(H_3(M_3)^{\lambda_3}, v_{u_3}) \prod_{i=1}^{3} e(H_3(M_{2-i})^{\lambda_{2-i}}, v_{u_{2-i}})$

が成り立つか検証する．

3. 6　考察

2. 2. 1節と2. 2. 2節で説明した署名方式によって以下のことが実現される．

1.　コンテンツの二次利用により新たに作成されたコンテンツがオリジナルコンテンツを正しい内容で引用していること．

2.　引用されているコンテンツをグループとして区別すること．

提案方式では，コンテンツの利用推移の各段階におい

てグループ識別子 λ を生成することによりマッシュアップコンテンツのグループの識別を実現している．そして各編集者自身がその識別子 λ を用いて，生成した二次利用コンテンツに対して署名を施すとした．識別子 λ により二次利用コンテンツの構成集合を識別でき，マッシュアップされたコンテンツの署名生成においてもその識別子とコンテンツを関連付けた署名生成を実現している．さらに最終編集者がマッシュアップに参加した編集者の署名を集約することで，その署名検証を通じてマッシュアップコンテンツの権利の関係性を確認できる．検証処理においては，オリジナルコンテンツの署名検証を最初に行い，検証者自身で識別子 λ を算出し，その後に二次利用コンテンツの署名検証処理を行うとした．そしてこの手順によって，署名者の関係性を確認することにした．

文献[3]での方式では，二次利用コンテンツの生成に伴い，新たな公開鍵の発行が必要となっていた．しかし，提案方式ではその手間を必要とせずに同等の機能を実現している．提案方式における二次利用コンテンツの再編集に関して言及する．提案方式では新たに組み合わされるコンテンツの識別子をマッシュアップ過程に組み込み直すことだけを必要とする．よって新たな公開鍵の発行を必要とせずに，提案方式では二次利用コンテンツの再編集を行える．

４．まとめ

二次利用コンテンツの構成集合を定義し，その集合に対して識別子を生成し，二次利用コンテンツへの著作権保護方式を提案した．提案方式では，生成された識別子を BLS 署名方式による署名生成に組み込み，二次利用コンテンツが正しい内容で引用されていることとその引用順序を規定できる方式を実現した．さらにコンテンツの構成要素をグループとして定義することで，マッシュアップコンテンツの再構成へと対応した．

今後は安全性の検討とシミュレーション実験によって従来方式との処理速度の簡単からの比較が必要と考える．

５．参考文献

[1] youtube，http://www.youtube.com

[2] パソコン決裁，
http://www.shachihata.co.jp/interweb/
index.php

[3] 稲村勝樹，渡辺龍，田中俊昭，“コンテンツの二次利用を実現する著作権保証方式” 暗号と情報セキュリティシンポジウム-SCIS2009，1B2-4，2009

[4] 白川瑞樹，岩村惠一，“ボトムアップ型デジタルコンテンツ編集システム” Vol2010-CSEC-49，No7，2010

[5] 稲村勝樹，渡辺龍，田中俊昭，“Gap Diffie-Hellman 署名に基づいた順序付きアグリゲート署名とその拡張方式現” 暗号とセキュリティシンポジウム-SCIS2010，1A-2-2，2010

[6] D. Boneh, B, Lynn, and, H. Shacham, “Short signature from the Weil pairing,” Asiacrypt 2001, LNCS 2248, pp. 514-532, Springer-Verlag, Berlin, 2001.

[7] D. Boneh, C. Gentry, B, Lynn, and, H. Shacham “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” EUROCRYPT 2003, LNCS2656，pp．416-432，2003.

[8] 梶克彦，長尾確：部分引用の管理に基づく web コンテンツのマッシュアップ，情報処理学会第 69 回全国大会，5D-1(2007).

[9] 齊藤泰一：順序指定可能な多重署名，暗号と情報セキュリティシンポジウム-SCIS’97，33A(1997).