コンテンツの再編集を許可する木構造表記型多重署名方式

伊佐仁 † a)　小出雅史 †　稲村勝樹 †　岩村惠市 †

† 東京理科大学大学院工学研究科

102-0073 東京都千代田区九段北 1-14-6
a)E-mail : isa@sec.ee.kagu.tus.ac.jp

あらまし　消費者生成メディア(CGM:Consumer Generated Media)において，近年ではデジタル署名を用いて著作者の権利関係を保証するといった研究が行われている．それらの研究では，引用されたコンテンツが正しい内容で引用されていること，また引用順序を規定できるという方式が提案されている．しかし，これらの方式では引用元のコンテンツが変更・削除などされた場合，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欠点がある．そこで本稿では，グループ化という処理を加えることによって既存方式の良さを持ったまま，上記の欠点を解決する新方式を提案する．また，提案方式に対して安全性の評価を行うことによって，方式が数学的に安全であることを示した．

A Tree-structured-specified Multisignature Scheme Permitted Re-editing Content Data.

Hitoshi Isa † a)　Koide Masashi †　Masaki Inamura †　Keiichi Iwamura †

† Graduate school of Engineering, Tokyo University of Science.

1-14-6 Kudankita, Chiyoda-ku, Tokyo 102-0073 Japan
a)E-mail : isa@sec.ee.kagu.tus.ac.jp

Acstract Recently, somesystems to protect the copyright of content data for CGM with a digital signature scheme is researched. The systems, with which the content makers/editors can establish the rightness and the quoting order of the “Mash-up” contents, have been proposed. However, in the systems, they need to generate the renewal signature from the first signer to the last if the original content data has been altered, deleted or inserted in editing, and so the signature processing takes extra time and throughput.In this paper, we propose the new system to solve such defects by adding a process so called "grouping". Furthermore we explain that our proposal is provable secure.

■　1．はじめに

近年，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，二次利用・三次利用・・・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．

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ことを目的としていた．そのため従来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いた．しかし，CGMサービスの普及により，コンテンツの二次利用を制限するというものではなく，CGMコンテンツに対するマッシュアップを考慮したシステム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システムが検討されている[3][8]．また，それを拡張し，複数の署名者によ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]がなされおり，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という欠点がある．

それらを踏まえたうえで，本稿ではグループ化を用いた新たな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．

本稿では 2 章において BLS 署名[6][7]について説明する．3 章において提案方式について説明する．そして 4 章で提案方式の安全性を評価し，5 章でまとめとする．

■ 2. BLS 署名

BLS 署名のアルゴリズムを以下に示す．

また，双線形写像の詳細については省略する．

1. 鍵生成：$x \in Z_p$ を選択し，$v \leftarrow g_2^{\,x}$ を計算する．x を署名に使用する秘密鍵とし，v をその公開鍵とする．
2. 署名：一方向性ハッシュ関数 $H:\{0,1\}^* \rightarrow G_1$ を定義する．m を署名対象となる平文として，$\sigma \leftarrow H(m)^x$ を計算する．そして σ を m に対するデジタル署名とする．
3. 検証：検証者に公開鍵 v，平文 m と署名 σ が与えられているとして $e(\sigma, g_2) = e(H(m), v)$ であるかを検証する．

双線形写像の特性によって

$e(\sigma, g_2) = e(H(m)^x, g_2) = e(H(m), v)$ と展開され，平文の正当性を署名 σ によって検証できる．

■ 3. 提案方式

この章では，本稿の提案方式について説明する．そして，ここでは以下の図 1 を例にとる．

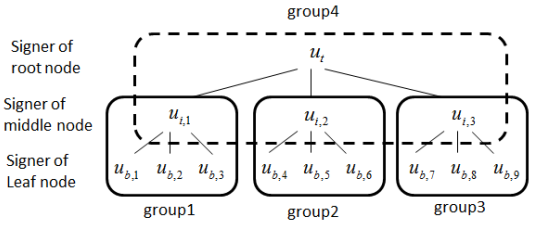

図 1　グループ化を用いた木構造表記型多重署名

Figure 1 Tree-structured-specified multisignature scheme with grouping

なお，図1はあくまでも例であって，方式の一般性を損なうものではない.

■ 3. 1 記号，前提条件，要求条件

提案方式において使用される記号と前提条件，要求条件を以下に示す.

記号：

G :ペアリングの演算が可能な楕円曲線上の点の集合.

g :G の要素である生成元.

e :ペアリング関数.

u_t :ルートノード署名者(1 人).

u_{i,o_i} :中間ノード署名者.

u_{b,o_b} :リーフノード署名者.

x_t, v_t :ルートノード署名者の署名鍵，および検証鍵.

x_{i,o_i}, v_{i,o_i} :中間ノード署名者の署名鍵，および検証鍵.

x_{b,o_i}, v_{b,o_i} :リーフノード署名者の署名鍵，および検証鍵.

L_t :リーフノード署名者からルートノード署名者までの全署名者の位置情報.

L_{i,o_i} :リーフノード署名者から中間ノード署名者 u_{i,o_i} までの署名者の位置情報.

L_{b,o_b} :リーフノード署名者 u_{b,o_b} の位置情報

V_t :多重署名を検証する最終中間鍵.

V_{i,o_i}, V_{b,o_b} :V_t を求めるためのサブ中間鍵.

M_t :署名対象となるルートノードでのコンテンツ.

M_{i,o_i} :署名対象となる中間ノードでのコンテンツ.

M_{b,o_i} :署名対象となるリーフノードでのコンテンツ.

$\sigma_{i,o_i}, \sigma_{b,o_b}$:BLS 署名による u_{i,o_i} ，あるいは u_{b,o_b} の m への署名.

S_t :全署名者の署名を集約した木構造表記型多重署名.

S_{i,o_i}, S_{b,o_b} : S_t を求めるための中間ノード署名者 u_{i,o_i} ，あるいはリーフノード署名者 u_{b,o_b} までのサブ木構造表記型多重署名.

$\lambda_{i,j}$:リーフノードと中間ノード間で生成される集合識別子.

α :編集・加工によって作成されていないコンテンツを証明する情報. α をもつコンテンツをオリジナルコンテンツとする.

λ_t :中間ノードとルートノード間で生成される集合識別子.

$\|$:コンテンツなどを連接するための記号.

また，以下のハッシュ関数、変換関数 f も定義する.

$$H_1 : \{0,1\}^n \to \{0,1\}^l (n \geq l)$$
$$H_2 : \{0,1\}^n \times \{0,1\}^n \to Z / pZ$$
$$H_3 : \{0,1\}^n \to G$$
$$f : Z / pZ \to \{0,1\}^n$$

前提条件：

1. 公開鍵基盤は整備されており，認証局により全ての署名者の署名鍵, および検証鍵のペアが正当に発行されている.

2. 署名者に発行されている鍵ペア以外に，新たな鍵ペアの発行は行わない．
3. 多重署名作成中において，署名者間の通信は安全に行われ，作成中の中間情報を第三者が入手することは不可能である．

要求条件：

CGM コンテンツの特性を考えて，以下のことについて保護をしなければならないと考える．

1. [多重署名正当性] 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. [署名順序正当性] コンテンツの構造内における署名者間の署名順序が正しく示されていること．
3. [偽装参加不能性] 第三者が不正にコンテンツの構造に署名者を追加することが困難なこと．
4. [集合識別子偽装不能性] 第三者が完成したコンテンツの構造に対し，不正に署名者の差し替えや削除を行うことが困難なこと．

■ 3．2 プロトコル

■ 3．2．1 鍵生成

鍵生成手順を以下に示す．

1. $g \in G$ を生成元とする．署名者 u_i は $x_i \in Z_p^*$ を選び(全ての署名者の署名鍵は各々異なるものとする)，$v_i = x_i g$ を計算する．

■ 3．2．2 署名生成

署名生成手順を以下に示す．

1. リーフノード署名者 $u_{b,1}$ は

$$h = H_3(M_{b,1})$$
$$\sigma_{b,1} = x_{b,1} h$$
$$S_{b,1} = \sigma_{b,1}$$
$$L_{b,1} = \{(0, u_{b,1})\}$$
$$V_{b,1} = v_{b,1}$$

を計算し，$(u_{b,1}, v_{b,1}, \sigma_{b,1}, S_{b,1}, L_{b,1}, V_{b,1})$ を署名者 $u_{b,1}$ の署名セットとして，上位の中間ノード署名者 $u_{i,1}$ に送信する．他のリーフノード署名者も同様の手順を行う．

2. 中間ノード署名者 $u_{i,1}$ は $h = H_3(M_{i,1})$ を計算する．また，中間ノード署名者は，各リーフノード署名者のコンテンツを連接し，ハッシュ値を生成する．

$$h \leftarrow H_1(M_{b,1} \| M_{b,2} \| M_{b,3})$$

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情報 $\alpha \in \{0,1\}^n$ から集合識別子 $\lambda \in Z / pZ$ を生成する．

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

$\lambda_{i,1}$ と秘密鍵を用いて，編集者 $u_{i,1}$ による二次利用コンテンツへの署名を生成する．

$$\sigma_{i,1} \leftarrow H_3(M_{i,1})^{x_{i,1}\lambda_{i,1}}$$

また，以下の値を計算する．

$$S_{i,1} = S_{b,1} + S_{b,2} + S_{b,3}$$
$$+ (x_{i,1} - 1)\sigma_{b,1} + (x_{i,1} - 1)\sigma_{b,2}$$
$$+ (x_{i,1} - 1)\sigma_{b,3} + \sigma_{i,1}$$

$$L_{i,1} = L_{b,1} + L_{b,2} + L_{b,3}$$
$$+ \{(u_{b,1}, u_{i,1}) + (u_{b,2}, u_{i,1}) + (u_{b,3}, u_{i,1})\}$$

$$V_{i,1} = V_{b,1} + V_{b,2} + V_{b,3}$$
$$+ (x_{i,1} - 1)v_{b,1} + (x_{i,1} - 1)v_{b,2}$$
$$+ (x_{i,1} - 1)v_{b,3} + v_{i,1}$$

そして $(u_{i,1}, v_{i,1}, \sigma_{i,1}, \lambda_{i,1}, S_{i,1}, L_{i,1}, V_{i,1})$ を署名者 $u_{i,1}$ の署名セットとして，上位のルートノ

ード署名者u_tに送信する．また，集合2・集合3においても同様の署名処理を行う．

3. ルートノード署名者u_tは$h = H_3(M_t)$を計算する．次に各コンテンツからハッシュ値を生成する．

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

そしてルートノード署名者は，各中間ノードのコンテンツを連接し，ハッシュ値を生成する．

$h = H_1(M_{i,1} \| M_{i,2} \| M_{i,3}))$

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，

$\beta = H_1(f(\lambda_{2-1}) \| f(\lambda_{2-2}) \| f(\lambda_{2-3}))$

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

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

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

$$\sigma_t \leftarrow H_3(M_t)^{x_t \lambda_t}$$

さらに，最終編集者u_tはそれぞれの署名を集約する．

$$\sigma \leftarrow \sigma_t \times \prod_{j=1}^{3} \sigma_{i,j} \times \prod_{j=1}^{9} \sigma_{b,j}$$

さらに以下の計算を行う．

$$S_t = S_{i,1} + S_{i,2} + S_{i,3}$$
$$+ (x_t - 1)\sigma_{i,1} + (x_t - 1)\sigma_{i,2} + (x_t - 1)\sigma_{i,3} + \sigma_t$$
$$L_t = L_{i,1} + L_{i,2} + L_{i,3}$$
$$+ \{(u_{i,1}, u_t) + (u_{i,2}, u_t) + (u_{i,3}, u_t)\}$$
$$V_t = V_{i,1} + V_{i,2} + V_{i,3}$$
$$+ (x_t - 1)v_{i,1} + (x_t - 1)v_{i,2} + (x_t - 1)v_{i,3} + v_t$$

最後に最終編集者u_tは$(\sigma, \sigma_3, \lambda_t, S_t, L_t, V_t)$を出力する．

■ 3．2．3 署名検証

署名検証手順を以下に示す．

検証者はそれぞれのコンテンツに対して，

$$e(\sigma_{b.j}, g_2) = e(H_3(M_{b,j}), v)$$

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，集約署名を算出する．

   $$\sigma_{veri} \leftarrow \prod_{j=1}^{9} \sigma_{b,j}$$

2. 各集合内でそれぞれの識別子λを計算する．

3. $\sigma^* \leftarrow \sigma / \sigma_{veri}$によって，最終編集者によって生成された$\sigma$からの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

   $$e(\sigma^*, g)$$
   $$= e(H_3(M_t)^{\lambda_t}, v_t) \prod_{j=1}^{3} e(H_3(M_{i,j})^{\lambda_{i,j}}, v_{i,j})$$

   が成り立つかを検証する．

■ 3．2．4 コンテンツ構造が再編集された場合の処理

$u_{b,1}$のコンテンツ$M_{b,1}$が変更され，$M'_{b,1}$となった場合を考える．リーフノードでの署名者$u_{b,1}$は新たな署名を生成する．

$h' = H_3(M'_{b,1})$

$\sigma_{b,1}' = x_{b,1} h'$

この値を用いて，3．2．2の手順1と同様に署名セッ生成し，トを中間ノードに送信する．中間ノード署名者は再び各リーフノード署名者のコンテンツを連接し，ハッシュ値を生成する．

$h' \leftarrow H_1(M'_{b,1} \| M_{b,2} \| M_{b,3})$

そして新たに集合識別子を生成する．

$\lambda_{i,1}' \leftarrow H_2(\alpha, h')$

また新たに編集者$u_{i,1}$による二次利用コンテンツへの署名を生成する.

$$\sigma_{i,1}' \leftarrow H_3(M_{i,1})^{x_{i,1}\lambda'_{i,1}}$$

残りの$S_{i,1}, L_{i,1}, V_{i,1}$は3. 2. 2の手順2と同様に計算を行う.

コンテンツの削除・追加に関しても同様の手順を行う.

上記においては, 集合識別子の再計算のみを考慮すればよく, 提案方式の利便さがわかる.

■ 4. 安全性評価

この章では, 3.1 節で示した要求条件を考慮し, 安全性の評価を行う.

■ 4. 1 多重署名正当性

ここで言う多重署名とは, 最終編集者u_tが集約した署名であるσのことである. これへの偽造が困難であれば多重署名の正当性が保証される. 以下の定理によって証明する.

定理 1 : ランダムオラクルモデルにおいて, σの偽造困難性と BLS 署名の偽造困難性は等価である.

証明 1 : A を σを偽造しようとする攻撃者とし, B を BLS 署名を偽造しようとする攻撃者とする.

B の攻撃が成功するなら A の攻撃も成功するということは自明であるため, A の攻撃が成功するなら B の攻撃も成功することを示すことで, 両者の攻撃が等価であることを示す. まず, 攻撃者 B は検証鍵$v_{b,1}$を保持しており, ランダムオラクルおよび署名オラクルへの応答を行う. この$v_{b,1}$に対し, $x_{b,1}g$と表すことができるが, B には$x_{b,1}$の値は未知とする. B は A を Honest Player として実行する. ここでそして B は A に$v_{b,1}$を渡し, A はランダムオラクル, および署名オラクルへの応答を行う. この操作によって A は他の鍵および識別子λ, コンテンツMを得ることが可能である.

これを踏まえて以下の計算を行う. 最終署名者u_3が集約したσは

$$\sigma \leftarrow \sigma_t \times \prod_{j=1}^{3} \sigma_{i,j} \times \prod_{j=1}^{9} \sigma_{b,j}$$

でありこれを展開すると,

$$\sigma \leftarrow H_3(M_t)^{x_t\lambda_t} \times H_3(M_{i.1})^{x_{i,1}\lambda_{i,1}} \times H_3(M_{i,2})^{x_{i,2}\lambda_{i,2}} \times H_3(M_{i,3})^{x_{i,3}\lambda_{i,3}} \times H_3(M_{b,1})^{x_{b,1}} \times H_3(M_{b,2})^{x_{b,2}} \times \cdots \times H_3(M_{b,8})^{x_{b,8}} \times H_3(M_{b,9})^{x_{b,9}}$$

となる. ここで$x_{b,1}$以外の署名鍵, 識別子, メッセージは既知のため, 以下のように計算すると,

$$\sigma \times \{\sigma_t \times \sigma_{i,1} \times \sigma_{i,2} \times \sigma_{i,3} \times \sigma_{b,2} \times \cdots \times \sigma_{b,8} \times \sigma_{b,9}\}^{-1} = H_3(M_{b,1})^{x_{b,1}}$$

となる. これは BLS 署名の式と同型となっているため, $x_{b,1}$を知らずして署名が得られたということになる.

■ 4. 2 署名順序正当性

2 人以上の署名者がいたとするとき, 署名者はそれぞれの署名鍵x_1, x_2からx_1x_2gなどの演算を行い, 中間鍵V_tを生成する. 上記における署名鍵の席の値であるx_1x_2が署名者の順序が隣接していることを示す. よって最終的な中間鍵であるV_tの偽造が困難であることが示されれば, 署名順序の正当性が保証される. V_tの安全性は以下の定理によって証明される.

定理 2: ランダムオラクルモデルにおいて, V_tの偽造困難性と CDH 問題[10]の困難性は等価である.

証明 2: A をV_tを偽造しようとする攻撃者とし,

B を CDH 問題を解く攻撃者とする．4．1 と同様に A の攻撃が成功するなら B の攻撃が成功することを示すことで，両者の攻撃が等価であることを示す．

リーフノード署名者が 1 人の場合は V_t は生成されない．リーフノード署名者が 2 人，中間ノード署名者が 1 人の場合，B は $v_{b,1}, v_{b,2}, v_{i,1}$ の検証鍵を保持しており，ランダムオラクルへの応答を行う．この検証鍵に対し，$x_{b,1}g, x_{b,2}g, x_{i,1}g$ と表すことができるが，B には $x_{b,1}, x_{b,2}, x_{i,1}$ の値は未知とする．B は A を Honest Player として実行する．まず B は A に $v_{b,1}, v_{b,2}, v_{i,1}$ を与えて，A は $V_t = x_{i,1}(v_{b,1} + v_{b,2}) + v_{i,1}$

を出力する．B はこの出力値と保持している $v_{i,1}$ から $x_{b,1}x_{i,1}g + x_{b,2}x_{i,1}g = V_t - v_{i,1}$

を計算することにより，$x_{b,1}g, x_{b,2}g, x_{i,1}g$ から $x_{b,1}x_{i,1}g, x_{b,2}x_{i,1}g$ が得られ，B の攻撃が成功する．さらにリーフノードが 3 人以上，中間ノードが 2 人以上のどの組み合わせでも同じことが言えるため，定理 2 が成立する．

■ 4．3 偽装参加不能性

偽装参加とは，ある署名者が署名体系に参加していない署名者を不正に体系に加える攻撃のことを言う．方式において，S_t とは全署名者の署名を集約した木構造表記型多重署名である．よって S_t の偽造が困難であることが示されれば，偽装参加不能性の正当性が保証される．S_t の安全性は以下の定理によって証明される．

定理 3：ランダムオラクルモデルにおいて，S_t の偽造困難性と離散対数問題の困難性は等価である．

証明 3：A を S_t を偽造しようとする攻撃者とし，B を離散対数問題を解く攻撃者とする．4．1 と同様 A の攻撃が成功するなら B の攻撃が成功することを示すことで，両者の攻撃が等価であることを示す．

署名者を不正に 1 人追加しようとする場合：B は検証鍵 $v_{b,d}$ を保持しているとする．この検証鍵に対し，$x_{b,d}g$ と表すことができるが，B には $x_{b,d}$ の値は未知であるものとする．B は A を Honest Player として実行する．まず，B は A に $v_{b,d}$ を与え，A は以下の値を手に入れる．

$(x_{b,j}, v_{b,j}), (x_{i,j}, v_{i,j})$ ここで j の値は任意とし，各ノードに何人の署名者がいてもいいことを意味する．ここでは簡易のためリーフノードが 2 人，中間ノードが 1 人の場合を考える．

A は手に入れた値をランダムオラクルへ応答する．そして以下の署名と S_t を得る．

$\sigma_{b,1}, \sigma_{b,2}, \sigma_{i,1}$

$S_t = x_{i,1}(x_{b,1} + x_{b,2} + x_{b,d})h_{i,1} + x_{i,1}h_{i,1}$

B は $\sigma_{b,1}, \sigma_{b,2}, \sigma_{i,1}$ を用いて以下の計算をする．

$(S_t - \sigma_{i,1}) \times (\sigma_{i,1})^{-1} - x_{b,1} - x_{b,2}$

$= x_{b,d}$

となり，$v_{b,d} = x_{b,d}g$ から $x_{b,d}$ が求まる．追加する人数が 2 人以上の場合でも同様の証明が可能である．またルートノードの計算過程においても同様のことが言える．

■ 4．4 集合識別子偽装不能性

ここではコンテンツの構造が完成した後に，攻撃者が不正にコンテンツの構造を操作しようとす

る場合について考察する．操作として，コンテンツの構造への不正な追加・削除・変更が考えられるが，変更については追加・削除を証明でき，それらを組み合わせれば同様の証明がつくため省略する．

1. コンテンツの構造が完成した後に，不正に署名者を追加する攻撃

中間ノードでの計算方法を例に考える．2 人以上追加した後に集合 1 において算出されるハッシュ値は以下のようになる．

$h' \leftarrow H_1(M_{b,1} \| M_{b,2} \| M_{b,3} \| M_{b,4} \| M_{b,5} \cdots)$

コンテンツ情報αは公開されているため，$h = h'$となれば攻撃者は$\lambda_{i,1}$を生成でき，攻撃が成功する．しかし，$h = h'$となるような都合の良いコンテンツを見つけることは難しく，それはハッシュ値の衝突問題に帰着するということがわかる．

2. コンテンツの構造が完成した後に，署名者を不正に削除する攻撃

中間ノードでの計算方法を例に考える．署名者を削除した後に算出されるハッシュ値は以下のようになる．ここでは$u_{b,1}$を削除した．

$h' \leftarrow H_1(M_{b,2} \| M_{b,3})$

$h = h'$となれば攻撃が成功といえる．しかしこれも先ほどの例と同様に，ハッシュ値の衝突問題に帰着できることが容易にわかる．またルートノードの計算過程においても，同様のことが言える．

■ 5. まとめ

CGM サービスにおいて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，また引用順序を規定できる方式を提案した．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，従来の方式に比べ，コンテンツの構造が完成した後に，構造の再構成が容易であるという方式であった．そしてこの方式に対して安全性の評価を行い，提案方式が数学的に安全であるということを示した．

今回提案した方式は，CGM コンテンツに特化した方式である．よって今後は，より一般的なコンテンツに対しての研究を行う予定である．また提案方式に対しては，効率化によるパフォーマンスの向上，シミュレーション実装による実用性の評価を行う予定である．

■ 6. 参考文献

[1] You Tube, http://www.youtube.com

[2]CreativeCommons, http://creativecommons.org/

[3] 小出雅史，岩村惠市，“2 次利用コンテンツの構成集合を識別可能とする署名方式”，暗号と情報セキュリティシンポジウム 2011，3E3-3，2011.

[4] 稲村勝樹，渡辺龍，田中俊昭，“回覧文書閲覧確認に適した階層表記型多重署名方式の提案と実装評価”電気情報通信学会論文誌，Vol.J93-B, No.10，2010.

[5] 稲村勝樹，岩村惠市，“新しい階層表記型アグリゲート署名を用いたコンテンツ引用過程表記手法”，情報処理学会論文誌，Vol.533，No.9, Sep.2012.

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

[7] D.Boneh，C.Gentry，B.Lynn，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.

[10] A. Boldyreva: Threshold signatures,Multis-ignaturesand Blind Signatures Based on the Gap Diffie-Hellman-Group Signature Scheme,Public Key Cryptography-PKC 2003, LNCS 2567,pp.31-46,Springer-Verlag , 2003.