Vol.2010-CSEC-49 No.7
2010/5/11

ボトムアップ型デジタルコンテンツ編集システム

白川瑞樹† 岩村恵市††

GDH 署名と Aggregate 署名を組み合わせ拡張することで，コンテンツの合成における n 次利用に関して，全てのコンテンツ作成者の著作権を保証する署名方式を提案する．また本稿では，一度作り上げたコンテンツを新たな他のコンテンツに組み合わせる署名方式の構想に基づき，コンテンツを何度他のコンテンツと組み込んでも署名の正当性を検証できることを示す．

キーワード：コンテンツ編集，著作権保護，Gap Diffie-Hellman グループ

A system of the bottom up type to edit digital content

Mizuki Shirakawa† and Keiichi Iwamura††

We propose a signature scheme which guarantees the copyright of the creator of all contents about the n-th use of contents, in adapting GDH signature and Aggregate signature. In this paper, if it makes contents by combining contents many times, we show that we can verify the signature based on the design of our signature scheme.

1. はじめに

近年のインターネット接続の一般化に伴い，多様化したネットワークサービスが大きな広がりを見せている．その中で，インターネットを活用し一般のユーザーが内容を生成していくメディア(CGM:Consumer Generated Medeia)が発生している．CGM では一般のユーザーがコンテンツを流通させることで，リアルタイムにコンテンツが提供・集積されるようになった．対象となるコンテンツには，楽曲，静止画像，動画像，テキストなどネットワーク上を流通するあらゆるコンテンツが挙げられる．そして様々なコンテンツが流通する中で多くのデータや情報が相互に活用され，そのプロセスを経て新たなアイデアや作品が生み出される機会が増えてきている．

デジタルコンテンツの流通や CGM サービスの拡大に伴って，デジタルコンテンツに関する著作権問題が取り上げられている．CGM サービスにおいては，他者の制作したコンテンツを素材とした加工や，自己の制作したコンテンツとの合成により新たな著作物を作るといったコンテンツの二次利用が特徴となっている．この時，作成したコンテンツを構成する元々のコンテンツがどういった著作物かを分かるようにしないと，コンテンツ制作者の著作権を保証することが出来ない．またデジタルデータに対し何の保護策も施さなければ，そのデータは自由に複製・編集され流通してしまう可能性がある．これによりコンテンツ制作者はデータの権利主張の機会損失を生じてしまうことになる．これまでコンテンツ保護に対し，コピー制御を中心とした技術が開発され取り上げられていた．しかし行き過ぎたコピー防止技術は，コンテンツを組み合わせ新たなコンテンツの作成といったコンテンツの n 次利用を制限してしまっている．そのためコンテンツの二次利用，三次利用といった n 次利用を考慮した著作権保護システムについて考えていく必要がある．

本論文ではコンテンツの n 次利用を考慮した上で，コンテンツの著作者のインセンティブを阻害されることのないよう，全てのコンテンツの不正利用を防ぎ著作権を保護する署名方式を提案することを目的とする．ここでは，コンテンツの n 次利用による新たなコンテンツの作成に対して，ボトムアップ型とトップダウン型に分類する．ボトムアップ型とは利用可能なコンテンツが多数あり，編集者がそれらの中からいくつかを取り上げて合成（その中に自分が作成したコンテンツを含んでもよい）していく方式であり，トップダウン型とは 1 つのコンテンツを基に編集者がその一部を削除・変更・追加していく方式[1][2]である．本論文ではボトムアップ型を考慮したアプリケーションに基づいたデジタルコンテンツの著作権を保証するためのシステムを考案する．

一般に，コンテンツ保護には暗号技術が利用される場合が多いが，本論文では，電子署名を用いた編集システムを提案する．提案方式では，それぞれのコンテンツに付随するメッセージに対し署名を行い，その複数の署名を用いて新たに制作したコンテンツの署名を作成することによって，n 次利用される全てのコンテンツの不正利用を防ぎ著作権を保証することが出来る．すなわち Aggregate 署名，GDH 署名[3,4]を拡張することで，新たに作成したコンテンツと関係する全てのコンテンツに関する著作権を保証し，さらに一次著作者，二次著作者といった全ての著作者の上下関係を規定する．またこのようなシステムについては，既に稲村らによって提案されている署名方式[5]がある．この方式は隣接する署名者間において，自身とその前者のメッセージに

† 東京理科大学大学院 工学研究科 電気工学専攻， 〒 102-0073 東京都千代田区九段北 1-14-6，Tokyo University of Science，1-14-6 Kudankita，Chiyoda-Ku，Tokyo 102-0073，Japan，shirakawa@sec.ee.kagu.tus.ac.jp
†† 東京理科大学 工学部 第一部 電気工学科， 〒 102-0073 東京都千代田区九段北 1-14-6， Tokyo University of Science，1-14-6 Kudankita，Chiyoda-Ku，Tokyo 102-0073，Japan，iwamura@ee.kagu.tus.ac.jp

署名し、その署名を順次合成する署名方式を提案しており，さらに隣接署名者間の関係性を一対多数にすることで，本論文の提案方式と同様の目的を実現している．しかし，新たにコンテンツの構成を作り上げる際に稲村方式だと１からコンテンツの構成をし直す必要が生じてしまう．

そこで本論文では，一度複数のコンテンツを組み合わせて作り上げたコンテンツを新たな他のコンテンツに組み合わせる場合を想定し，これを一次著作物と一次著作物以外の著作物とで検証方法を変えることで可能とした．これにより新たにコンテンツの構成を作り上げる際に１から構成を作り直す必要がなくなる．

以下，本論文では２章に提案方式に関する要素技術である GDH 署名，Aggregate 署名について説明を行う．３章で提案方式についてその安全性を含め説明を行い，４章において稲村方式との比較を行う．

2. 関連技術

2.1 GDH 署名

Ｇを素数位数 p の有限巡回乗法群とする．ここで g をＧの要素としたとき，Ｇにおいて DDH 問題，CDH 問題といった問題が定義される．

DDH 問題：Z_p^* の要素で互いに素な a,b,c があり，g, g^a, g^b, g^c が与えられた時，$c = ab$ かどうかを判定する問題.

ＣDH 問題：Z_p^* の要素で互いに素な a,b があり，g, g^a, g^b が与えられた時，g^{ab} を計算する問題.

ここで，DDH 問題は解くのが容易であるが，CDH 問題は解くのが困難とされるＧのグループのことを GDH グループという．

さらに GDH グループに基づいた GDH 署名について説明する．GDH 署名は任意の{0,1}*からなるメッセージ m に対し適応することができる．ここで全値域型ハッシュ法 h:{0,1}*→G*を定義し用い，また作成される署名 σ はＧの要素となる．この署名方法では鍵生成，署名，検証の３つのアルゴリズムで構成される．

・鍵生成：$x \in Z_p^*$ を選び，$v = g^x$ となる v を計算する．この v を公開鍵とし，x を秘密鍵とする．
・署名：秘密鍵 x とメッセージ $m \in \{0,1\}^*$ を用意し，メッセージからハッシュ $h = H(m)$ を計算する．h に対し x 乗したものを σ とし，$\sigma = h^x$ をメッセージｍの署名とする．
・検証：公開鍵 v とメッセージｍ，署名 σ を用意し，メッセージ m からハッシュ h を計算し，(g, v, h, σ)から $e(\sigma, g) = e(h, v)$ を計算することで検証する．

2.2 Aggregate 署名

Aggregate 署名とは，複数の署名を１つにまとめ，n 個のメッセージと n 個の公開鍵を用いることで一度の検証により全ての署名の安全性を確かめることのできるデジタル署名である．ここで 2-2 に記述されている GDH 署名で１つ１つの署名を作成し，作成した複数の署名を１つにまとめる署名方式について以下に説明する．

・鍵生成：複数の署名者がいたとき，全ての署名者にインデックス i を割り当てる．i 番目の署名者は $x_i \in Z_p^*$ を選び，$v_i = g^{x_i}$ となる v_i を計算する．この v_i を公開鍵とし，x_i を秘密鍵とする．
・署名：i 番目の署名者は秘密鍵 x_i とメッセージ $m_i \in \{0,1\}^*$ を用意し，メッセージからハッシュ $h_i = H(m_i)$ を計算する．h_i に対し x_i 乗したものを σ_i とし，$\sigma_i = h_i^{x_i}$ をメッセージ m_i の署名とする．
・アグリゲート：全ての署名者が作成した署名を集め $\sigma = \prod_i \sigma_i$ を計算する．σ を複数の署名を集約し新たに作成した署名とする.
・検証：検証者はアグリゲートした署名 σ に加え，全ての署名者のメッセージ，公開鍵 v_i を得ることができる.そしてアグリゲートした署名 σ を検証するために
(ⅰ)メッセージ m_i は全て異なっているか確認する．
(ⅱ) $h_i \leftarrow H(M_i)$ を計算し，$e(\sigma, g) = \prod_{i=1}^{k} e(h_i, v_i)$ かどうかを検証し，成功すれば正しく署名は作成されておりメッセージの正当性を確認することができる.

3. 提案方式

3.1 ボトムアップ型アプリケーションへの署名適応に関する概要

本提案方式では，想定するボトムアップ型アプリケーションに基づいた署名方式を提案する．以下にボトムアップ型アプリケーションについて説明する．

ネット上に利用可能なキャラ，BGM，設定などのコンテンツが複数存在し管理するサービスが存在するものとする．そして一次著作者は作成したコンテンツをサービスに登録し，使用されると利益を得る．ただし，サービスに登録するコンテンツは暗号または半開示状態で暗号化されており，許可されたユーザが有する正当な再生機または編集機以外では復元できないとする。二次著作者は登録されているコンテンツの利用をサービスに申請し，サービスが許可すればそのコンテンツの暗号化を復号でき，サービスが定めた利用条件の範囲において自由に組み合わせることで新たなコンテンツを作成し楽しむことができる．さらに二次著作者が作成したオリジナルコンテンツがあれば，それも組み合わせることもでき，作成した二次著作物をサービス全体に公開できる．また，そのサービスに登録している一般ユーザは公開された二次著作物を

楽しみ，アイデアがあれば三次著作者となる希望をサービスに申請することで，二次著作物と自分が作ったデータや他のデータを組み合わせて新たなコンテンツを作成することができるといったアプリケーションとなる．

提案方式ではサービス内の著作者のインセンティブが阻害されることのないよう，コンテンツの n 次利用を考慮した上で全てのコンテンツ作成者の著作権を保証する署名方式を考案している．提案する署名方式において，サービス内のコンテンツ再生機または編集機が署名方式における検証者の立場にあたる．この時，コンテンツの作成者自身がコンテンツに対する署名を行う署名者となる．さらにコンテンツには著作者の意図を含んだメッセージが付随している．具体的には，コンテンツの利用条件，コンテンツの作成者・構成情報等などがメッセージに記載されている．また提案方式の前提条件として，

・メッセージとコンテンツの正当性は保証されている．
・コンテンツの著作者は必ず正当に署名作成を行う．
・コンテンツを組み合わせ新たなコンテンツを作成する際は，必ず署名行為を行うものとする．
・署名鍵，検証鍵はサービスまたは公開鍵基盤(PKI:Public Key Infrastructure)などによって，正当なコンテンツ編集者のみに発行される．

が成立しているものとする．

3.2 アルゴリズム

複数のコンテンツを組み合わせて新たにコンテンツを作成する際の鍵生成，署名，検証過程を示す．署名は各々のメッセージまたはコンテンツからハッシュをとり，署名鍵とハッシュを用いて行っている．コンテンツを組み合わせる際，一次著作物となるコンテンツのみを組み合わせて作りあげたコンテンツと，既に何度かコンテンツを組み合わせて作り上げたコンテンツ（ここでは n 次著作物とする）と一次著作物となるコンテンツを組み合わせて作り上げる新たなコンテンツとでの鍵生成・署名過程が違うためそれぞれ 3.2.1，3.2.2 と別々に説明をする．なお 3.2.2 については，n 次著作物と一次著作物の組み合わせについて説明を行うが，n 次著作物と m 次著作物との組み合わせとした場合にも適用できる．また今回新たなコンテンツを作成する著作者を上位者，その際に組み合わせるコンテンツの複数の作成者を下位者とし，一対多数の二階層の場合を想定して説明する．

3.2.1 一次著作物のみを組み合わせて作り上げたコンテンツに関する署名方式

一次著作物のコンテンツを複数組み合わせ，新たなコンテンツを作り上げることを想定した署名方式について説明する．図 1 に多重署名過程を示し，さらに多重署名過程に基づいたアルゴリズムを示す．図 1 について簡単に説明すると，まず下位者はそれぞれ署名鍵 x_i を用いて検証鍵 v_i を作成する．さらに下位者は作成したコンテンツ（一次著作物）に付随するメッセージ m_i に対し署名 σ_i を作成する．ここで m はコンテンツでも構わないが，その場合は必ずコンテンツの構成情報等をコンテンツ自体に掲載することを前提にする．なお本論文では，以降 m はメッセージとして扱う．そして上位者は下位者が作成した署名を集め，自身が作成するコンテンツ（二次著作物）に付随するメッセージ m_k を用いてハッシュ h_k を計算し，自身の署名鍵 x_k とハッシュ h_k と下位者から集めた署名 $\sigma_1,\sigma_2,\sigma_3$ を用いて新たな署名 σ_k を作成し公開する．さらに上位者は，下位者が作成した検証鍵 v_1,v_2,v_3 と自身の署名鍵 x_k を用いて V_1,V_2,V_3 といった新たな検証鍵のペアを作成し公開する．検証者は，コンテンツの構成情報を確認した上で全てのコンテンツのハッシュを計算する．そして検証に必要な値を集め，ペアリング演算 $e(\sigma_k,g)=\prod_i(h_i^{h_k},v_i^{x_k})$ を計算し，演算が成立するか否かで上位者の作成した署名を検証するといった流れとなる．よって，具体的なアルゴリズムは以下になる．

署名: $\sigma_k=(\sigma_1\times\sigma_2\times\sigma_3)^{x_k h_k}=(h_1^{x_1}h_2^{x_2}h_3^{x_3})^{x_k h_k}$

検証鍵: $V=\{V_1,V_2,V_3\}=\{v_1^{x_k},v_2^{x_k},v_3^{x_k}\}$

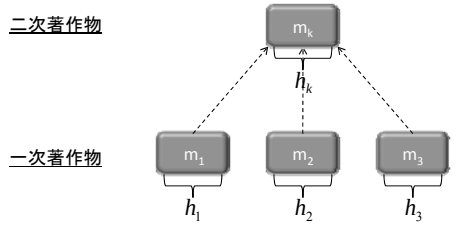

図 1　多重署名過程（3.2.1）

１．下位者（一次著作物の作成者）

鍵生成：i 番目の下位者は $x_i\in Z_p^*$ を選び，x_i を署名鍵とする．次に g と署名鍵 x_i を用いて検証鍵 $v_i=g^{x_i}$ を計算する．

署名：i 番目の下位者は m_i を用いて $H_2(m_i)=h_i$ を計算する．そして h_i を用いて $\sigma_i=h_i^{x_i}$ を計算し，この σ_i を i 番目の下位者の署名とし上位者へ送信する．また m_i はコンテンツに付随するメッセージとする．

※H_2 は $H_2:\{0,1\}^*\rightarrow G$ の機能を持つハッシュ関数とする．

２．上位者

鍵生成：上位者は $x_k \in Z_p^*$ を選び，x_k を署名鍵とする．次に g と署名鍵 x_k を用いて上位者の検証鍵 $v_k = g^{x_k}$ を計算する．また上位者は下位者から受け取った検証鍵を集め，上位者の署名鍵 x_k を用いて $V = \{V_1, V_2, V_3\} = \{v_1^{x_k}, v_2^{x_k}, v_3^{x_k}, \cdots\}$ を作成する．

署名：上位者は m_k を用いて $H_1(m_k) = h_k$ を計算する．そして上位者は下位者から受け取った署名を集め，さらに h_k, x_k を用いて
$\sigma_k = \prod_i (\sigma_i^{h_k})^{x_k} = \prod_i (h_i^{x_i})^{x_k h_k}$ を計算し，σ_k を上位者の署名とする．また m_k はコンテンツに付随するメッセージとする．

※H_1 は H_1:$\{0,1\}^* \times G \rightarrow Z_p^*$の機能を持つハッシュ関数とする．

３．検証者

検証鍵検証：全ての下位者の検証鍵 v_i，上位者が作成した V_i について，上位者の検証鍵 v_k を使ってペアリング演算を行い，$e(V_i, g) = e(v_i, v_k)$ が成立すれば上位者が作成した新たな検証鍵 V_i は正当である．

署名検証：検証者は上位者の m_k，下位者の m_i を用いて，
・上位者による値 $h_k : h_k = H_1(m_k)$
・下位者（一次著作物の作成者）による値 $h_i : h_i = H_2(m_i)$
を計算する。そして検証者は g, v, h_k, h_i, σ を用いてペアリング演算を行い，
$e(\sigma_k, g) = \prod_i (h_i^{h_k}, v_i^{x_k})$ が成立すれば署名 σ は正当な署名となる．

3.2.2 一次著作物と n 次著作物を組み合わせて作り上げたコンテンツに関する署名方式

一次著作物や n 次著作物のコンテンツを複数組み合わせ，新たなコンテンツを作り上げる際を想定した署名方式について説明する．図 2 に多重署名過程を示し，さらに多重署名過程に基づいたアルゴリズムを示す．ここで図 2 について簡単に説明する．3.2.1 の手法により，m_1, m_2, m_3 といったメッセージが付随している３つのコンテンツを組み合わせた，メッセージ m_p が付随した n 次コンテンツ（署名：σ_p'，検証鍵：v_p'）が存在していたとする．そしてこのコンテンツの検証情報として n 次コンテンツの作成者が δ（δ は後述）を計算し，署名 σ_p'，検証鍵 v_p' とともに保持する．次にこのコンテンツとメッセージ m_t, m_q …が付随した一次コンテンツや m 次コンテンツを組み合わせて，m_k が付随するコンテンツを作成したとする．以下，m_t, m_q は一次コンテンツとして説明するが，m_p と同様の構成をもつ m 次コンテンツであってもよい．

一次コンテンツにおける署名，検証鍵生成は 3.2.1 と同様に行い，n 次コンテンツについては署名 σ_p'，検証鍵 v_p' を PKI により新たに生成された署名鍵 x_s を用いて，新しく署名 σ_p，検証鍵 v_p を作成する．そして上位者は 3.2.1 と同様に署名 σ_k，検証鍵ペア $V_t, V_q, \cdots V_p$ を作成し公開する．検証者は，コンテンツの構成情報を確認した上で検証に必要な値を集める．ここで，一次著作物における h_i はメッセージ m_i からハッシュを計算することで求まるが，n 次著作物における h_i は m_i とコンテンツの検証情報として保持してある δ を用いて $h_i = \delta^{H_1(m_i)}$ を計算するといった違いがある．また δ は h_i を求める際にしか用いられず，新たな署名 σ_p，検証鍵 v_p の扱いは 3.2.1 と同様のものとする．そして最後に検証者はペアリング $e(\sigma_k, g) = \prod_i (h_i^{h_k}, v_i^{x_k})$ を計算し，下記関係が成立するか否かで上位者の作成した署名を検証する．

すなわち，
$e(\sigma_k, g) = \prod_i e(\sigma_i^{h_k x_k}, g) = \prod_i e(h_i^{h_k}, v_i^{x_k})$ となる．

一次著作物については 3.2.1 より自明であるので，n 次著作物である m_p における上記のペアリング演算を考える．

$$e(\sigma_p{}^{h_k x_k}, g) = e(\sigma_p{}'^{x_s h_k x_k}, g) = e(\delta^{h_p{}' x_p x_s h_k x_k}, g)$$

$$e(h_p{}^{h_k}, v_p{}^{x_k}) = e(\delta^{h_p{}' h_k}, v_p{}'^{x_s x_k}) = e(\delta^{h_p{}' h_k}, g^{x_p x_s x_k})$$

となることから，

$$e(\sigma_p{}^{h_k x_k}, g) = e(\delta^{h_p{}' x_p x_s h_k x_k}, g) = e(\delta^{h_p{}' h_k}, g^{x_p x_s x_k})$$

$$= e(h_p{}^{h_k}, v_p{}^{x_k})$$

となり $e(\sigma_p{}^{h_k x_k}, g) = e(h_p{}^{h_k}, v_p{}^{x_k})$ が成り立つことから，上記の署名検証が成立することが分かる．すなわち，n 次著作物に対しては δ を介して，一次著作物の署名を含めて検証できていることがわかる。

よって，具体的なアルゴリズムは以下になる．

Vol.2010-CSEC-49 No.7
2010/5/11

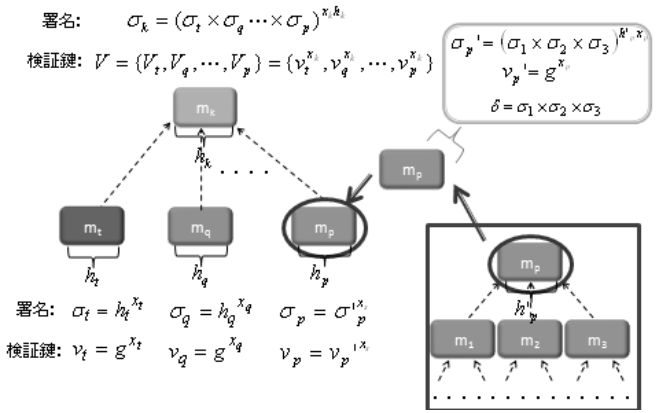

図 2　多重署名過程（3.2.2）

１－１．下位者（一次著作物の作成者）
3.2.1 の図１における１．と同様の鍵生成・署名を行う．

１－２．下位者（n 次著作物のコンテンツの作成者）
鍵生成：i 番目の下位者は $x_i \in Z_p^*$ を選び，x_i を新たな署名鍵とする．また x_i と既に所持している検証鍵 v_i' を用いて $v_i = v_i'^{x_i}$ を計算し，v_i を新たな署名鍵とする．
署名：既に作成してある署名 σ_i' と署名鍵 x_i を用いて $\sigma_i = \sigma_i'^{x_i}$ を計算し，σ_i を新たな署名とし上位者へ送信する．また σ_i を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．

２．上位者
3.2.1 の図１における２．と同様の鍵生成・署名を行う．

３．検証者
検証鍵検証：全ての下位者の検証鍵 v_i，上位者が作成した V_i について，上位者の検証鍵 v_k を使ってペアリング演算を行い，$e(V_i, g) = e(v_i, v_k)$ が成立すれば上位者が作成した新たな検証鍵 V_i は正当とする．
署名検証：検証者は上位者の m_k，下位者の m_i, δ を用いて，
・上位者による値 $h_k : h_k = H_1(m_k)$
・下位者（一次著作物の作成者）による値 $h_i : h_i = H_2(m_i)$
・下位者（n 次著作物のコンテンツの作成者）による値 $h_i : h_i = \delta^{H_1(m_i)}$
をそれぞれ計算し，そして検証者は g, v, h_k, h_i, σ を用いてペアリング演算を行い，$e(\sigma_k, g) = \prod_i (h_i^{h_k}, v_i^{x_k})$ が成立すれば署名 σ は正当な署名となる．

3.3 安全性

提案方式における署名方式は，GDH グループの条件にある CDH 問題の困難性，また指数部分の積で求まっている値の構造を解消する必要性により，署名や検証鍵の偽装は不可能となっている．これを 3.2.1 を例に挙げて考えてみる．図１の状況を仮定し，m_1, m_2, m_3 を組み合わせて m_k を作成する上位者Ｋがいたとする．上位者の署名 σ_k について考えてみると，σ_k は $\sigma_k = \prod_i (\sigma_i^{h_k})^{x_k}$ で表される．ここで第三者Ｓが悪意を持ってコンテンツ m_1, m_2, m_3 を組み合わせ，コンテンツ m_s を作成したとする．Ｓが仮に公開情報から $\prod_i \sigma_i^{h_k} = (\sigma_1 \times \sigma_2 \times \sigma_3)^{h_k}$ を入手することが出来たとしても，上位者の署名鍵 x_k は非公開なため $\sigma_k' = \prod_i \sigma_i^{x_k} = (\sigma_1 \times \sigma_2 \times \sigma_3)^{x_k}$ を計算することができない．また前提条件より，署名鍵，検証鍵は正当なコンテンツ編集者のみに発行されるためＳには鍵ペアは発行されない．ここでＳはＫの公開情報の検証鍵を用いて，自身の検証鍵 V_{si}（$V_{si} = V_i$）とすることができたとして，偽造が成功するためにはＳの署名 $\sigma_s = \prod_i (\sigma_i^{h_s})^{x_k}$（$m_s$ のハッシュ h_s を用いた署名）の計算が可能とならなければ検証は成功しないが，Ｓは σ_k' を求めることが出来ないため σ_s は計算できず（$\sigma_s = \sigma_k'^{h_k}$ のため）署名の偽造は不可能となる．また 3.2.2 の図２におけるコンテンツ m_k についても同様に，m_k の著作者の署名鍵 x_k が非公開となっているため $\sigma_k' = (\sigma_t \times \sigma_q \cdots \times \sigma_p)^{x_k}$ を計算することができず，署名の偽造は不可能となる．以上より，検証によって署名の正当性を正確に確かめることができる．

4. 本提案方式と稲村方式との比較

本提案方式により，それぞれのコンテンツに付随するメッセージに対し署名を行い，その複数の署名を用いて新たに制作したコンテンツの署名を作成し検証を行うことで，コンテンツの n 次利用における著作権保護の実現を可能とした．また稲村方式も提案方式２において同様の目的を実現可能としており，両方式の比較について述べる．
稲村新方式は GDH 署名方式を Aggregate 署名として拡張し，２つの提案方式を考案している．提案方式１は，複数の署名者がいた時，隣接する署名者間の前後において

自身とその前者のメッセージに署名し，署名を順次合成し署名順序まで検証することができる署名方式となっている．提案方式２は，提案方式１の隣接署名者の関係を一対多数に拡張することで実現されている．新たに作成したコンテンツの構造は一から全て構成することを前提とし，隣接する階層間の前後において上位階層は下位階層のそれぞれのメッセージと上位階層のメッセージに対し署名し，この署名を順次合成していくことで新たに作成したコンテンツの署名を作成し検証する．これによって，新たに作成したコンテンツと関係する全てのコンテンツに関する著作権を保証し，コンテンツの階層構造まで検証できる署名方式を提案している．しかしこの方式ではコンテンツの構造は一から全て構成するという前提のため，既存コンテンツを組み合わせて新たにコンテンツの構成を作り上げる場合にも，全てのコンテンツに対して署名作成処理をし直す必要が生じてしまい，計算量が大きい．

それに対し本論文の提案方式では，一度作り上げたコンテンツは署名の再計算無しに他のコンテンツに組み合わせることができるということを前提とする．よって，コンテンツ作成においては複数の署名を合成し新たに署名を作り上げ，署名と検証鍵とそのコンテンツに関する検証情報を保持した一つのデータ情報として他の階層に組み込むことが可能であり,それを用いて上記と同様の目的を実現する方式となっている．この特徴により，本提案方式では新たにコンテンツの構成を作り上げる際に全てのコンテンツに対して署名作成処理を１からし直す必要がない．またそのために，コンテンツ再構成時の計算量が少なく，かつ検証時における入力情報量が稲村方式と比べて少なくなっている．具体的には，本提案方式は検証時に最終的に作り上げたコンテンツとそのひとつ前の階層における署名者の情報を入力することで検証が可能となるのに対し，稲村方式は全ての署名者の情報を入力する必要がある．

ただし、本提案方式では検証鍵の個数が稲村方式と比べて多くなってしまう問題点がある．コンテンツを組み合わせ新たなコンテンツを作成する際，下位階層のコンテンツの検証鍵 v_i に対し新たに検証鍵 V_i を上位階層の署名者によって作成する必要がある．そのため稲村方式は全ての署名者の数だけ検証鍵を生成するのに対し，本提案方式は稲村方式と比べて約２倍の量の検証鍵を生成しなくてはならない．よって，検証鍵の生成量について今後考察していく必要がある．

5. まとめ

コンテンツに付随するメッセージに対し署名を行い，コンテンツの n 次利用を考慮した上で全てのコンテンツ作成者の著作権を保証する署名方式を提案した．提案方式では GDH 署名と Aggregate 署名を組み合わせ，ボトムアップに適した署名方式を可能としている．また，一度ボトムアップで作り上げたコンテンツを新たな他のコンテンツに組み合わせる場合を想定する構想を署名方式に組み込むことで，何度コンテンツを他のコンテンツと組み込んでも署名の正当性を検証することができるようになった．しかし本論文の提案方式では，一度ボトムアップで作りあげたコンテンツの中身の編集（追加，削除，変更等）には対応していない．よって今後の方針として，署名方式を拡張することで以上の問題について検討を進めていく予定である．

参考文献

[1] 齊藤 旭，柿崎 淑郎，岩村 惠市：“コンテンツの追加制御可能な電子署名システム”，2009 年 暗号と情報セキュリティシンポジウム，3B3-1，Jan.2009.
[2] 齊藤 旭，山田 裕也，岩村 惠市：“編集可能コンテンツに対する墨塗り署名を用いた 電子署名システムの提案”，第 39 回コンピュータセキュリティ研究会, pp.49-54. Dec.2007.
[3] D. Boneh, C. Gentry, B. Lynn, H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”, EUROCRYPT 2003, LNCS2656, pp. 416-432, 2003.
[4] D. Boneh, B. Lynn, and H. Shacham, “Identity-Based Encryption from Weil Pairing” Advances in Cryptology-CRYPTO2001,LNCS2139,Springer-Verlag,pp213-229,2001.
[5] 稲村勝樹，渡辺龍，田中俊昭，”Gap Diffie –Hellman に基づいた順序付きアグリゲート署名とその拡張方式”， The 2010 Symposium on Cryptography and Information Security,p19-22,2010.