電子署名を用いたコンテンツの編集制御方式の実装

田中　大地[†1]　岩村　恵市[†1]　稲村　勝樹[†2]

概要：昨今のインターネットが発達した社会では、一般の消費者がネット上にアップロードされたコンテンツを編集して新たなコンテンツを作成することが容易になっている。それに対して、電子署名を用いてコンテンツの編集を制御できる著作権保護方式が提案されている。この方式は著作者が自らのコンテンツの各部分に対して、変更・削除・追加・流用を制御する署名を設定でき、コンテンツそのものの合成も署名を用いて制御できる。本発表ではこの署名方式の活用例として、コンテンツ編集ツールである「MikuMikuDance」で作成されるコンテンツに対して先述した署名方式を実装し、編集を制御できるコンテンツ保護方式の実用性を示す。

キーワード：著作権保護、編集制御、電子署名、アグリゲート署名、コンテンツ保護

Implementation of editing control method of the content using an electronic signature

Daichi Tanaka[†1]　Keiichi Iwamura[†1]　Masaki Inamura[†2]

Abstract: Recently, in society developed internet, it become easy that ordinary consumers create a new content by editing the content uploaded on the internet by other user. In contrast, the copyright protection system capable of controlling the editing of the content using the digital signature been proposed. This method is the author can set a signature to control the change, delete, addition, and diversion, also the contents itself can be controlled using electric signature. In this presentation, I present utility of the content protection system that can control editing by using contents created by content editing tool "MikuMikuDance".

Keywords: Copyright protection , edit control , electronic signature , aggregate signature , content protection

1. はじめに

昨今のインターネットでは、従来の出版社が出す雑誌やテレビで放送される映像などといったメディアなどに比べて、消費者がコンテンツを制作し、そのコンテンツを容易に公開、流通させることができるようになっている。

これまでのコンテンツ配信サービスでは、特定の発信者が一般ユーザーに対してコンテンツを一方向に提供するサービスが主流であったが、近年ではインターネットという媒体を用いてユーザーが誰でも発信者になることができるようになった。このようなコンテンツ流通プラットフォームは消費者生成メディア (CGM: Consumer Generated Media)と呼ばれていて、CGM サービスを扱うサイトとしては YouTube[1]やニコニコ動画[2]などがある。

この CGM サービス内では、新たなコンテンツを作成するだけではなく、あるユーザーが作ったコンテンツを編集・引用し、これを新しいコンテンツとして公開していくコンテンツ循環が行われていることも多く、また元のコンテンツ製作者もこのような流通が展開されることを望んだ上でコンテンツを作成することがある。

このような環境の中では、現在の著作権保護技術で主流となっている視聴制御やコピー制御技術では、コンテンツの編集や追加に制限が掛かってしまうため、CGM コンテンツには不向きである。コンテンツ循環が想定される場合は、コンテンツが 2 次利用されても元の著作者が守られる権利継承や編集制御といった新たな著作権保護技術が求められる。

それに対して、[4]では上記権利継承や編集制御を実現する著作権保護技術が電子署名を用いた方式として提案されている。ここで、権利継承とは、あるコンテンツが二次利用されたとき、二次コンテンツの中でそのコンテンツの著者の権利が保障されることである。また、編集制御とは著作者がコンテンツに対して、二次コンテンツとして利用する範囲を制御するための技術である。

これらの技術を web サービス上で実現するために、実際にコンテンツ作成環境である”MikuMikuDance”で作成されたコンテンツについて、この署名方式を適用し、署名作成と署名の検証のデモを行うことができるプログラムのプロトタイプを作成した。

本稿では、2 章で[4]に記された署名継承と編集制御を同時に実現可能なコンテンツ保護方式およびそれに必要な基礎知識を説明し、3 章で作成したプログラムでの”MikuMikuDance”のコンテンツ作成環境より電子署名を作成するコンテンツ保護方式のシミュレーション環境、シミュレーション方法について示し、4 章でシミュレーション結果及びについて示し、5 章においてまとめとする。

†1 東京理科大学
Tokyo University of Science
†2 東京電機大学
Tokyo Denki University

2. 使用する認証方式について

以下に今回のコンテンツ保護方式で用いる署名作成方式を示す。

2.1Aggregate 署名

Aggregate 署名は Boneh らによって提案された署名方式である。この署名方式では複数の署名者が各自の文書に対して生成した署名を 1 つに集約することが可能であり、検証者はその集約された署名の検証を通じて全ての個別署名を検証することができる。Aggregate 署名の説明を以下に示す。

U を署名に参加するユーザーの集合とし、それぞれのユーザー$u_i \in U(1 \le i \le n)$は1つの鍵ペア(pk_{ui}, sk_{ui})を持つとする.各ユーザーは署名対象であるm_{ui}を選び,それに対して署名σ_{ui}を生成する。そして、これらの署名は 1 つの Aggregate 署名へと結合される。Aggregate 署名は GDH 署名に基づいて、安全性は co-CDH 問題に依存している。そして、そのアルゴリズムは鍵生成、署名、集約、検証の 4 つからなる。また BLS 署名の説明と同様(G_1, G_2)における GDH グループ上での定義を利用する。以下にそのアルゴリズムを示す。

1. 鍵生成 : 署名者$u_i \in U$について$x_{ui} \in Z_p$を選択し、$v_{ui} \leftarrow \mathrm{g}_2^{x_{ui}}$を計算する。$x_{ui}$を署名に使用する秘密鍵とし、 v_{ui}をその検証鍵とする。
2. 署名 : 一方向性ハッシュ関数$H: \{0,1\}^* \rightarrow G_1$を定義する。$m_{ui}$を各ユーザーの署名対象となる平文として、$\sigma_{ui} \leftarrow H(m_{ui})^{x_{ui}}$を計算する。そして$\sigma_{ui}$を$m_{ui}$に対するデジタル署名とする。
3. 集約 : 個別署名σ_{ui}を全て集め$\sigma \leftarrow \prod_{i=1}^{n} \sigma_{ui}$を計算する。
4. 検証 : 検証者が$1 \le i \le n$までの公開鍵をv_{ui}、平文m_{ui}と集約された署名 σを得ているとき、$h_{ui} \leftarrow (m_{ui})$を計算する。そして$e(\sigma, \mathrm{g}_2) = \prod_{i=1}^{n} e(h_{ui}, v_{ui})$であるかを判定する。
5. 双線形写像の特性によって $e(\sigma, \mathrm{g}_2) = e\left(\prod_{i=1}^{n} h_{ui}^{x_{ui}}, \mathrm{g}_2\right) = \prod_{i=1}^{n} e\left(h_{ui}^{x_{ui}}, \mathrm{g}_2\right) = \prod_{i=1}^{n} e(h_{ui}, v_{ui})$と展開され、それぞれの平文の正当性を署名 σによって検証できる。

2.2 順序付き Aggregate 署名

Aggregate 署名では、複数の署名を 1 つに集約することが可能である。しかし、署名を作る際、その複数の署名間の関係を規定することとそれを判断することができない。そのため、前後の署名者間の関係を示す演算を導入し、前後の署名の順序を規定する順序付き Aggregate 署名が提案された。以下に順序付き Aggregate のアルゴリズムを示す。

順序付き Aggregate 署名において使用される記号，および前提条件について以下に示す。

記号:

$\mathrm{G}_1, \mathrm{G}_2$: ペアリング演算が可能な楕円曲線上の点集合

g: G_1の要素である生成元

e: ペアリング関数

u_o: o 番目の署名者

$\mathrm{x}_o, \mathrm{v}_o$: u_oの署名鍵，および検証鍵

L_o: u_1からu_oまでの署名順序情報

m_o: u_oが署名対象とする平文

$H: \{0,1\}^* \rightarrow G_2$となる一方向性ハッシュ関数

σ_o: u_1からu_oまでの順序付き Aggregate 署名

2.2.1 鍵生成

$g \in \mathrm{G}_1$を生成元とする。署名者u_1について$\mathrm{x}_i \in Z_p^*$を選び（全ての署名者の署名鍵は各々異なるものとする）、$\mathrm{v}_i = \mathrm{x}_i g$ を計算する。

2.2.2 署名生成

以下の手順で，Aggregate 署名が作成される。

(1)第 1 署名者u_1は、平文m_1から$h_1 = \mathrm{H}(m_1)$ を求め、BLS 署名と同様な署名作成処理を行うことで$\sigma_1 = x_1 h_1$を計算する。また、$L_1 = \{0, \mathrm{u}_1\}$を作成する。この$\sigma_1$、$L_1$、L1 および$m_1$（または$h_1$）を第 2 署名者$\mathrm{u}_2$に送信する。

(2)第 i 署名者u_iは、第 i-1 署名者u_{i-1}から受信したm_{i-1}を用いて$\mathrm{h}_{i-1} = H(\mathrm{m}_{i-1})$を求める。さらに署名者$\mathrm{u}_i$は、自分が本来署名したい平文$\mathrm{m}_i$から$\mathrm{h}_i = H(\mathrm{m}_i)$を求める。これと署名者$\mathrm{u}_{i-1}$から受信した$\sigma_{i-1}$を用いて

$$\sigma_{\mathrm{i}} = \sigma_{\mathrm{i}-1} + x_i h_{i-1} + x_i h_i = x_1 h_1 + \sum_{j=2}^{i} (x_j h_{j-1} + x_j h_j)$$

を計算する。また、受信したL_{i-1}を用いて

$$\begin{aligned} L_i &= L_{i-1} + \{(\mathrm{u}_{i-1}, \mathrm{u}_i)\} \\ &= \{(0, \mathrm{u}_1), (\mathrm{u}_1, \mathrm{u}_2), \cdots, (\mathrm{u}_{i-1}, \mathrm{u}_i)\} \end{aligned}$$

を作成する。σ_{i}、L_iおよびm_i（またはh_i）を第 i+1 署名者u_{i+1}に送信する。

この手順を最後から一人前の署名者まで再帰的に行う。

(3)最後の署名者u_nは、直前の署名者u_{n-1}から受信したm_{n-1}を用いて$\mathrm{h}_{n-1} = H(\mathrm{m}_{n-1})$を求める。さらに署名者$\mathrm{u}_n$は，文$\mathrm{m}_n$から$\mathrm{h}_n = H(\mathrm{m}_n)$を求める。これと署名者$\mathrm{u}_{n-1}$から受信した$\sigma_{n-1}$を用いて

$$\begin{aligned} \sigma_i &= \sigma_{\mathrm{i}-1} + x_n h_{n-1} + x_n h_i \\ &= x_1 h_1 + \sum_{j=2}^{n} (x_j h_{j-1} + x_j h_j) \end{aligned}$$

を計算する。また、受信したL_{n-1}を用いて

$$\begin{aligned} L_n &= L_{n-1} + \{(\mathrm{u}_{n-1}, \mathrm{u}_i)\} \\ &= \{(0, \mathrm{u}_1), (\mathrm{u}_1, \mathrm{u}_2), \cdots, (\mathrm{u}_{n-1}, \mathrm{u}_n)\} \end{aligned}$$

を作成する。

このようにして作成したσ_n、L_nおよびm_n（またはh_n）を第 n+1 署名者u_{n+1}に送信する。

2.2.3　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_nに示されている全ての署名者の検証鍵 $v_1, \cdots, v_n$、および署名対象となる
全ての平文$m_1, \cdots, m_n$を集める。

(2) 検証者は、集めた平文から$\mathrm{h}_i = H(\mathrm{m}_i)$を求める。

(3) 検証者は

$$e(v_1, \mathrm{h}_1)(\prod_{j=2}^{n} e(v_j, \mathrm{h}_{j-1}) = e(v_1, \mathrm{h}_1)e(v_2, \mathrm{h}_1 + \mathrm{h}_2) \cdots e(v_n, \mathrm{h}_{n-1} + \mathrm{h}_n)$$

を計算し、この値と$e(g, \sigma_n)$の値が一致することを確認する。

2.3 木構造表記型 Aggregate 署名

順序付き Aggregate 署名は、署名者が一列方向で順番に署名を作成する際に、前後の署名者間の関係を示す演算を導入し、その手順を繰り返すことで、署名者の順序を規定している。この手順を 1 対 1 のみではなく隣接する署名者に対応して 1 対多に拡張し、多段的に繰り返すことで、木構造表記型 Aggregate 署名を構成することが可能となる。

2.4 権利継承と編集制御を同時に実現する コンテンツ制御方式

2.4.1 エンティティ

権利検証と編集制御を同時に実現するコンテンツ制御方式における役割を持つ二つのエンティティを以下のように定義する。

i 次著作者　あるコンテンツの制作・編集に係り、自分が設定可能なコンテンツの編集制御署名を設定する。以下の図 1 のように木構造を定義したとき、木構造の一番深いところになっている、いわゆる親のコンテンツを持つ著作者を 1 次著作者とし、ルートとなっている著作者を n 次著作者とする。

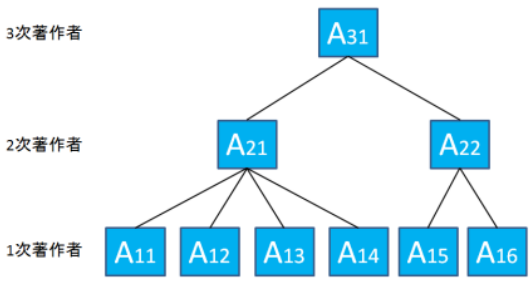

図 1:木構造の例

検証者　正当な署名のあるコンテンツかどうかを検証する。この役割をプレーヤーなどのコンテンツを再生する機器に持たせることで、正当な署名を持たないコンテンツを再生できないようにすることができる。

2.4.2 コンテンツ及び部分コンテンツ

著作者は l つ以上の部分コンテンツを作成し、その集合をコンテンツとして公開する。一つのコンテンツ内で、各部分コンテンツに対する追加・削除・変更制御に加えて、部分コンテンツの流用に関する制御も行い、コンテンツの合成も制御可能にする。ここで、部分コンテンツに含まれるデータとして実データと空データを定め、削除とはコンテンツを空データとすること、追加とは空データを実データに置き換えることを示すこととする。

これらのデータのほかにも、開始データと最終データという制御用データをコンテンツの先頭と末尾に配置する。

2.4.3 コンテンツ内の編集制御

コンテンツを編集制御するための署名として、以下のふたつの署名をそれぞれ専用の乱数と部分コンテンツのハッシュ値より生成する。

・変更制御用署名：この署名が公開されていればこの部分コンテンツは変更可能である。

・削除制御用署名：この署名が公開されていればこの部分コンテンツは削除可能である。

2.4.4:コンテンツ間の編集および権利継承

コンテンツ間での編集として、他コンテンツへの流用及びコンテンツ間合成を考えた時、著作権を侵害されずとも流用を制限したい意図があるときのために、流用制御署名を流用制御用の乱数と部分コンテンツのハッシュ値より流用制御署名を作成し、開始データと最終データのそれぞれのハッシュ値を署名して前合成制御署名、後合成制御署名を作成する。

2.4.5:実装を行う上での制約

本稿では、この署名保護システムを実現するためとして、実装を行うために上記の制御方式にいくつかの暫定的な制約を加えた。

・検証者と著作者両方が変更、削除の権限の変更をできるようになっている

・編集操作として可能な操作が同じ部分コンテンツから同じ部分コンテンツへ変更する「更新」と部分コンテンツを削除する「削除」のみとなっている。

3. “MikuMikuDance”上での実装について

以下に、本稿で署名作成プログラムの動作対象とする”MikuMikuDance”でのコンテンツを元にした署名作成および署名検証方式について示す。“MikuMikuDance”で作成されたコンテンツは 3DCG を用いたムービーとして公開されるが、後述のように.pmm ファイルなどを用いてその容姿や動きなどを規定している。よって、[4]における電子署名は 3DCG を用いたムービーではなく、.pmm ファイルなどのオリジナルソースに対して実施される。これによって、オリジナルソースに対して、[4]に示す編集制御や権利継承が実現できるため、オリジナルソース自体を流通させてもその著作者の著作権が保護される。

3.1 MikuMikuDance とは

MikuMikuDance とは、樋口優氏が無償公開[5]しているフリーの 3DCG ムービー制作ツールである。当初は初音ミクというキャラクターのダンスムービーの制作のために作られたソフトであるが、バージョンアップを重ねて初音ミクなどのキャラクター以外のキャラクターのモデルを読み込む事ができるようになり、ダンス以外にも様々な動画が作れるようになった。現在ではあらゆるジャンルの 3DCG プラットフォームとして機能している。

MikuMikuDance 固有の特徴としては、3DCG 制作のツールとしての敷居が低く、素材が多く公開されていて、モデルの他にも、小道具、背景などの素材が公開されているので、登場以前に比べて、3DCG ムービー作成そのもののハードルが大きく下がることとなった。

以下の図 2 に、MikuMikuDance でのコンテンツ製作のようすを示す。

図 2:MikuMikuDance でのコンテンツ制作画面

3.2MikuMikuDance で作成するコンテンツの構成

MikuMikuDance では、ムービー制作の素材として、各種 3D モデルや音楽ファイルを読み込み、編集して 3DCG ムービーを作成する。そのため、ここでは先述したコンテンツの構成について、[4]と実際の MikuMikuDance について以下の表 1 のような対応を定義して、署名作成ツールおよび署名検証プログラムを作成した。

表 1:本実装での各コンテンツの対応関係の定義

[4]での定義	″MikuMikuDance″上での定義
集約コンテンツ	3DCG ムービー
3DCG ムービーを構成するコンテンツ	3D モデル
3D モデルを構成する部分コンテンツ	モデルのボーン（モデルの形を構成するデータ）

3.3 プログラムの仕様

本稿で作成した著作権保護プログラムでは、以下のように署名を作成、検証する。

3.3.1:署名対象の抽出プログラム

“MikuMikuDance”で作成した.pmm ファイルを指定して、署名対象ファイル(csv 形式)を作成する。

.pmm ファイルには作成しているムービーで使用されているモデルのボーンなどの、ある程度の情報が入っている。そのため、これを抽出して署名対象として列挙する。

また、この時に、著作者の立場から、変更、削除の可否を設定することができる。

署名対象抽出プログラムはコマンドライン上で動作する。その様子を図 3 に示す。そうして作成された署名対象ファイルを図 4 に示す。また、図 3 にはファイルのパスを示す文字列があるが、一部隠している。

なお、署名対象抽出プログラムは後述する TEPLA[6]を用いた暗号処理に用いる計算手法は用いておらず、プロジェクトファイルから標準の入出力関数を用いてファイルを抽出している。

```
d-tanaka@dtanaka-VirtualBox:/media/sf_ubuntu/files/m1files$ ./makecsvfrompmm 2
model no:2
now model no:0
model jp name:◆◆◆◆◆~◆N
model en name:Miku Hatsune
file name:C:\Users\       \Documents\MikuMikuDance_v926x64\MikuMikuDance_v926x64\
◆◆~◆N.pmd
bone count:122
skin count:16
ik count:7
now model no:1
model jp name:◆◆◆◆◆◆◆J
model en name:Luka Megurine
file name:C:\Users\       \Documents\MikuMikuDance_v926x64\MikuMikuDance_v926x64\
◆◆◆◆J.pmd
bone count:188
skin count:40
ik count:4
time dist is:0.008421[s]
d-tanaka@dtanaka-VirtualBox:/media/sf_ubuntu/files/m1files$
```

図 3:署名対象抽出プログラムの動作画面

	A	B	C	D	E	F	G
1	0	start	313	1	2	1	124
2	1	Miku Hatsu	0	0	2	2	123
3	2	センター	0	0	0	0	
4	3	上半身	0	0	0	0	
5	4	首	0	0	0	0	
6	5	頭	0	0	0	0	
7	6	左目	0	0	0	0	
8	7	右目	0	0	0	0	
9	8	ﾈｸﾀｲ1	0	0	0	0	
10	9	ﾈｸﾀｲ2	0	0	0	0	
11	10	ﾈｸﾀｲ3	0	0	0	0	
12	11	下半身	0	0	0	0	
13	12	腰飾り	0	0	0	0	
14	13	左髪1	0	0	0	0	

図 4:署名対象ファイルを excel にて閲覧した画面

3.3.2:署名プログラム

3.3.1 で作成した署名対象ファイルに、[4]での著作権保護方式にて提案されている手法にて、署名作成処理を行い、集約署名と公開されるべき検証鍵やメッセージを署名ファイル(.csv 方式とする)として出力する。

署名作成プログラムはコマンドライン上で動作する。その様子を図 5 に示す。そうして作成された署名対象ファイルを図 6 に示す。

```
d-tanaka@dtanaka-VirtualBox:/media/sf_ubuntu/files/m1files$ ./csvtosign
init tepla
time dist is:0.078221[s]
read uservalue
1   124   uservalue:313
time dist is:0.000466[s]
get user info
init
get usr mess
313
1

make key
start type1
start type2
2 to 123
start type2
125 to 312
make sig file
0   1   1   2
sign make complete
sig::: [12e17532bf8a17b902c5777fc19ca44cc1f255bea2bea2384d1825a0b2f1f7e5 22e2590
aeff5af3dbfe1860b8e4e8be6de2dd8769bd967,fc2eab51d27faa6099c26e63f51e55f5aa1d474c
c7523 57480428a9a33d98a3b6f96c851399f95bf06a845d43e75659c302d2f3197b6]
time dist is:3.652665[s]
sign menu
to press 0 auth sign
to press 1 to edit mode
to press 2 to delete  mode
to press 3 to watch mode
 to press 4 is attack mode
 to press 5 is remake mode
```

図 5:署名作成プログラム

	A	B	C	D	E
1	髪･ｴ･ｽｲ蜷	19c483ad5	cc00233f879fd3e19950574f41	0	
2	Miku Hatsu	[bdbcd444	631960026534c0f51eec556569d		
3	センター	[16a5b536	d6cf70750cd513ca64d52cec165		
4	上半身	[2224e2a7	a467d98324e6432ab782ddd70f6		
5	首	[1a940165	12a72d8acfb82a2239874f2dd7fb		
6	頭	[106b738d	b43f8db26e7d83950227bc711e1		
7	左目	[ef9967e8	35fe20c06da861e09d1403688a9c		
8	右目	[fb86ab49	d5f4ccccfa18435ec0f4abd6d622		
9	ﾈｸﾀｲ1	[207b73e1	a33ec1539313ba974fc88e4b0a6		
10	ﾈｸﾀｲ2	[227341d1	6e47ae3e620ebf164a094639369		
11	ﾈｸﾀｲ3	[1d8a97c8	12e10e3fec10a3ee74e1aa2a3e2		
12	下半身	[90047019	193e26e76a99d6f304023d23223		
13	腰飾り	[14a37a6d	b31205eae40717db48a579d4e15		
14	左髪1	[11c44fd8	15f9e20301e0cf3e0b4ddf41da71		
15	左髪2	[17d97fa8	124eb18f9632f8082ebcc73bc44		
16	左髪3	[1b0b2847	39d3f5b52753d1e582659cee861		

図 6:作成した署名ファイルを excel で閲覧した画面

3.3.3 検証プログラム

検証用のプログラムは、検証者および編集者、不正編集者が行うことができる操作をまとめて行うことができる。

3.3.3.1:署名の検証

3.3.2 で作成された署名と、3.3.1 で作成された署名対象データを用いて先述してある署名検証方式に即して検証を行う。

3.3.3.2:署名の編集

3.3.2 で作成した署名に対して、コンテンツの更新とコンテンツの削除を行うことができる。コンテンツの更新とは、あるコンテンツを削除して、同じコンテンツを挿入する、制限された変更操作として扱う。

また、コンテンツの削除とは、あるコンテンツをコンテンツ全体の関係から削除することを示す。

編集を行ったあとには、編集前の署名から編集後の署名へと署名の更新を行う。

3.3.3.3:コンテンツに付加された権利の変更

コンテンツに対して、変更、削除の可否を設定することができる。この時、モデル単位で権限の変更を行う。

編集、削除不可能の状態から可能の状態へ戻すことはできず、可能な状態から不能な状態へ戻すことのみ許可される。

この操作を行ったあと、3.3.3.2 で示された署名の編集を行うことで正当な署名が正当な署名でなくなることが確認でき、不正な編集を検出したと言えるようになる。

この変更操作を行う様子を図 7 に示す。

```
to press 0 auth sign
to press 1 to edit mode
to press 2 to delete  mode
to press 3 to watch mode
 to press 4 is attack mode
 to press 5 is remake mode1
choose change model no:1
now model 1 isdeletable
 do you want to change this model undeletable?
0 is undeletable 1 is deletable
command?:0
model no 1 is changed undeletable
sign menu
to press 0 auth sign
to press 1 to edit mode
to press 2 to delete  mode
to press 3 to watch mode
 to press 4 is attack mode
 to press 5 is remake mode
```

図 7:署名の変更操作プログラム

3.3.3.4:コンテンツに含まれる部分コンテンツの確認

.pmm ファイルから作成された署名より、使用された 3D モデルの名称および変更、削除の現在の権限の確認を図 8 のように行うことができる。

```
to press 0 auth sign
to press 1 to edit mode
to press 2 to delete  mode
to press 3 to watch mode
 to press 4 is attack mode
 to press 5 is remake mode3
model no:0   model name:Miku Hatsune
model changeable:0 model startid:1 model finish id:123
model no:1   model name:Luka Megurine
model changeable:1 model startid:124 model finish id:312
model no:2   model name:Rin Kagamine
model changeable:0 model startid:313 model finish id:415
model no:3   model name:Len Kagamine
model changeable:1 model startid:416 model finish id:520
model no:4   model name:KAITO
model changeable:0 model startid:521 model finish id:627
```

図 8:コンテンツに使用されているモデルの詳細確認

4.　シミュレーション方法および評価

4.1 シミュレーションの概要

実装したシステムの動作確認及び評価のために、以下に示す環境でシミュレーションを行い、署名対象の.pmm データで、3D モデル数及び、各モデルが有する部分コンテンツ数の合計に応じて処理時間がどれだけ変動するかを複数回計測して、処理ごとにその平均時間を取り、結果として算出した。

4.2 シミュレーション環境

シミュレーションは 1 台のコンピュータで Oracle VirtualBox を動作させ、仮想 OS 上で動作を評価した。その詳細を以下の表 2 に示す。

表 2:シミュレーションに用いたシステム環境

主マシン	詳細
OS	Windows 7 home premium SP1 64bit
CPU	Intel® Core™ i5-2450M CPU 2.5GHz
メモリ	8.00GB

VirtualBox	バージョン 5.0.26
仮想 OS	ubuntu16.04LTS
CPU	主マシンと同じ
メモリ	3.00GB

また、コンピュータ上で暗号計算に用いるペアリングと呼ばれる関数の計算や楕円曲線上の点の演算や、有限体の元の演算を行うためのライブラリとして TEPLA[6]を用いている。こちらは、2015 年 12 月 20 日に公開されたバージョン 2.0(2016 年 8 月 12 日現在の最新版)を用いた。

これらの仮想 OS の環境は現行のコンピュータに比較してスペック面で制限されているが、この仮想 OS 上で実用に耐えうる結果が出ると確認できれば、実際に現行のコンピュータ、ネットワーク環境上でも処理速度面での支障は及びにくいと考えられる。

4.3 シミュレーションで用いたプログラム

シミュレーションは、gcc（バージョン 5.4.0）を用いて C 言語で作製したプログラムをコンパイル、ubuntu の端末上で実行することで行った。

また、署名対象となる MikuMikuDance のバージョン及び、今回のシミュレーションで用いた、MikuMikuDance 標準付属の 3D モデル[†3]を以下の表 3 および 4 に示す。なお、集約署名を作成する際に署名対象として、以下の表 5 のように対応関係を定義した。

表 3:使用した MikuMikuDance のバージョン

MikuMikuDance 本体	バージョン 9.26

表 4:署名対象とした 3D モデルおよび部分コンテンツ数

モデル名	部分コンテンツ数
初音ミク	122
巡音ルカ	188
鏡音リン	102
鏡音レン	104
KAITO	106

表 5:コンテンツの分類とコンテンツが有する署名対象

	構成する部分コンテンツ	署名対象
.pmm ファイル	3D モデル	特定の文字列
3D モデル	ボーン	モデルの名称
ボーン	なし	ボーンの名称

この表 5 での署名対象とは、署名を作成する際に h を構成するためにハッシュ関数を取り出す、メッセージ m を示している。

4.4 シミュレーションの方法

4.4.1:署名の作成

署名作成プログラムを用いて、モデルを 1~5 種類用いた.pmm ファイルより抽出した署名対象ファイルより、集約署名をそれぞれ作成して、その所要時間を計測する。

4.4.2:署名の検証(正当なコンテンツ)

正当な集約署名を作成し、この署名を署名対象ファイルを用いて検証する。集約署名でモデルを集約した数ごとにその所要時間を計測する。

4.4.3:署名の検証(不正なコンテンツ)

正当な集約署名を作成し、その署名に対して署名対象データを改ざんして検証を行い、検証に失敗することを確認する。その所要時間を計測する。

4.4.4:署名の編集

正当な集約署名を作成し、その署名に対して使用した署名対象のうち､編集が許可されている署名対象 50 か所を書き換えた後、更新することで正当な集約署名として編集制御を行えているかを確認し、その操作に要した合計の時間を計測する。

4.5 シミュレーションの結果

4.5.1 署名作成の結果

署名作成にかかった時間および 1 部分コンテンツあたりの所要時間を署名対象のモデル数、コンテンツ総数ごとに 10 回測定し、その平均値を以下の表 6 に示す。

表 6:署名作成に要した時間

モデル数	署名対象総数	所要時間[ms]	署名対象毎の所要時間[ms]
1	124	1311	10.57
2	313	3374	10.77
3	416	4480	10.77
4	521	5679	10.90
5	628	6956	11.08

なお、署名対象となるコンテンツには、モデルそのものと動画そのものの集約署名も個数に含まれている。

†3 「初音ミク、巡音ルカ、鏡音リン、鏡音レン、KAITO」はクリプトン・フューチャー・メディア株式会社の著作物です。[7]

4.5.2:署名の検証(正当なコンテンツ)の結果

署名の検証にかかった時間および 1 部分コンテンツあたりの所要時間を署名対象のモデル数、コンテンツ総数ごとに 10 回測定し、その平均値を以下の表 7 に示す。

表 7:正当な署名検証に要した時間

モデル数	署名対象総数	所要時間[ms]	署名対象毎の所要時間[ms]
1	124	1088	8.77
2	313	2863	9.15
3	416	3836	9.22
4	521	4863	9.33
5	628	5703	9.08

なお、署名対象となるコンテンツには、モデルそのものと動画そのものの集約署名も個数に含まれている。

4.5.3:署名の検証(不正なコンテンツ)の結果

署名作成にかかった時間および 1 部分コンテンツあたりの所要時間を署名対象のモデル数、コンテンツ総数ごとに 10 回測定し、その平均値を以下の表 8 に示す。

表 8:不正な署名検証に要した時間

モデル数	署名対象総数	所要時間[ms]	署名対象毎の所要時間[ms]
1	124	1109	8.94
2	313	2862	9.14
3	416	3786	9.10
4	521	4709	9.04
5	628	5670	9.03

なお、署名対象となるコンテンツには、モデルそのものと動画そのものの集約署名も個数に含まれている。

4.5.4:署名の編集の結果

署名の更新にかかった時間および 1 部分コンテンツあたりの所要時間を署名対象のモデル数、コンテンツ総数ごとに 10 回測定し、その平均値を以下の表 9 に示す。

表 9:署名の編集および更新に要した時間

モデル数	署名対象総数	所要時間[ms]	署名対象毎の所要時間[ms]
1	124	1288	10.39
2	313	3312	10.58
3	416	4369	10.50
4	521	5388	10.34
5	628	6493	10.34

なお、署名対象となるコンテンツには、モデルそのものと動画そのものの集約署名も個数に含まれている。

5. まとめ

本稿では、編集制御に対応した署名方式にて、3D ムービーにて動いている 3D モデルのパーツを署名対象として署名を作成、検証するプログラムを作成した。

署名作成に必要な時間は署名を集約するコンテンツの数におおよそ比例し、署名検証に必要な時間も署名を集約する対象のコンテンツの数におおよそ比例して、1 秒間におよそ 100 個のコンテンツの署名に関する各種計算ができることが確認できた。

署名を集約するコンテンツの数は、本稿での署名作成方式だと 3D モデルを用いた署名においては 3D モデル 1 体につきおよそ 100 程度に分割されているが、署名集約対象がもっと細分化されて、ボーンの名称の情報だけではなく、時間に応じた動きなどを署名対象に含め、コンテンツを構成する要素を盛り込めると考えられる。

これらの要素を盛り込み、その上で現状では.pmm ファイルから署名用ファイルを経由して署名を作成しているところを.pmm ファイルから直接署名を作成できるようにすることが今後の課題として挙げられる。

6. 謝辞

今回の研究においてご指導いただいた姜　玄浩助教に心より感謝いたします。また、研究のサポートをしていただいた岩村研究室のメンバーの皆さまにもこの場を借りて非常に感謝いたします。

参考文献

[1] “Youtube” https://www.youtube.com/ (参照 2016-08-11)
[2] “ニコニコ動画” www.nicovideo.jp/, (参照 2016-08-11).
[3] 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　Vol53 No9
[4] 岩村惠市　稲村勝樹　古賀克磨　金田北洋　「権利継承と編集制御を同時に実現するコンテンツ制御システム」
[5] 樋口優「VPVP」http://www.geocities.jp/higuchuu4/(閲覧日 2016-08-12)
[6] 筑波大学　暗号・情報セキュリティ研究室　「TEPLA」http://www.cipher.risk.tsukuba.ac.jp/tepla/index.html(閲覧日 2016-08-12)
[7]クリプトン・フューチャー・メディア　「キャラクター利用のガイドライン http://piapro.jp/license/character_guideline#example_e（閲覧日 2016-08-12）