コンテンツ二次利用に適した編集制御可能な電子署名システム

佐野達彦† 柿崎淑郎† 稲村勝樹‡ 岩村惠市†

†東京理科大学 102-0073 東京都千代田区九段北 1‐14‐6
sano@sec.ee.kagu.tus.ac.jp
‡株式会社 KDDI 研究所 356-8502 埼玉県上福岡市大原 2-1-15

あらまし 本稿では、コンテンツ循環を想定した上で、著作権の保護と継承を実現するコンテンツ二次利用に適した編集制御可能な電子署名システムを提案・実装する。コンテンツの編集制御とは、二次著作者に対し著作者が許可する編集の内容と範囲を事前に設定することである。本方式は、著作者がコンテンツ作成後にコンテンツを分割し、部分コンテンツ毎に個別署名とAggregate署名を設定することで、部分コンテンツの変更と削除の制御を実現する。また、分割したコンテンツの間に空データを設定することで、二次著作者はデータの変更・削除だけでなく、新しいデータの追加を行うことを可能とする。

E-signature system that can control the edit for the contents

Tatsuhiko Sano† Yoshio Kakizaki† Masaki Inamura‡ Keiichi Iwamura†

†Tokyo University of Science 1-14-6 Kudankita, Chiyoda-Ku, Tokyo 102-0073, JAPAN
sano@sec.ee.kagu.tus.ac.jp

‡KDDI R&D Laboratories Inc. 2-1-15 Ohara,Kamihukuoka-shi,Saitama,356-8502 Japan

Abstract In this paper, we propose a signature system for the contents circulation that can control the edit. Our method divides an original content, and sets the individual signature of each partial contents and the aggregate signature. The individual signatures represent the state of control, and secondary author can edit the content within the permitted range of the state. Moreover, we set empty data between divided contents, thus achieve the control of not only the change and the deletion of partial contents, but also the addition of new data.

1 はじめに

近年、YouTube やニコニコ動画に代表される動画共有サイトが流行している。そこでは一般利用者がコンテンツを作成・投稿共有するだけでなく、すでに公開されているコンテンツに編集を施して新たなコンテンツを作成、流通させていくといったコンテンツ循環が展開されている。現在の著作権保護技術としては視聴制御や、コピー制御が一般的であるが、コンテンツ循環の市場には不十分であり、コンテンツの編集を考慮した、編集制御という新たな著作権保護技術が必要である。

編集制御とは、著作者がコンテンツの作成後に編集を制御したい領域ごとに分割し、各領域に編集可否を事前に設定・公開し、コンテンツの利用者は著作者が編集を許可した範囲内でのみ編集を行うことを可能とする。また、コンテンツの検証者は編集されたコンテンツに対して、著作者が設定した制御状態を守った編集がなされているかどうか検証することが必要である。編集者はコンテンツに対して自由に変

更、削除、追加の編集を行うことができるが、著作者の設定した変更、削除、追加の可能・禁止の制御状態を違反する編集が行われた場合、編集後のコンテンツは利用することができなくなる。

本稿では、上述した機能を実現するために、個別署名と Aggregate 署名を設定することで、部分コンテンツの変更、削除、追加を制御する方式を提案する。

2 コンテンツ二次利用に適した編集制御可能な電子署名システム

2.1 コンテンツ二次利用に適した編集制御可能な電子署名システム

本方式は循環型コンテンツを想定しており、一般ユーザがコンテンツを作成または既存コンテンツを独自に編集して新たなコンテンツとして創出・流通させていく状況を考える。一般に、著作者がネット上に公開したコンテンツに対して、二次著作物への利用はある程度許可するが、その二次著作物が自分の意図に反する利用がされたときそれを禁止したり、利用されたコンテンツは自分の著作物であることを主張したいという要求があると考えられる。このような場合、従来のコンテンツの不正利用を防止する視聴制御やコピー制御では不十分であり、コンテンツが編集されてもそれが確認できる仕組みが必要となる。

このような仕組みに対して、泉らの墨塗り署名方式[2]や伊豆らのPIAT署名方式[3]がある。しかし、墨塗りではない部分任意変更や新たなデータの追加を含む事前編集制御はできていない。そこで本稿では、従来の墨塗り方式を拡張して、変更・削除・追加の事前制御を実現する電子署名システムを提案する。

2.2 エンティティ

本方式ではディジタルコンテンツ全般における署名システムの提案として、以下の役割を持つ3エンティティを想定する。

- 著作者（作成者）
  コンテンツ作成及び編集許諾制御設定
- 編集者
  編集権限を持ったコンテンツ編集
- 検証者
  正当な署名のあるコンテンツか検証

著作者は自らの秘密鍵と公開鍵を持ち、コンテンツに署名を施し著作権を保持すると同時に部分分割されたどのデータを編集（変更・削除・追加）してよいかの制御を事前に設定する。編集権限を持つ編集者も自らの秘密鍵と公開鍵を持ち、許可された範囲内でデータを編集する。検証者は署名検証に必要な著作者と編集者の公開鍵を持ち検証する。検証者の機能はコンテンツ再生機器などに持たせることにより、正当な署名を持たないコンテンツは再生できないシステムを構築する。

2.3 部分データの制御状態

本方式では著作者によるコンテンツの部分データの編集制御を実現するために以下の状態を設定する。ただし、部分データとしては既存データと後述する空データの二種類を考える。既存データとは実在する部分データであり，再生器が再生対象とするコンテンツの部分を構成する。既存データは下記(a)～(e) の状態を持つ。一方、空データは再生器が再生対象とはしないデータであり、データの追加を制御するための制御データと考えることができる。空データは下記(f)、(g)の状態を持つ。追加可能状態の空データは編集者による追加データと置き換え可能であり、追加されたデータは以降既存データとなり、(a)～(e)の状態が定められる。各部分データの制御状態の概略を図に示す。

(a) 変更可能・削除可能(CADA)
(b) 変更可能・削除禁止(CADP)
(c) 変更禁止・削除可能(CPDA)
(d) 変更禁止・削除禁止(CPDP)
(e) 削除状態(D)
(f) 追加可能(AA)
(g) 追加禁止(AP)

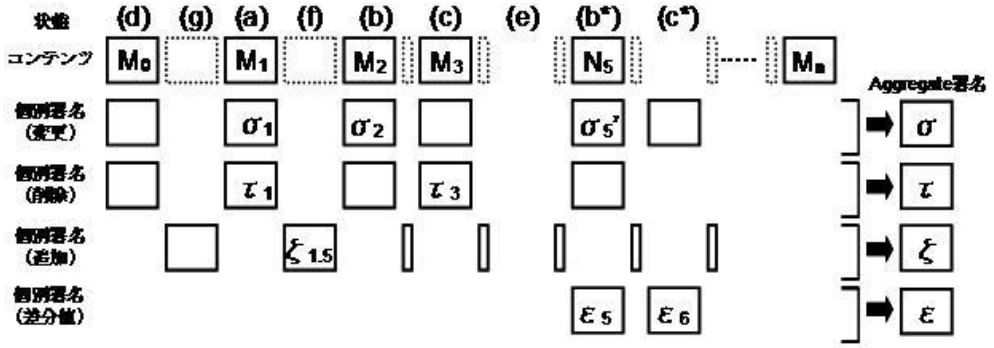

図 1：制御状態の概略

制御状態は、各部分データにおいて、既存データでは 2 種類、空データでは 1 種類の署名を用いることで変更、削除、追加の制御を実現している。著作者は既存データに対して、著作者の秘密鍵を用いて変更用個別署名σ_i、削除用個別署名τ_iを作成し、それを乗算して変更用 Aggregate 署名σ、削除用 Aggregate 署名τに代入している。空データに対しては、著作者の秘密鍵を用いて追加用個別署名ξ_iを作成し、それを乗算して追加用 Aggregate 署名ξに代入している。さらに、著作者は変更、削除、追加に対応する署名集合$\mathbb{G}_c$、$\mathbb{G}_d$、$\mathbb{G}_a$を用意し、編集を許可する部分データの個別署名だけを各集合に出力する。編集を許可されていない部分データの個別署名は出力しない。

2.4 変更制御

編集者が(a)の状態の部分データを変更する場合、署名集合$\mathbb{G}_c$にある個別署名で Aggregate 署名 σ を除算し、変更データに対する個別署名を著作者が行ったのと同様の手順で作成し、それを Aggregate 署名 σ に乗算する。ただし、署名作成時には編集者の秘密鍵を用いる。編集者は署名集合$\mathbb{G}_c$から前の個別署名を削除し、変更データの個別署名を加える。署名集合$\mathbb{G}_d$についても同様に行う。

次に編集者が(b)の状態の部分データを変更する場合、署名集合$\mathbb{G}_c$については(a)の状態の部分データと同様のことを行う。ただし、署名集合$\mathbb{G}_d$については個別署名が出力されていない。そこで、元の部分データと変更データからハッシュ値を作成し、その差分値を署名することでデータの正当性を示している。この様子を図 2 に示す。これによって、変更後の部分データでも元の部分データのハッシュ値を復元し、Aggregate 署名の検証に用いることができる。また、ハッシュ差分値は検証のための署名集合$\mathbb{G}_{bc}$に保持し、 (b*)の制御状態になる。

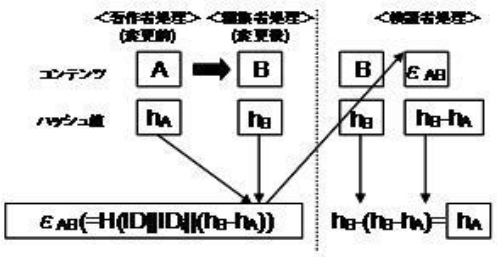

図 2：ハッシュ差分値の仕組み

2.5 削除制御

編集者が(a)の状態の部分データを削除する場合、署名集合$\mathbb{G}_c$にある個別署名で Aggregate 署名 σ を除算し、署名集合$\mathbb{G}_c$から前の個別署名を削除する。署名集合$\mathbb{G}_d$についても同様に行う。

次に編集者が(c)の状態の部分データを削除する場合、署名集合$\mathbb{G}_d$については(a)の状態の部分データと同様のことを行う。ただし、署名集合$\mathbb{G}_c$については個別署名が出力されていない。そこで、(b)の変更制御と同様にハッシュ差分値を用いる。このとき、削除制御では部分データを削除するので図 2 の部分データ B は 0 と考えることができる。また、ハッシュ差分値は検証のための署名集合$\mathbb{G}_{bc}$に保持し、 (c*)の制御状態になる。

2.6 追加制御

編集者が(f)の状態の空データに追加する場合、署名集合$\mathbb{G}_a$にある個別署名で Aggregate 署名ξを除算し、その個別署名を削除する。また、編集者の秘密鍵を用いて追加データに対する新たな変更用個別署名と削除用個別署名を作成し、Aggregate 署名 σ、τ に乗算する。ここで、追加データに対する変更、削除制御を許可する場合、各個別署名を署名集合$\mathbb{G}_c$、$\mathbb{G}_d$に加える。

2.7 提案方式アルゴリズム

鍵生成

著作者の秘密鍵、公開鍵ペア(sk0,pk0)∈Z/pZ ×G_2 生成(群 G_2 の生成元 g_2、位数 p、pk0 ← g_2^{sk0})。

編集者の秘密鍵、公開鍵ペア(sk1,pk1)∈Z/pZ ×G_2 生成(群 G_2 の生成元 g_2、位数 p、pk1 ← g_2^{sk1})。

署名生成

① オリジナルコンテンツM を編集制御する領域ごとに n 分割。

$M \rightarrow M_1, M_2, \ldots, M_n$

② コンテンツ識別子 ID、部分コンテンツ識別子 ID_i を生成し、以下のように設定(i = 1,…, n)。

$M_0^* \leftarrow ID$, $M_i^* \leftarrow ID \| ID_i \| M_i$

③ 分割されたデータ間に空データ $B_{i.5}$ が存在すると仮定し、既定のダミーデータ d を用いて以下の様に設定(i = 0,…, n)。

$B_{i.5}^* \leftarrow ID \| ID_{i.5} \| d$

④ 各既存データに状態(a)～(e)、空データに状態(f)～(g) のうちの一つを定める。

⑤ M_0^*, M_i^* ,$B_{i.5}^*$よりハッシュ値を生成(i = 0,…, n)。

$h_i \leftarrow H(M_i^*)$, $h_{i.5} \leftarrow H(B_{i.5}^*)$

⑥ 変更・削除・追加制御用の個別署名生成(i = 0,…, n)。

変更用：$\sigma_i \leftarrow H(ID \| ID_i \| h_i \| 0^c)^{sk0}$

削除用：$\tau_i \leftarrow H(ID \| ID_i \| h_i \| 1^c)^{sk0}$

追加用：$\xi_{i.5} \leftarrow H(ID \| ID_{i.5} \| h_{i.5} \| 0^c)^{sk0}$

⑦ Aggregate 署名 σ、τ、ξ を生成し、変更、削除、追加を許可する部分データの個別署名を各々$\mathbb{G}_c$、$\mathbb{G}_d$、$\mathbb{G}_a$へ出力。

$$\sigma \leftarrow \Pi_{i=0}^{n}\sigma_i,\ \tau \leftarrow \Pi_{i=0}^{n}\tau_i,\ \xi \leftarrow \Pi_{i=0}^{n}\xi_{i.5}$$

⑧ コンテンツ $M^* = \{M_0^*, M_1^*, \ldots, M_n^*\}$、編集許可データ個別署名集合$\mathbb{G}_c$、$\mathbb{G}_d$、$\mathbb{G}_a$、Aggregate 署名 σ、τ、ξ を出力。ただし、M_0^*は変更・削除禁止。

データ更新

以下の処理を(i),… ,(v) として定義する.

(i) $M_i^* \leftarrow N_i^*$, $h'_i \leftarrow H(N_i^*)$, σ/σ_i , $\mathbb{G}_c \setminus \{\sigma_i\}$, $\sigma'_i \leftarrow H(ID \| ID_i \| h'_i \| 0^c)^{sk1}$, $\sigma \leftarrow \sigma \times \sigma'_i$

(ii)τ/τ_i, $\mathbb{G}_d \setminus \{\tau_i\}$, $\tau'_i \leftarrow H(ID \| ID_i \| h'_i \| 1^c)^{sk1}$, $\tau \leftarrow \tau \times \tau'_i$

(iii) $\xi_i \leftarrow H(ID \| ID_i \| (h_i - h'_i) \| 1^c)^{sk1}$

(iv) $B_{i.5}^* \leftarrow N_{i.j}^*$, $h'_{i.j} \leftarrow H(N_{i.j}^*)$, $\xi/\xi_{i.5}$, $\mathbb{G}_a \setminus \{\xi_{i.5}\}$, $\sigma'_{i.j} \leftarrow H(ID \| ID_{i.j} \| h'_{i.j} \| 0^c)^{sk1}$, $\sigma \leftarrow \sigma \times \sigma'_{i.j}$

(v) $\tau'_{i.j} \leftarrow H(ID \| ID_{i.j} \| h'_{i.j} \| 1^c)^{sk1}$, $\tau \leftarrow \tau \times \tau'_{i.j}$

① M^*から編集する部分データ M_i^*または $B_{i.5}^*$を決定。

② 編集の種類によって以下の処理を行う。ただし、元の既存データを M_i^*, 変更データを N_i^*, 空データを $B_{i.5}^*$, 追加データを $N_{i.j}^*$とする(j は複数可能)。

・(a)→(a) の場合：

(i), $\mathbb{G}_c \leftarrow \mathbb{G}_c \cup \sigma'_i$, (ii), $\mathbb{G}_d \leftarrow \mathbb{G}_d \cup \tau'_i$

・(a)→(b) の場合：状態のみ変える場合$\mathbb{G}_d \setminus \{\tau_i\}$.

(i), $\mathbb{G}_c \leftarrow \mathbb{G}_c \cup \sigma'_i$, (ii)

・(a)→(c) の場合：状態のみ変える場合$\mathbb{G}_c \setminus \{\sigma_i\}$.

(i), (ii), $\mathbb{G}_d \leftarrow \mathbb{G}_d \cup \tau'_i$

・(a)→(d) の場合：状態のみ変える場合$\mathbb{G}_d \setminus \{\tau_i\}$,$\mathbb{G}_c \setminus \{\sigma_i\}$.

(i), (ii)

・(a)→(e) の場合：

$M^* \setminus M_i^*$, σ/σ_i, $\mathbb{G}_c \setminus \{\sigma_i\}$, τ/τ_i, $\mathbb{G}_d \setminus \{\tau_i\}$

・(b)→(b) の場合：

(i), $\mathbb{G}_c \leftarrow \mathbb{G}_c \cup \sigma'_i$, (iii), $\mathbb{G}_{bc} \leftarrow \mathbb{G}_{bc} \cup (h_i - h'_i)$

・(b)→(d) の場合：状態のみ変える場合$\mathbb{G}_c \setminus \{\sigma_i\}$.

(i), (iii), $\mathbb{G}_{bd} \leftarrow \mathbb{G}_{bd} \cup (h_i - h'_i)$

・(c)→(d) の場合：

$\mathbb{G}_d \setminus \{\tau_i\}$

・(c)→(e) の場合：($h'_i = 0$).

M^*＼M_i^*, (iii), $\mathbb{G}_{bc} \leftarrow \mathbb{G}_{bc} \cup (h_i)$, τ/τ_i, $\mathbb{G}_d$＼$\{\tau_i\}$

・(f)→(a) の場合：

(iv), $\mathbb{G}_c \leftarrow \mathbb{G}_c \cup \sigma'_{i,j}$, (v), $\mathbb{G}_d \leftarrow \mathbb{G}_d \cup \tau'_{i,j}$

・(f)→(b) の場合：

(iv), $\mathbb{G}_c \leftarrow \mathbb{G}_c \cup \sigma'_{i,j}$, (v)

・(f)→(c) の場合：

(iv), (v), $\mathbb{G}_d \leftarrow \mathbb{G}_d \cup \tau'_{i,j}$

・(f)→(d) の場合：

(iv), (v)

・(f)→(g) の場合：

$\mathbb{G}_a$＼$\{\xi_{i.5}\}$

③ 計算された ζ_i を Aggregate し、編集履歴データ r とその署名 σ_r を生成する。

$\varepsilon \leftarrow \Pi\varepsilon_i$, r, σ_r

④ 更新済コンテンツ、個別署名集合$\mathbb{G}_c$、$\mathbb{G}_d$、$\mathbb{G}_a$、Aggregate 署名 σ、τ、ξ、ε、ハッシュ差分値の集合$\mathbb{G}_{bd}$, $\mathbb{G}_{bc}$、編集履歴データ r、σ_r を出力。

検証

① 各部分データのコンテンツ識別子 ID が M_0^*に等しいか検証。

② データが追加されていない位置の空データ $B_{i.5}^*$から以下を生成し検証。

$z_{i.5} \leftarrow H(ID \parallel ID_{i.5} \parallel H(B_{i.5}^*) \parallel 0^c)$

$z \leftarrow \Pi z_{i.5}$, $e(\xi, g_2) = e(z, pk0)$

③ 検証が正しければ、追加データも既存データとし、全既存データ M_i^*の部分コンテンツ識別子 ID_i が昇順になっているか検証。

④ $\mathbb{G}_{bd}$にハッシュ値差分をもつ既存データが存在することと、$\mathbb{G}_{bc}$ にハッシュ値差分をもつ既存データが存在しないことを確認し、正しければ$\mathbb{G}_{bd}$, $\mathbb{G}_{bc}$内のハッシュ値差分とεを用いて署名検証

$w_i \leftarrow H(ID \parallel ID_i \parallel (h_i - h'_i) \parallel 1^c)$

$w \leftarrow \Pi w_i$, $e(\xi, g_2) = e(w, pk1)$

⑤ 検証が正しければ、その既存データのハッシュ値 $h'_i = H(M_i^*)$ を生成し、元データに対するハッシュ値を算出。

$h'_i + (h_i - h'_i) = h_i$

⑥ $\mathbb{G}_{bd}$にハッシュ値差分がある場合は削除検証、$\mathbb{G}_{bc}$ にハッシュ値差分がある場合は変更検証用に前記ハッシュ値を用い, それ以外の検証に対しては既存データのハッシュ値を用いて以下を生成。

変更用：$x_i \leftarrow H(ID \parallel ID_i \parallel h_i \parallel 0^c)$、

削除用：$y_i \leftarrow H(ID \parallel ID_i \parallel h_i \parallel 1^c)$

⑦ 編集履歴データ r から以下を作成し、検証

著作者用：$X_0 \leftarrow \Pi x_i$, $Y_0 \leftarrow \Pi y_i$,

編集者用：$X_1 \leftarrow \Pi x_i$, $Y_1 \leftarrow \Pi y_i$,

$e(\sigma, g_2) = \Pi_{i=0}^{1} e(X_i, pki)$, $e(\tau, g_2) = \Pi_{i=0}^{1} e(Y_i, pki)$

3 考察

3.1 違反検証

前章のアルゴリズムに対する禁止処理の違反検証を検証アルゴリズムの手順に合わせ検討する。まず、追加データは ID が元々の既存データと異なる($ID_{i,j}$)ため識別できる。よって、検証の②において追加データのない位置における空データのハッシュ値 $z_{i.5}$ を作成する。ここで考えられる違反は(g)状態の空データへの追加であるが、その場合その空データに対するハッシュ値は②で計算されない。しかし、ξ にはその部分の空データに対する個別署名が含まれており、それが$\mathbb{G}_a$になく除算できないため、Aggregate 署名が整合しない。よって、追加に対する違反が検証できる。

次に、既存データの(b),(d) 状態におけるデータ削除と(c),(d) 状態におけるデータ変更に関する違反を考える。まず、(b) 状態にある既存データを削除する場合を考える。この場合、$\mathbb{G}_c$にある個別署名で σ を除算して、それを$\mathbb{G}_c$から削除し、さらに ε_i を作って ε に Aggregate し、$\mathbb{G}_{bd}$にそのハッシュ値差分を入れれば整合性が保たれる。しかし、正当な(b) における変更処理の場合、(b) は変更のみ許可であるので、必ず部分データは存在する。すなわち、$\mathbb{G}_{bd}$にハッシュ値差分をもつ部分データは必ず存在するが、違反の場合存在しない。よって、検証の④における既存データとハッシュ値差分の組合せの確認により違反が検証される。もし，ハッシュ値差分を$\mathbb{G}_{bc}$に入れれば⑤、⑥において正しいハッシュ値が復元されず Aggregate 署名が整合しない。逆に、(c) においてデータを

変更する場合、$\mathbb{G}_d$にある個別署名でτを除算して、新たな署名と入れ替え、さらにεiを作ってεにAggregate し、$\mathbb{G}_{bc}$ にそのハッシュ値差分を入れれば整合性が保たれる。しかし、この場合も正当な(c) における削除処理では部分データは削除され存在しないのに対して、部分データが存在することになる。よって、これも④における既存データとハッシュ値差分の組合せの確認により違反が検証される。また、(d) の部分データを変更または削除すると、ハッシュ値差分を$\mathbb{G}_{bd}$，$\mathbb{G}_{bc}$のどちらにも持たせることによって整合性がとれる。しかし、変更に対しては$\mathbb{G}_{bd}$のハッシュ値差分と既存データの組合せ，削除に対しては$\mathbb{G}_{bc}$のハッシュ値差分と既存データの組合せが上記と同様になるので、④によって検証される。よって、禁止に対しても提案アルゴリズムは有効に動作していると言える。

以上に問題がなければ、⑦において全既存データの Aggregate 署名を検証することにより全体の整合性を検証できる。

4 実装・評価

4.1 実装環境

仮想マシン VMware Player version 7.0.0 build-203739 上の仮想 OS Ubuntu 8.04 Japanese を用いて実装を行い、使用言語は C 言語を用いる。また、実装は MPEG-1 ファイルを対象として行った。ページの都合上、詳細は省略し、デモを示す。

4.2 署名・検証回数

従来の署名方式を含めて、変更・削除・追加における事前制御の可否、各署名方式における事前署名数、検証回数を比較し、表 1 に示す。なお、○は事前制御が可能、△は事前制御ではなく不正更新の検知が可能、×はどちらも満たさない場合とする。

表 1 より、提案方式のみが 3 つの状態を事前制御可能である。ただし、提案方式は変更・削除・追加のすべての制御においてそれぞれの署名を生成しているため、他の方式に比べ署名回数や検証回数は多い。特に、著作者は分割データ数の約 3 倍の署名が必要になるが、これはコンテンツ作成時に一度だけ事前にやることであるので、実用上はほとんど問題ないと考えられる。

	変更	削除	追加	事前署名回数	検証回数
提案方式	○	○	○	3n+1	4
墨塗り署名	△	○	×	2n	2
PIAT方式	△	△	△	1	1

表 1:各署名方式の署名回数と検証回数

5 まとめ

本稿では、コンテンツの二次利用に適した編集制御可能な電子署名システムについて提案・実装を行った。本実装プログラムは、実用性を考えて動画ファイルに対して事前に制御状態の定義ができるようになっているが、プログラムの拡張子を変更すれば画像や音楽にも応用できる。しかし、動画内の音声と画像といった素材ごとに分けることなどには対応していない。今後は、n 次編集に対する制御状態の再定義・継承について検討し、よりコンテンツ循環に適したものを目指す。

6 謝辞

本研究を進めるにあたり、ご指導を頂いた岩村教授に感謝致します。また、本実装において協力いただいた柿崎助教授、北川氏に感謝致します。

7 参考文献

[1] 岩村 惠市、斎藤 旭、“コンテンツ編集を事前制御可能な電子署名システム”、電子情報通信学会論文、2009

[2] 泉雅巳、伊豆哲也、國廣昇、太田和夫、“墨塗り・削除署名の拡張”、2007-CSEC-38, pp.355-361, 2007.

[3] T. Izu, N. Kunihiro, K. Ohta, M. Takenaka, T. Yoshioka,“A Sanitizable Signature Scheme with Aggregation,” ISPEC 2007, LNCS 4464, pp.51-64, 2007.