2007－CSEC－39 (9)
2007／12／14

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

齊藤 旭　　山田 裕也　　岩村 惠市

東京理科大学　〒102-0073　東京都千代田区九段北 1-14-1
E-mail：iwamura@ee.kagu.tus.ac.jp

あらまし　ディジタルコンテンツに対する改ざんの防止・著作権の保護を実現する技術として電子署名がある。その応用技術として、コンテンツの一部を非開示にしても真正性確保とプライバシー保護の両立が可能な墨塗り署名がある。しかし、従来技術は行政文書に対する一部削除や墨塗りを目的としたものが多く、それらに加えてディジタルコンテンツの更新や追加までを含む著作権保護への応用についてはあまり考察されていなかった。そこで本稿は編集可能コンテンツに対してハッシュ値の差分値やAggregate署名を用いてディジタルコンテンツの更新・追加までを可能にする電子署名システムを提案する。

キーワード　コンテンツ，編集，電子署名，墨塗り署名，ハッシュ，Aggregate署名

The Proposal of E-signature System using Sanitizable Signature for Editional Contents

Akira SAITO　　Yuya YAMADA　　Keiichi IWAMURA

Tokyo University of Science　1-14-1 Kudan-Kita, Chiyoda-Ku, Tokyo 102-0073, Japan
E-mail：iwamura@ee.kagu.tus.ac.jp

Abstract　There is digital signature as a technology that can prevent falsification and can protect the copyright to digital contents. As applied technology, there is the Sanitising signature that coexistence of the genuine security and privacy protection is possible with the part of contents as non-disclosure. However, the technology was aimed for deletion and sumi coating for the administration document conventionally, and it was not considered update and the addition of the application for the digital contents that needed copyright protection very much. Therefore I suggest the e-signature system enabling update and addition which used for the aggregate signature and the difference of the hash value for editional contents.

Keyword　Contents, Editing, E-signature, Sanitizable Signature, Hash, Aggregate Signature

1. はじめに

現在様々なディジタルコンテンツが活用される中で、署名後のディジタルコンテンツがそのまま利用されるとは限らない。例として、行政文書が情報公開制度に基づき開示される場合には、個人情報保護の観点から墨塗りを行った上で開示される必要がある。しかし、従来の電子署名技術では署名後のコンテンツに対してはいかなる変更も加えられないので、文書の真正性確保とプライバシー保護の両立が難しかった。そこで提案されたのが墨塗り署名技術であり、この署名方式によって署名後の文書に対して部分情報の秘匿が可能になった[1][2]

一方、冒頭で述べたように現在のディジタルコンテンツはあらゆる場面で活用されている。しかし、ディジタルコンテンツに対する従来の著作権保護技術はその視聴制御が中心で、コンテンツの編集許諾を細かく制御する技術はあまり考慮されていない。また、前述の墨塗り署名技術[1][2]も行政文書に対する一部削除や墨塗りを目的としたものが多く、ディジタルコンテンツの更新や追加までを含む著作権保護への応用についてあまり考慮されていなかった。それには、以下の理由が考えられる。

まず、行政文書に適用される墨塗り署名方式では、最終的な墨塗り文書において開示部分の完全性と非開示部分の秘匿性が守られていれば正当な墨塗り文書として扱われる。すなわち墨塗り箇所を墨塗り者が決定し、署名者が墨塗り箇所につ

いて関与しなくても、それが行政機関で作成された墨塗り文書であれば正当な墨塗り文書ということになる。しかし、著作権の保護が必要になるコンテンツにおいては問題が生じる。墨塗り者によって墨塗り箇所が決定されていることから、その時点で署名者の著作権は失われていると考えるべきである。著作権を守るためには、墨塗り箇所について署名者が何らかの形で関与する必要がある。

また、墨塗り署名方式は部分情報の非開示を目的とし、別のデータと入れ替えることについてはあまり考慮されていない。墨塗り署名方式の一つである PIAT 署名方式[3]では、墨塗り者によってランダムに決められたデータ列と入れ替えが可能となっているが、この場合も署名者の著作権は保護されない仕組みになっている。また、PIAT 署名方式ではコンテンツに添付する情報量が多すぎることも指摘されており、これについての改良案が提案されている[5]。しかし、この方式も著作権の観点から見た場合はオリジナルの PIAT 署名方式同様に保護されない。

また、従来の墨塗り署名方式でデータの追加まで対処可能なものはない。しかし、著作権保護においてはコンテンツの一部削除・変更に加えて追加を考慮することは重要である。

そこで本稿では、署名者が署名生成時に一定の制限を加えることでデータの変更・追加と著作権の保護を両立させられるような署名方式を 3 つ提案する。提案方式 1 ではデータベースを設け、更新可能データを予め用意することで実現し、提案方式 2 では方式 1 のデータベースをなくす代わりに著作者・利用者間でのデータのやり取りを行う。また、提案方式 3 では方式 2 のやり取りを Aggregate 署名に置き換えて実現するものとする。以下、2.～4.においてその 3 つの提案方式を説明し、5.において各方式の評価を行う。

2. 提案署名方式 1

2.1.1 提案方式 1 の概要

提案方式 1 は PIAT 署名方式と同様に部分情報を別のデータと入れ替えることが可能で、さらに、更新できるデータを署名者（オリジナルコンテンツの著作者）があらかじめデータベースに用意しておくという制限を加えることで更新後のコンテンツに関しても署名者の著作権が守られるようになっている。（図 1 は署名者がオリジナルコンテンツとともに更新可能なデータ群をデータベースに格納し公開していることを示している。）この方式は部分情報の墨塗りではなく著作権保護を目的としたものであり、想定するエンティティとしては署名者・変更者（コンテンツの編集を希望するユーザ）・検証者（コンテンツを表示するエディタであり、正当な署名のない不正なコンテンツは表示しない）という 3 者を用意する。それぞれの役割としては墨塗り署名方式とほぼ同様で「署名者によるコンテンツの作成」「変更者による許可されたデータの更新（変更・追加・削除）」「検証者による署名の検証」となっている。

2.1.2 原理

ディジタルコンテンツに対して効率よく部分データの更新（変更・追加・削除）が行えるような仕組みとして更新前と更新後の部分データの差分値を利用することを考える。さらに、差分値と署名をペアとして更新可能な部分データに付加情報として添付しておくことで簡単に検証を行えるようにできる。（図 2 では第 2 ブロックの内容を M_2 から M_2'に変更した場合を示しており、変更可能データ M_2'に元データ M_2 とのハッシュ値の差分値とその署名が添付されている。）まず、ハッシュ値集合とその署名を検証する。そのハッシュ値が正しければ、更新データのハッシュ値から差分値を減算して更新前のハッシュ値を復元し、ハッシュ値集合の更新位置のハッシュ値と比較検証する。これによって、追加や削除の場合に関しても変更の場合と同じ処理で実現することができる。どの場合も、添付すべき情報は変更箇所についての差分値・署名ペアなので変更箇所数に依存する。

2.1.3 アルゴリズム

署名者・変更者・検証者のそれぞれの役割をアルゴリズムとともに説明する（図 3 参照）。

署名者は最初にオリジナルコンテンツとその署名を生成し、更新を許可するデータ群を作成する。その後に各更新データとオリジナルの部分データとの差分値を求め、署名（厳密には部分データの位置を示す ID_i と差分値の連接値に対する署名）を生成する。その差分値・署名ペアを更新データに添付し、データベースに保存する。変更者はオリジナルコンテンツとデータベースから更新箇所とデータを選択し、新たな更新後のコンテンツを作成する。このときコンテンツには更新したブロックの差分値・署名ペアを添付する。それらを受け取った検証者は、まずハッシュ値集合とその署名を検証し、さらに、署名 $\sigma^{(j)}$から更新部分データに添付された差分値データの正真性を検証する。それらが正しければ、更新データからハッシュ値を生成し、更新データに添付された差分値を減算することでオリジナル部分データのハッシュ値が得られる。次に、更新部分データの署名検証で差分値と同時に得られる ID_j を参照し、ハッシュ値集合 h から j に該当するオリジナル部

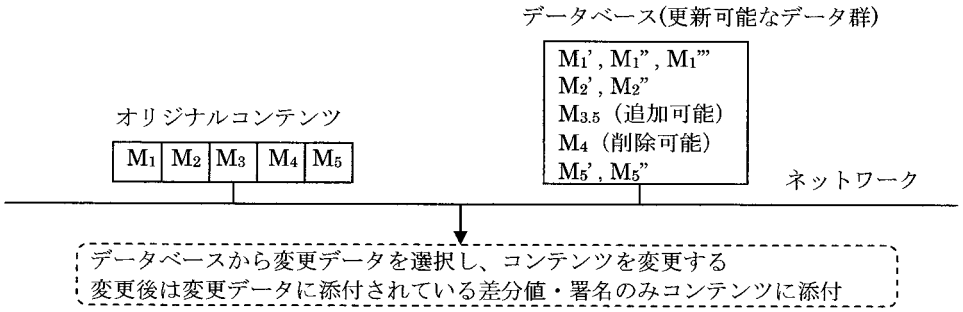

図 1. 提案方式 1 の構成

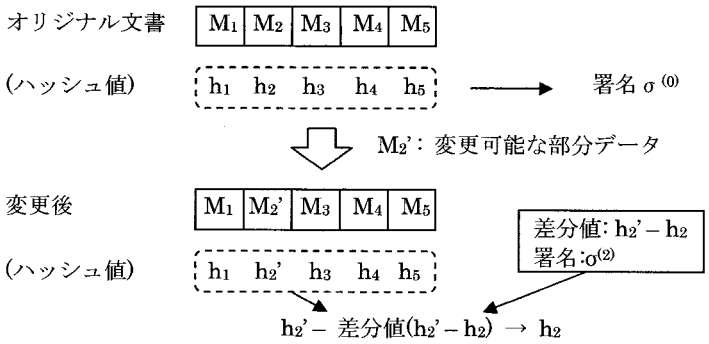

図 2. 提案方式 1 の原理

分データのハッシュ値 h_j を得る。それらのハッシュ値を比較検証し、成功すれば署名者の著作権が保護されているか確認できる。

以上のような手順で、更新後のコンテンツに対する署名者の著作権保護が実現できる。

墨塗り署名方式とは違い、変更者は墨塗り処理を行わない。また、部分データ変更と同じ処理で、データの削除や追加を行うこともできることもわかる。ブロックデータ自体を削除・追加したい場合には、そのハッシュ値をそのまま利用することで簡単に実現できる。削除の場合にはハッシュ値に－をつけ保存し、$-h_i$, $\sigma^{(i)}=Sign_{sk}(ID_i \| -h_i)$を添付する。検証の際は、$-h_i$ を減算することでオリジナルのハッシュ値集合を復元できる。また、追加の場合には h_i', $\sigma=Sign_{sk}(ID_i \| h_i')$を添付し、検証の際に h_i'を減算することでオリジナルのハッシュ値集合を復元できる

3. 提案方式 2

3.1.1 提案方式 2 の概要

提案方式 1 では更新可能なデータがあらかじめ署名者によって用意されており、その範囲内でのデータ変更は可能であるが、変更者による編集の自由度は小さくなってしまっている。そこで提案方式 2 ではデータベースを用いることなく、変更者が自由に更新データを作成し、そのデータを署名者が審査することによって、より自由度の高い編集許諾システムを実現することを目的としている。想定するエンティティ、それぞれの役割は提案方式 1 と同様である。図 5 は、署名者によって作成されたコンテンツデータを変更者が更新した際の署名者・変更者間でのやり取りを示しており、（①署名者が変更者にコンテンツを配布②変更者は自分が変更した部分データを署名者に送信③署名者は許可するデータには署名をつけて変更者に送り返す。）更新後のデータは検証者によって検証されることを示している。

3.1.2 原理

提案方式 1 同様である。この方式もコンテンツの変更・削除・追加が可能である。

3.1.3 アルゴリズム

著作者は図 3 の「署名者によるコンテンツの作成」の 1 の処理を行う。それ以外の処理は前処理

○ 署名者によるコンテンツ作成

入力：オリジナルコンテンツ(M_1~M_n)

1．各部分コンテンツ M_i(i=1,.....,n)に対し、ハッシュ値 $h_i \leftarrow H(M_i)$生成
署名 $\sigma^{(0)} \leftarrow Sign_{sk}(h)$を生成（sk：署名者の秘密鍵）　※$h = h_1 \| \ldots \| h_n$

2．更新を認める部分データ M_j'の作成
※ 変更・削除の場合：j=i、iは1~nの中で署名者により任意決定
※ 追加の場合：j>nとするか、iとi+1の間への挿入の場合、jは便宜上i.xの実数
※ 更新を認めるデータは1つの部分データに対し複数作成できるが、ここではM_j'で説明

3．h_iとh_j'から差分値を求め、署名を生成する。(ID_j：部分データ識別子)
※ 変更の場合：$\sigma^{(j)} \leftarrow Sign_{sk}(ID_j \| h_j' - h_i)$　　但し　$h_j' \leftarrow H(M_j')$
※ 削除の場合：$\sigma^{(j)} \leftarrow Sign_{sk}(ID_j \| 0 - h_j)$　　但し　$h_j \leftarrow -H(M_j)$
※ 追加の場合：$\sigma^{(j)} \leftarrow Sign_{sk}(ID_j \| h_j' - 0)$　　但し　$h_j' \leftarrow H(M_j')$

4．3の処理で得られた差分値・署名ペアを更新可能データM_j'に添付し、データベースに保存

出力：オリジナルコンテンツ($M_1 \sim M_n$)，差分値・署名ペア$\{(ID_j \| h_j' - h_i), \sigma^{(j)}\}$
署名 $\sigma^{(0)}$，ハッシュ値集合 h

○ 変更者によるデータベースにあるデータの更新（変更・追加・削除）

1．オリジナルコンテンツと参照可能なデータベースから更新を行うブロックとデータを決定

2．参照可能なデータベースから差分値・署名ペアが添付された更新可能データを選択
該当するブロックのオリジナル部分データを更新　※ $M_i \leftarrow M_j'$ または …,M_i, $\boxed{M_j'}$,M_{i+1},…

3．更新後のコンテンツに対応する差分値・署名ペアを添付して出力

出力：更新後のコンテンツ(M_1~M_n)，差分値・署名ペア$\{(ID_j \| h_j' - h_i), \sigma^{(j)}\}$
署名 $\sigma^{(0)}$，ハッシュ値集合 h

○ 検証者による署名の検証

1．署名 $\sigma^{(0)}$からハッシュ値集合 h の正真性検証

2．更新されたデータからハッシュ値生成。さらに署名 $\sigma^{(j)}$から更新データに添付された差分値データの正真性を検証後それぞれのハッシュ値を減算
※ $Dec_{pk}(\sigma^{(j)}) \Rightarrow ID_j \| h_j' - h_i$　さらに　$h_i' - (h_i' - h_i) \rightarrow h_i$　（pk：署名者の公開鍵）

3．検証2の署名検証でハッシュ値と同時に得られるID_jを参照し、ハッシュ値集合 h から j に該当するオリジナル部分データのハッシュ値 h_j を得る

4．検証2・3で得られたハッシュ値を比較検証

図 3. 提案方式 1 のアルゴリズム

○変更者と署名者によるデータの更新（変更・追加・削除）

1．変更者はオリジナルコンテンツから自分が変更・追加したいブロック M_i を決定し、自由に編集
署名者に安全な通信路を用いて送信

2．部分コンテンツデータ：M_i　⇒　更新データ：M_j'（追加の場合 $M_j = h_j = 0$，削除の場合 $M_j' = h_j' = 0$）

※ 署名者は送られてきた更新データの可否を決定し、可に限って更新前と更新後の部分データのハッシュ値の差分値と、その差分値の署名を更新データに添付して変更者に送信
※ 添付データは図3の署名者によるコンテンツ作成3参照

3．署名者から返ってきた自分の更新データに差分値・署名ペア$\{(ID_j \| h_j' - h_i), \sigma^{(j)}\}$が添付されたものをオリジナルの部分データと変更、または追加・削除。

出力：更新後のコンテンツ(M_1~M_n)，差分値・署名ペア$\{(ID_j \| h_j' - h_i), \sigma^{(j)}\}$
署名 $\sigma^{(0)}$，ハッシュ値集合 h

図 4. 提案方式 2 のアルゴリズム

としては行わないため、図4では記述を省略する。

変更者は部分データを自由に編集し、更新（変更・追加・削除）するデータを署名者に送信する（図4参照）

データを受け取った署名者はその更新の可否を決定し、可の場合のみ更新前と更新後の部分データのハッシュ値の差分値と、その差分値の署名を変更者に送り返す。変更者は返ってきた自分の更新データとオリジナルデータを入れ替える。

検証者は、方式1と同様の処理により検証に成功すれば署名者の著作権が保護されているか確認する。（検証処理も図3と同様のため図4から省略）

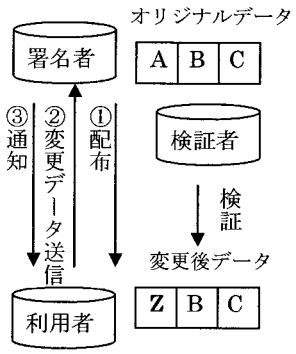

図5. 提案方式2，3の流れ

4. 提案方式3

4.1.1 提案方式3の概要

提案方式1，2では部分コンテンツごとにハッシュ値を取り、それらを連接させることによって一つの署名を生成していた。それに対し、提案方式3ではAggregate署名を用いることによって、各部分データごとに署名を生成し、それらを一つの署名に集約させる方式をとる。また、提案方式2と同様に図5の流れを利用する。

4.1.2 原理・アルゴリズム

ページ数の都合でAggregate署名の原理やアルゴリズムに関する詳細説明は省略する。[6]の手法を用いれば、コンテンツの変更と削除の制御が可能である。本方式ではそれに加えて図5の流れにより署名者が関与することで図6に示すように追加制御も可能になる。

5. 各提案方式の比較

各方式について事前署名数、更新時署名数、検証回数の3つの項目において比較し、表1に示す。

方式1において、データベースに保存する更新可能データ数をLとすると、署名者は事前にL+1（1はオリジナルコンテンツに対する署名）回の署名を行う必要がある。しかし、更新時に変更者によって変更・追加・削除されても改めて署名を生成する必要はない。また、検証回数は更新時にm回の更新を行ったとすれば、m+1回になる。添付される署名データ数も同様である。

方式2では事前の署名は全体署名の1つである。変更者によってm個のデータが変更・追加・削除されると、その分だけ署名を行い、検証回数及び添付署名データ数はm+1となる。

方式3ではAggregate署名を用いるに当ってブロックデータ数nだけ事前に署名を施しておく必要があり、各個別署名をAggregateした署名を含めると+1となる。変更者によるm個の変更・追加・削除データに対する署名数は方式2同様mである。また、検証回数は全体に対する検証だけの場合1回、部分データの真正性も検証する場合n+1回となる。添付署名データ数は削除可能なデータ数（すべて削除可能であればn）+1（Aggregate署名）となる。

L＞m，L+1＞n+mとすると、施す署名数は方式2が最も少なく、次いで方式3，方式1の順で効率的と言える。検証時の署名検証回数は方式3では全体検証のみであれば1回と最も少なく、個別検証まで行うとn+1回となる。n＞mであれば、方式2が最も効率的であるといえる。

表1. 各方式の比較

	事前署名数	更新時署名数	検証回数
1	L+1	0	m+1
2	1	m	m+1
3	n(+1)	m	(n+)1

6. まとめ

著作権保護の観点から、著作者の権利が保護できるコンテンツの編集システムを提案した。このシステムでは更新データの内容を見て更新可否が判断できるので著作者の意思が最も反映できる。しかし、方式1は大量の更新可能データを予め準備しておく必要があり、また方式2はデータの変更・追加・削除処理において、著作者の処理が必要であり著作者の負荷が大きい。方式3は削除を除いて著作者の処理が必要である。今後は方式3を拡張し、データの追加・変更に対しても著作者の負荷のないシステムを検討していく。特に、著作権の保護と著作者の負荷軽減がバランスするシステムという観点からの拡張を検討していく。

○署名者

入力：オリジナルコンテンツ(M_1~M_n)

1．コンテンツ識別子 ID・部分コンテンツデータ識別子 ID_i 生成

2．各部分コンテンツ M_i(i=1,.....,n)に対し、ハッシュ値 $h_i \leftarrow H(ID \parallel ID_i \parallel M_i)$を求める。
$\sigma^{(0)} \leftarrow Sign_{sk}(ID)$，個別署名 $\sigma^{(i)} \leftarrow Sign_{sk}(ID \parallel ID_i \parallel h_i)$を生成する。(sk は署名者の秘密鍵)

3．Aggregate 署名 $\sigma \leftarrow \prod_{i=0}^{n} \sigma^{(i)}$

出力：オリジナルコンテンツ($ID \parallel ID_i \parallel M_1 \sim ID \parallel ID_i \parallel M_n$)，Aggregate 署名 σ，
個別署名集合$\{\sigma^{(i)}(※i \in D,0)\}$，削除可能集合 $D=\{d_1,\cdots,d_s\}$

○変更者

＜変更の場合＞

1．オリジナルコンテンツから自分の変更したいブロック M_j を決定し、自由に編集、署名者に送信。
部分コンテンツデータ：M_j ⇒ 変更データ：M_j'

※ 署名者は送られてきた編集データの可否を決定し、可に限って変更後の部分データの
ハッシュ値とその署名を変更者に送信
ハッシュ値：$h_j' \leftarrow H(ID \parallel D_j \parallel M_j')$，署名：$\sigma^{(j)}{}' \leftarrow Sign_{sk}(ID \parallel ID_j \parallel h_j')$

2．署名者から返ってきたハッシュ値・署名ペア$\{h_j', \sigma^{(j)}{}'\}$に対応する編集データをオリジナルの部分データと入れ替え

3．Aggregate 署名 σ から $\sigma^{(j)}$を削除し、$\sigma^{(j)}{}'$を追加
$\sigma = \sigma^{(1)} \times \cdots \times \sigma^{(j)} \times \cdots \times \sigma^{(n)} \rightarrow \sigma = \sigma^{(1)} \times \cdots \times \sigma^{(j)}{}' \times \cdots \times \sigma^{(n)}$

＜削除の場合＞

1．削除可能集合から自分が削除したいブロックが削除可能か確認。(※ $ID \parallel ID_k \parallel M_k$ の削除)

2．削除可能なら、部分データ $ID \parallel ID_i \parallel M_k$ を削除
削除可能集合 $D=\{d_1,\cdots,d_s\}$から d_k を、Aggregate 署名 σ から $\sigma^{(k)}$ 削除

＜追加の場合＞

1．変更の場合と同様、追加したいブロック M_p を作成し、署名者に送信（※ $ID \parallel ID_p \parallel M_p$ の追加）

※ 署名者は可に限って追加データのハッシュ値とその署名を変更者に送信

2．署名者のハッシュ値・署名ペア$\{h_p, \sigma^{(p)}\}$に対応する更新データをオリジナルコンテンツに挿入

3．Aggregate 署名 σ に $\sigma^{(p)}$ 追加 $\sigma \leftarrow \sigma \times \sigma^{(p)}$

出力：変更・削除・追加後のコンテンツ($ID \parallel ID_i \parallel M_1 \sim ID \parallel ID_i \parallel M_n$)，Aggregate 署名 σ ($\sigma^{(k)}$除く)，
個別署名集合$\{\sigma^{(i)}(※i \in D,0)\}$，削除可能集合 $D=\{d_1,\cdots,d_s\}$(d_k 除く)

○検証者

1．コンテンツ識別子 ID が各部分コンテンツの ID に等しいか検証

2．各部分コンテンツの ID_i が昇順になっているか検証

3．コンテンツのすべての部分データから $h \leftarrow \prod_{i=0}^{n} h^{(i)}$を算出

4．$h \leftarrow Dec_{pk}(\sigma)$が成立するか検証（pk：署名者の公開鍵）

5．部分データの検証を行う場合、個別署名 $\sigma^{(i)}$からデータの正真性検証

図 6. 提案方式 3 のアルゴリズム

文献

[1] 宮崎邦彦, 洲崎誠一, 岩村充, 松本勉, 佐々木良一, 吉浦裕, “電子文書墨塗り問題”,信学技法 ISEC2003-20, 61-67, 2003

[2] 増渕孝延, 小川典子, 鹿志村浩志, 石井真之, 佐々木良一, “より効率的な墨塗りシステムの開発と評価”,信学技法 ISEC2004-25, pp.7-13, 2004

[3] 武仲正彦, 吉岡孝司, 金谷延幸, “検証者が署名者と墨塗り者を検証可能な墨塗り方式”,CSS2004, 6C-3, pp. 475-480, 2004 年 10 月

[4] 武仲正彦, 伊豆哲也, 吉岡孝司, “電子文書の訂正・流通を考慮した部分完全保障方式の改良”, SCIS2006 2B2-4

[5] 伊豆哲也,佐野誠,國廣昇,太田和夫,竹仲正彦,“Aggregate 署名を用いた墨塗り署名方式”,SCIS 2007,2007

[6] 佐野誠,伊豆哲也,國廣昇,太田和夫,竹仲正彦,“部分情報の墨塗りと削除が可能な電子署名方式について”,SCIS 2007,2007