社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－134 (36)
2008－CSEC－ 40 (36)
2008／3／7

墨塗り必須箇所を指定可能な電子文書墨塗り技術の検討

千賀 渉[†‡] 土井 洋[†]
[†]情報セキュリティ大学院大学 [‡]KDDI 株式会社
E-mail: [†]{mgs074503, doi}@iisec.ac.jp

あらまし 署名つき電子文書の一部を秘匿（墨塗り）しようとすると，署名の検証ができなくなる．この問題を電子文書墨塗り問題とよび，これを解決する電子文書墨塗り技術が近年活発に提案されている．本稿では，署名者が墨塗り必須箇所を指定可能な電子文書墨塗り技術として，3つの方式を提案する．最後に示す方式では，あらかじめ指定した箇所の不正開示を防ぐことが可能となる．

A Study of Designated Sanitizing Area Signature Scheme

Wataru Senga[†‡] Hiroshi Doi[†]
[†]Institute of Information Security [‡]KDDI Corporation
E-mail: [†]{mgs074503, doi}@iisec.ac.jp

Abstract. In this paper, we propose three sanitizable signature schemes that the signer can specify the closed area. The third scheme accomplishes that any sanitizer cannot reveal the sanitizing area where the signer specified.

1. はじめに

署名付きの文書を公開する際，オリジナルの文書から個人情報や機密情報などを秘匿し，開示しなければならない場合がある．紙文書であれば，秘匿したい箇所に墨塗り処理を施してから開示すればよい．しかし電子文書では，一度署名を付けられた文書に対して，墨塗りのように変更を行うと文書の改ざんとみなされ，署名の検証ができなくなってしまう．

宮崎らはこの問題を電子文書墨塗り問題と呼び，これを解決する技術を墨塗り技術と名付けた[2]．墨塗り技術とは，一般に以下のエンティティから構成される署名方式である．

- 署名者：オリジナル文書に対し署名を生成する．
- 墨塗り者：署名者によって署名の付与されたオリジナル文書に対し，情報を秘匿する箇所に墨塗りを行い，部分開示する．
- 検証者：署名検証し，部分開示された文書が署名者によって保証されていることを確認する．

例えば，署名者と墨塗り者は同一組織内の人物で，検証者は組織外の第三者の場合が多い．

本研究では，組織内の不正者が組織外の第三者に不正に情報開示を行うことを想定し，墨塗り技術を用いてこれを防止する方法について検討を行った．あらかじめ指定された墨塗り必須箇所を開示しようとすると，開示した人物が特定できる方法（4.1 節, 4.2 節），および，第三者に対しては開示が不可能な方法（4.3 節）を検討した結果を報告する．

2. 従来の墨塗り技術

Steinfeld らは文献[1]にて Content Extraction Signatures(CES)方式を，宮崎らは文献[2]にて SUMI-4 方式を提案している．これらの方式は，署名付きの文書を入手すると誰でも墨塗り者として振る舞うことができ，墨塗り箇所の制約もなく自由に墨を塗ることができる．

宮崎らは SUMI-4 で可能であった無制限の墨塗りを防止するため，墨塗り禁止箇所を指定できる SUMI-5 方式[3]，SUMI-6 方式[4]を提案した．

SUMI-6 方式では Boneh らの Aggregate 署名[5]の技術を用いることにより，検証者には墨塗りされた部分文書の存在すら検知できないように構成しており，これは伊豆らの提案した方式[6]や，佐野，泉らの提案した墨塗り・削除署名[7][8]にも応用されている．

Ateniese らはカメレオンハッシュ関数を用いることにより，あらかじめ指定された墨塗り者のみが墨塗りを行える Sanitizable Signatures(SS) 方式を提案した[9]．伊豆らの提案した PIAT 署名[10]では，墨塗りを行う都度，墨塗り者の署名を追加していくことで，墨塗り者を特定することが可能となっている．また永村らは，墨塗り者が署名者によってあらかじめ指定され，かつ検証者からは誰が墨塗りを行ったのか識別できない ID ベース型墨塗り署名を提案した[11]．

増渕らはデータ量を削減して効率的な墨塗り方式を提案した[12]．また，増渕らは文書開示後に墨塗り箇所変更要求があった場合に，対応可能な方式(MNI+05)を提案した[13]．

2.1. 墨塗り技術の安全性要件

墨塗り技術の安全性要件として，以下の 2 つの要件が定義されている[1][3]．

偽造不可能性：検証可能な署名付き墨塗り文書を偽造できないこと．

秘匿性：墨塗り部分に対応する元の文書の情報が漏れないこと．

2.2. SUMI-4 方式[2]の概要

本節では，従来方式として代表的な墨塗り技術である宮崎らの SUMI-4 方式の概要を述べる．

オリジナル文書を任意の数の部分文書に分割し，各部分文書のハッシュ値を計算する（各部分文書が墨塗り可能な最小単位となる）．そしてこれらのハッシュ値を結合したデータに対して署名を付与する．開示する際には，開示部分には元の部分文書を，墨塗り部分には元の部分文書の代わりにそのハッシュ値を選び，署名とともに開示する．なお墨塗り前の情報の推測を防ぐため，各部分文書のハッシュ値計算に，乱数を入力に加えている．

3. 研究動機

SUMI-4 を含め従来方式の多くは．墨塗りを行うかどうかは，墨塗り者の判断に依存する．

さて，組織内の（正当な権限を持つ）人物が，墨塗り前の署名つき文書を入手したとする．この文書には個人情報や機密情報が含まれており，外部に開示する際には墨塗り必須の箇所があったとしても，この人物が墨塗りを行わなければ，そのまま開示することができてしまう．

CES 方式[1]や増渕らの方式[12]では，各部分文書毎に墨塗り可否を指定したフラグを付与し，検証者が開示された文書との整合を確認できるようになっている．しかし，墨塗り者がフラグの指定に従わずに文書を公開しても署名検証自体は可能である．すなわち，不正開示された文書に対して，検証者がフラグの内容を無視して検証処理を行えば，開示された文書の正当性を確認できる．

MNI+05 方式[13]は，部分文書に対応する乱数情報を，秘密分散法により複数の墨塗り者でシェアして管理している．この方式ではシェアされた秘密を復号しなければ，部分文書を開示することができない．

本研究では MNI+05 方式とは別の手法で，開示情報の制御に関する問題の解決を目指す．墨塗り者が単独で墨塗りや開示処理を行うことができ，なおかつ，署名者があらかじめ墨塗り必須箇所を指定可能な，電子文書墨塗り技術の提案を行う．

4. 提案方式

本章では，墨塗り必須箇所を指定可能な 3 つの提案方式について述べる．

4.1. 提案方式 1

SUMI-4 をベースとして，墨塗り必須の部分文書の乱数データに，墨塗り者の ID を埋め込む．もし墨塗り者が墨塗り必須箇所を開示した場合，墨塗り者の ID も同時に開示されるため，不正な墨塗り者の追跡が可能となる．

準備

1. 鍵生成アルゴリズムを $\mathcal{K}_1$，署名生成アルゴリズムを $\mathcal{S}_1$，署名検証アルゴリズムを $\mathcal{V}_1$ とする任意の署名方式を用意する．適切なハッシュ関数 $\mathcal{H}$ を用意する．

2. セキュリティパラメータ k_1 を入力として，$\mathcal{K}1(1^{k_1})=(pk_A,sk_A)$ を計算し，署名者の公開鍵 pk_A，秘密鍵 sk_A を出力する．

署名生成

1. 元の文書 m を n 個の部分文書 $(m_1,\cdots,m_n)$ に分割し，乱数 $(r_1,\cdots,r_n)$ を生成する．
2. 開示時に墨塗り必須とする部分文書を選び，その添え字集合を X とする．
3. 墨塗り者を指定する．ここで，指定された墨塗り者は一意に識別可能な識別子 ID_B を持つものとする．
4. $i=(1,\cdots,n)$ に対し，以下を計算する．

$$M_i=\begin{cases} m_i \| r_i & (i\notin X) \\ m_i \| r_i \| ID_B & (i\in X) \end{cases},\quad h_i=\mathcal{H}(M_i)$$

5. 署名 $\sigma_A=\mathcal{S}1_{sk_A}(h_1\|\cdots\|h_n)$ を計算する．
6. 署名，乱数つき部分文書および添え字集合 $(M_1,\cdots,M_n,\sigma_A,X)$ を出力する．

墨塗り

1. $m_i(i\notin X)$ の部分文書のうち追加で墨塗りする部分文書を定め，その添え字集合を Y とする．
2. $i=(1,\cdots,n)$ に対し，以下を求める．

$$\bar{M}_i=\begin{cases} M_i & (i\notin X\cup Y) \\ \mathcal{H}(M_i) & (i\in X\cup Y) \end{cases}$$

3. 署名付き開示文書 $(\bar{M}_1,\cdots,\bar{M}_n,\sigma_A)$ を出力する．

検証

1. $i=(1,\cdots,n)$ に対し，以下を求める．

$$h'_i=\begin{cases} \mathcal{H}(\bar{M}_i) & \text{(開示箇所)} \\ \bar{M}_i & \text{(墨塗り箇所)} \end{cases}$$

2. $\mathcal{V}1_{pk_A}(\sigma_A,h'_1\|\cdots\|h'_n)$ を検証する．

評価

もし墨塗り者が，署名者の指定した墨塗り必須箇所を墨塗りせずに開示した場合，墨塗り者自身の ID も一緒に開示される（仮に ID を削除して開示しようとしても，署名検証に失敗する）．すなわち，不正開示された文書から，不正開示者を知ることが可能となる．

問題点として，署名者が墨塗り者に無断で墨塗り者の ID を埋め込んだ文書に署名することが可能なことがあげられる．この場合，署名者が墨塗り者になりすまして不正に文書を作成し開示しても，墨塗り者が不正開示したことにされてしまう．すなわち，署名者が墨塗り者を陥れることが可能である．

4.2. 提案方式 2

提案方式 1 の問題に対処するため，墨塗り者を識別する情報として，ID ではなく墨塗り者の署名を埋め込む方式を提案する．

準備

1. 任意の 2 組の鍵生成，署名生成，署名検証アルゴリズム $(\mathcal{K}1,\mathcal{S}1,\mathcal{V}1)$ および $(\mathcal{K}2,\mathcal{S}2,\mathcal{V}2)$ を用意し，$(\mathcal{K}1,\mathcal{S}1,\mathcal{V}1)$ を署名者用，$(\mathcal{K}2,\mathcal{S}2,\mathcal{V}2)$ を墨塗り者用のアルゴリズムとする．適切なハッシュ関数 $\mathcal{H}$ を用意する．
2. セキュリティパラメータ k_1，k_2 を入力として $\mathcal{K}1(1^{k_1})=(pk_A,sk_A)$，$\mathcal{K}2(1^{k_2})=(pk_B,sk_B)$ を計算し，署名者の公開鍵 pk_A，秘密鍵 sk_A，墨塗り者の公開鍵 pk_B，秘密鍵 sk_B を出力する．

墨塗り必須箇所の確定

1. 元の文書 m を n 個の部分文書 $(m_1,\cdots,m_n)$ に分割し，乱数 $(r_1,\cdots,r_n)$ を生成する．
2. 開示時に墨塗り必須とする部分文書を選び，添え字集合を X とする
3. 署名者は墨塗り者を指定し，m および $m_i(i\in X)$ を墨塗り者に送る．
4. 指定された墨塗り者は，各 $m_i(i\in X)$ に対して署名 $\sigma_{Bi}=\mathcal{S}2_{sk_B}(m_i)$ を計算し，署名者に送る．
5. 署名者は墨塗り者の公開鍵を用い，$\mathcal{V}2_{pk_B}(\sigma_{Bi},m_i)$ 検証する．

署名生成

1. $i=(1,\cdots,n)$ に対し，以下を計算する．

$$M_i=\begin{cases} m_i \| r_i & (i\notin X) \\ m_i \| r_i \| \sigma_{Bi} & (i\in X) \end{cases},\quad h_i=\mathcal{H}(M_i)$$

以下，提案方式 1 と同様の処理を行う．

墨塗り

提案方式 1 と同様の処理を行う．

検証

提案方式 1 と同様の処理を行う．

評価

方式 1 で問題となった，署名者自身が墨塗り者の同意を得ることなく文書を作成して内部情報などを開示しようとしても，開示された部分文書に対する墨塗り者の署名がなければ，墨塗り者は

潔白を主張できる．

しかし，署名者が墨塗り必須箇所の確定フェーズにて墨塗り者の署名を一旦入手してしまえば，その署名を（墨塗り者の同意なく）別の文書に使用することは可能である．文書の内容を自由に選ぶことはできないため，方式1と比べて陥れる条件が限定されるが，墨塗り者を陥れることも可能と考えられる．

4.3. 提案方式3

4.2節で示した問題点に対応する方法として，墨塗り必須と決めた箇所を開示しても，第三者には検証ができない方式を検討する．具体的には，署名者と墨塗り者には検証可能で，（第三者である）検証者には検証不可能となるように構成する．

署名者は，墨情報 h_i を構成する乱数データを直接送らず，自分が乱数データを知っていることを墨塗り者に証明する．こうすることで，仮に部分文書が開示されたとしても，検証者として指定された墨塗り者以外の第三者には検証ができない．

4.3.1. 指定検証者署名

上記のような性質をもつ署名方式として，指定検証者署名が知られている．以下，指定検証者署名の概要について述べる．

指定検証者署名（DVS）

指定検証者署名（DVS; Designated Verifer Signature）はJakobssonらによって提案された署名方式[14]である．AliceがBobを指定検証者として生成したDVSは，第三者Cindyからみると，Aliceが生成した署名なのかBobが生成した署名なのか区別がつかない．

強指定検証者署名（SDVS）

DVSでは，第三者CindyにはAliceかBobの少なくともどちらかが生成した署名であることは判る．一方，Bobのみが検証可能で，第三者Cindyには単なる乱数にしか見えない署名を強指定検証者署名（SDVS; Strong Designated Verifier Signature）という．文献[14]ではSDVSの概念だけが述べられ具体的な構成方法は示されていなかったが，その後Saeednia らによって初めて効率的な方式[15]が提案された．

4.3.2. 安全性要件

SDVSを用いて，署名者と墨塗り者には検証可能で，第三者には検証不可能な方式が満たすべき安全性要件を，以下のように定義する．

- **要件1（偽造不可能性）**
  署名 σ，指定検証者署名 δ と墨塗り文書の正しい組み合わせを，署名者以外に生成できないこと．
- **要件2（秘匿性）**
  墨塗り部分に対応する元の文書の情報が漏れないこと．
- **要件3（指定検証者性）**
  署名者および指定検証者である墨塗り者以外に，墨塗り必須箇所の検証ができないこと．

4.3.3. 提案方式3の概要

署名者は，墨塗り必須箇所に関して，墨情報を構成する乱数データを墨塗り者に直接送らず，乱数データを知っていることを証明するSDVSを送る．墨塗り者はSDVSにより，墨塗り必須箇所の検証ができるようにする．他はSUMI-4と同様である．

以下に処理の流れを示す．

署名生成

1. 文書 m を部分文書 $(m_1,\cdots,m_n)$ に分割し，対応する乱数データ $(r_1,\cdots,r_n)$ を生成する．部分文書のうち墨塗り必須箇所を定めてその添え字集合を X とし，対応する乱数データ $\{t_i\}_{i\in X}$ を生成する．
2. $i=(1,\ldots,n)$ に対し (m_i,r_i,t_i) から h_i を計算する．任意の署名方式を選び，h_i を結合した値 $(h_1\|\cdots\|h_n)$ に対する署名 σ を生成する．
3. 各墨塗り必須箇所に対して，乱数 t_i のSDVS δ_i を生成する．
4. $i=(1,\ldots,n)$ に対し，墨塗り必須箇所については (m_i,r_i,h_i,δ_i) を，それ以外の箇所については (m_i,r_i,h_i) を出力し，墨塗り者へ送る．

墨塗り

1. 署名 σ，およびSDVS δ_i を検証する．
2. 追加墨塗り箇所を定める．
3. $i=(1,\ldots,n)$ に対し，墨塗り箇所（墨塗り必須および追加墨塗り箇所）については $\bar{m}_i=h_i$ を，開示

箇所については $\bar{m}_i=(m_i,r_i)$ を出力し，検証者へ送る．

検証

1. 開示箇所については (m_i,r_i) から h_i を計算し，墨塗り箇所については $h_i=\bar{m}_i$ とする．
2. $i=(1,...,n)$ に対し h_i を結合し，署名者の公開鍵を用いて $(h_1\|\cdots\|h_n)$ および署名 σ を検証する．

4.3.4. 具体的な構成方法

提案方式 3 の具体的な構成方法を以下に示す．本方式で構成した SDVS と類似の方式に，Tso ら[16]による 2 者間鍵共有に基づく SDVS がある．

準備

1. 鍵生成アルゴリズムを $\mathcal{K}_1$，署名生成アルゴリズムを $\mathcal{S}_1$，署名検証アルゴリズムを $\mathcal{V}_1$ とする任意の署名方式を用意する．
2. セキュリティパラメータ k_1 を入力として，$\mathcal{K}_1(1^{k_1})=(pk_A,sk_A)$ を計算し，署名者の公開鍵 pk_A，秘密鍵 sk_A を出力する．
3. 大きな素数 $p,q(q\mid p-1)$ を生成し，乗法群 $\mathbb{Z}_p^*$ での位数が q となるような生成元 g を求める．墨塗り者の秘密鍵をランダムな $sk_B\in\mathbb{Z}_q$，公開鍵を $pk_B=g^{sk_B}\bmod p$ とする．適切なハッシュ関数 $\mathcal{H}:\{0,1\}^*\to\mathbb{Z}_p^*$ を用意する．

署名生成

1. 元の文書 m を n 個の部分文書 $(m_1,\cdots,m_n)$ に分割し，墨塗り必須とする部分文書を選び，その添え字集合を X とする．
2. 乱数 $(r_1,\cdots,r_n,t_1,\cdots,t_n)$ を生成する．但し，$i\notin X$ については，$t_i=0$ とする．
3. $i=(1,\cdots,n)$ に対し，$h_i=\mathcal{H}(m_i\|r_i)g^{t_i}$ を計算する．
4. 署名 $\sigma_A=\mathcal{S}_{1sk_A}(h_1\|\cdots\|h_n)$ を計算する．
5. $i\in X$ に対し，SDVS $\delta_i=pk_B^{t_i}\bmod p$ を計算する．
6. σ_A および，$i=(1,\cdots,n)$ に対し $(m_i,r_i,h_i,\delta_{i(i\in X)})$ を出力する．

墨塗り

1. $i\in X$ に対し，$\delta_i=(h_i/\mathcal{H}(m_i\|r_i))^{sk_B}$ の検証を行う．
2. $\mathcal{V}_{1pk_A}(\sigma_A,h_1\|\cdots\|h_n)$ を計算し，σ_A を検証する．
3. $m_i(i\notin X)$ の部分文書のうち追加で墨塗りする部分文書を決め，添え字集合 Y とする．
4. $i=(1,\cdots,n)$ に対し，以下を求める．

$$\bar{m}_i=\begin{cases}h_i & (i\in X)\\ \mathcal{H}(m_i\|r_i) & (i\in Y)\\ (m_i,r_i) & (i\notin X\cup Y)\end{cases}$$

5. 署名付き開示文書 $(\bar{m}_1,\cdots,\bar{m}_n,\sigma_A)$ を出力する．

検証

1. $i=(1,\cdots,n)$ に対し，以下を求める．

$$h'_i=\begin{cases}\mathcal{H}(m_i\|r_i) & \text{(開示箇所)}\\ \bar{m}_i & \text{(墨塗り箇所)}\end{cases}$$

2. $\mathcal{V}_{1pk_A}(\sigma_A,h'_1\|\cdots\|h'_n)$ を計算し，σ_A を検証する．

4.3.5. 評価

墨塗り必須箇所に関して，次のことがいえる．

署名 σ_A，部分文書 m_i の正当性を検証するためには乱数 t_i が必要だが，t_i は開示されない．しかし墨塗り者は，代わりに SDVS δ_i を用いて検証できる．

一方，墨塗り者が任意に選んだ $(\tilde{m}_i,\tilde{r}_i)$ に対して $\tilde{\delta}_i=(h_i/\mathcal{H}(\tilde{m}_i\|\tilde{r}_i))^{sk_B}$ を作ることができる．したがって，検証者は墨塗り者から $(\tilde{m}_i,\tilde{r}_i,h_i,\delta_i)$ を開示されたとしても，検証できない．

次に，安全性に関する評価を以下に述べる．

- **要件 1（偽造不可能性）**
  署名 σ_A は署名者の秘密鍵がなければ生成できない．したがって要件 1 は満たされる．
- **要件 2（秘匿性）**
  墨データ h_i は元の文書をハッシュ化して求めたものである．従ってハッシュ関数 $\mathcal{H}$ の一方向性を仮定すれば，要件 2 は満たされる．
- **要件 3（指定検証者性）**
  墨塗り必須箇所の検証には，SDVS を利用しているため，墨塗り者以外には検証不可能である．また第三者は墨塗り者の秘密鍵を用いたとしても，真の署名者による署名なのか墨塗り者による署名なのか区別ができないため，墨塗り必須箇所の正当性を検証することはできない．したがって要件 3 は満たされる．

5. 各方式の比較

従来方式と提案方式の機能を，①不正開示が発覚した場合，開示者を特定可能かどうか，②署名者によって墨塗り者を陥れることが防止できる

か，③不正開示自体が防止できるか，の観点から分類し比較を行った．結果を表 1 に示す．

表 1　従来方式と提案方式の比較

	①不正開示者の特定	②署名者の不正防止	③不正開示の防止
SUMI-4[2]	×	×	×
SS[8]	○	×	×
PIAT[9]	○	○	×
MNI+05[12]	−※1	×	○
提案方式 1	○	×	×
提案方式 2	○	△※2	×
提案方式 3	−※1	○	○

※1：不正開示ができないため．
※2：一部条件が限定される（詳細は 4.2 節を参照）．

6. まとめ

本稿では，署名者が墨塗り必須箇所を指定可能な電子文書墨塗り技術について検討を行い，具体的な実現方法として 3 つの方式を提案した．

今後は，効率的に複数墨塗り者に対応可能な方式を構成することと，提案方式における安全性要件の精査，およびフォーマルな安全性証明についての検討が課題である．

謝辞

本研究を進めるにあたり，様々な助言を頂いた東京電機大学の佐々木良一教授，齊藤泰一助教授，ならびに両研究室の方々に感謝の意を表する．

本研究の一部は，IISEC 客員研究員の大川直人氏の修士論文から着想を得た．また本研究の一部は，「研究と実務融合による高度情報セキュリティ人材育成プログラム」から支援を受けた．

参考文献

[1] R.Steinfeld, L.Bull, Y.Zheng, "Content Extraction Signatures", ICISC 2001,LNCS 2288, pp.285-304, 2001.

[2] 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"，信学技報 ISEC2003-20，pp.61-67，2003.

[3] K.Miyazaki, M.Iwamura, T.Matsumoto, R.Sasaki, H.Yoshiura, S.Tezuka, and H.Imai, "Digitally Signed Document Sanitizing Scheme with Disclosure Condition Control", ICICE Trans. Fundamentals, Vol.E88-A, No.1 January 2005.

[4] K.Miyazaki, G.Hanaoka, and H.Imai, "Invisibly Sanitizable Digital Signature Scheme", ICICE Trans. Fundamentals, Vol.E91-A, No.1 January 2008.

[5] D.Boneh, C.Gentry, B.Lynn, and H.Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. EUROCRYPT 2003, LNCS 2656, pp.416-443, 2003.

[6] T. Izu, N. Kunihiro, K. Ohta, M. Takenaka, and T. Yoshioka, "A Sanitizable Signature Scheme with Aggregation", ISPEC 2007, LNCS 4464, pp. 51–64, 2007.

[7] 佐野 誠，伊豆 哲也，國廣 昇，太田 和夫，武仲 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"，SCIS2007，2C4-1，2007.

[8] 泉 雅巳，伊豆 哲也，國廣 昇，太田 和夫，"墨塗り・削除署名の拡張"，信学技報 ISEC2007-67，pp147-153，2007.

[9] G. Ateniese, D. Chou, B. Medeiros, G. Tsudik, Sanitizable Signatures, ESORICS 2005, LNCS 3679, pp. 159-177, 2005.

[10] 伊豆 哲也，金谷 延幸，武仲 正彦，吉岡 孝司，"墨塗り者を特定可能な電子文書の墨塗り署名方式"，情報処理学会論文誌，Vol.48，No.9，pp2990-2997, Sep. 2007.

[11] 永村 建索，左 瑞麟，岡本 健，岡本 栄司，"墨塗り者の指定と匿名性を実現した ID ベース型墨塗り署名"，SCIS2007，2C4-2，2007.

[12] 増渕 孝延，小川 典子，鹿志村 浩史，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"，情報処理学会研究報告，2004-CSEC-25(2)，pp7-12，2004.

[13] 増渕 孝延，中村 創，石井 真之，小川 典子，鹿志村 浩史，佐々木 良一，"内部不正者を考慮した墨塗り箇所変更可能方式の提案"，信学技法，ISEC2005-33，pp.179-186，2005.

[14] M.Jakobsson, K.Sako, R.Impagliazzo, "Designated Verifier Proofs and Their Applications", EUROCRYPT1996, LNCS 1070, pp.143-154, 1996.

[15] S. Saeednia, S. Kremer, O. Markowitch, "An efficient strong designated verifier signature scheme", ICISC2003, LNCS 2971, pp.40–54, 2003.

[16] R. Tso, T. Okamoto, and E. Okamoto, "Practical Strong Designated Verifier Signature Schemes Based on Double Discrete Logarithms", CISC2005, LNCS 3822, pp.113-127, 2005.