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

2007－ＣＳＥＣ－38 (52)
2007／7／20

墨塗り・削除署名の拡張

泉 雅巳† 伊豆 哲也†† 國廣 昇† 太田 和夫†

† 電気通信大学 情報通信工学科 〒182-8585 東京都調布市調布ヶ丘 1-5-1
†† 株式会社富士通研究所 〒211-8588 神奈川県川崎市中原区上小田中 4-1-1
E-mail: †{masami,kunihiro,ota}@ice.uec.ac.jp, ††izu@labs.fujitsu.com

あらまし ディジタル署名が生成された電子文書において, 公開部分の完全性を保証したままで一部の情報を秘匿する技術として, 墨塗り署名・削除署名が注目を集めている. 佐野らはこれらを統合した墨塗り・削除署名を提案したが, 各部分文書が取り得る文書状態に制約が課されているという問題があった. また, 同じ条件下で同様な署名方式が構築できるかも議論されていなかった. 本論文は佐野らの方式を解析し, 特に上記の制約を必要としない墨塗り・削除署名方式が構築できることを示す. またいくつかの類似方式についても議論する.

キーワード 電子文書, ディジタル署名, 墨塗り署名, 削除署名, 墨塗り・削除署名, 部分文書状態, Aggregate 署名

An Extension of Sanitizable and Deletable Signature

Masami IZUMI†, Tetsuya IZU††, Noboru KUNIHIRO†, and Kazuo OHTA†

† The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
†† FUJITSU LABORATORIES Ltd., 4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan
E-mail: †{masami,kunihiro,ota}@ice.uec.ac.jp, ††izu@labs.fujitsu.com

Abstract Sanitizable or deletable signatures attract much attention because of their privacy-perserving property in which after generating a signature on an original document, partial information can be masked with keeping the integrity of disclosed part. Recently, Sano et al. combined these signatures into the sanitizable and deletable signature. However, there is a restrictions on subdocument status. In addition, possibilities whether similar signatures can be constructed was not discussed. This paper analyzes signatures of Sano et al., and constructs a new sanitizable and deletable signature with no restriction on subdocument status. Moreover, some similar siagntures are also constructed.

Key words E-document, digital signature, sanitizable signature, deletable signature, sanitizable and deletable signature , subdocument status, aggregate signature

1. はじめに

電子文書の普及につれて, 行政機関や企業は厳密な文書運用・管理が求められるようになっている. 例えば 2005 年に施行された e-文書法では, 行政機関は公文書の原則公開と公文書内の個人情報の秘匿の両立が求められている. マスキング (墨塗り) や削除は情報秘匿の手段として広く用いられているが, これら処理は文書内容の変更に当たるため, ディジタル署名の正当性を保持したままで文書内容を変更 (墨塗り, 削除) することは一般に困難である. 墨塗り署名 (Sanitizable Signature) はディジタル署名の一種で, 署名者が署名を生成した後に, 改訂者と呼ばれるエンティティが文書内容の一部を墨塗りすることを可能としている [1]. また削除署名 (Deletable Signauter) (注1) もディジタル署名の一種であり, 改訂者は部分的な文書内容 (と付随情報) を削除することによって, 情報秘匿を実現している [2]. 墨塗り署名や削除署名においては, 検証者による改訂文書の検証を通じて, 開示部分の完全性が保証される. また墨塗りされた文書, 削除された文書の内容を復元することが不可能なことから, 非公開部分の秘匿性も実現されている.

従来の墨塗り署名, 削除署名は墨塗りまたは削除の一方の機能しか有していないことから, 佐野らは両者を統合した墨塗り・削除署名を提案するとともに, 2 種類の具体的な方式 (SIKOT-I, SIKOT-II) を構築した [3]. 佐野らの墨塗り・削除署名では, 個々

(注1): 提案者らは墨塗り署名と分類しているが [2], 本稿は佐野らの分類 [3] に従った.

の部分文書の墨塗りや削除を行うことが可能である上に, 墨塗り・削除の可否を部分文書の状態として設定できるように設計されている. しかし部分文書状態に制限があり, 考えられる全ての状態を実現できていないという問題が指摘されていた [3]. また, 佐野らの墨塗り・削除署名は Aggregate 署名を利用しているが (SIKOT-I は 1 種類, SIKOT-II は 2 種類), 逆に同じ種類数の Aggregate 署名を用いた場合に, 同様な墨塗り・削除署名が実現できるかどうかについては検討されていなかった.

本稿は, 佐野らの墨塗り・削除署名の解析を目的として, Aggregate 署名の種類数を 1 種類, 2 種類に限定した場合の, 墨塗り・削除署名の実現性について検討する. 特に, Aggregate 署名が 1 種類の場合に SIKOT-I と (本質的に) 別の実現が可能であること, また 2 種類の場合については, SIKOT-II で成し得なかった全ての部分文書状態が実現可能であることを具体的な実現例を挙げて示す.

本稿の構成は以下の通りである. 2 節で Aggregate 署名についてまとめた後, 3 節にて佐野らの墨塗り・削除署名の概要と問題点を述べる. そして 4 節にて Aggregate 署名の種類数を 1 種類, 2 種類に限定した場合を議論し, 5 節にて全ての部分文書状態が実現される墨塗り・署名方式を提案する.

2. Aggregate 署名

本節では, Boneh らによる Aggregate 署名 [4] の概略を説明する. 宮崎らによる削除署名 [2], 佐野らによる墨塗り・削除署名 [3], 本稿で提案する墨塗り・削除署名は, すべて Boneh らの Aggregate 署名を利用している.

Aggregate 署名は, 複数の署名者が各自の文書に対して生成した署名 (個別署名) を同サイズの 1 つの署名 (Aggregate 署名) に集約する技術である. 検証者は Aggregate 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. 以下, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ を位数 p の乗法群とし, g_1, g_2 を $\mathbb{G}_1$, $\mathbb{G}_2$ の生成元とする. また $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ を双線形写像とする. さらに $H : \{0,1\}^* \to \mathbb{G}_1$ を理想的なハッシュ関数 (ランダムオラクル) とする. このとき Boneh らの Aggregate 署名は以下の 4 つのアルゴリズム KeyGen (鍵生成), Sign (署名生成), Agg (Aggregate 署名生成), AggVerify (Aggregate 署名検証) から構成される:

◇ KeyGen (of the i-th signer, $i = 1, \ldots, n$)
生成された乱数 $\mathsf{sk}_i \in \mathbb{Z}/p\mathbb{Z}$ に対して $\mathsf{pk}_i \leftarrow g_2^{\mathsf{sk}_i}$ を求め, 秘密鍵・秘密鍵ペア $(\mathsf{sk}_i, \mathsf{pk}_i) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{G}_2$ を出力する.

◇ Sign (of the i-th signer, $i = 1, \ldots, n$)
秘密鍵 $\mathsf{sk}_i \in \mathbb{Z}/p\mathbb{Z}$, 文書 $M_i \in \{0,1\}^*$ に対して $\sigma_i \leftarrow H(M_i)^{\mathsf{sk}_i}$ を求め, 個別署名 $\sigma_i \in \mathbb{G}_1$ を出力する.

◇ Agg
個別署名 $\sigma_1, \ldots, \sigma_n \in \mathbb{G}_1$ に対して $\sigma \leftarrow \sigma_1 \times \cdots \times \sigma_n$ を求め, Aggregate 署名 $\sigma \in \mathbb{G}_1$ を出力する.

◇ AggVerify
文書 $M_1, \ldots, M_n \in \{0,1\}^*$, 公開鍵 $\mathsf{pk}_1, \ldots, \mathsf{pk}_n \in \mathbb{G}_2$, Aggregate 署名 $\sigma \in \mathbb{G}_1$ に対し, $e(\sigma, g_2) = \prod_{i=1}^{n} e(H(M_i), \mathsf{pk}_i)$ が成立するかを検証する.

Agg の記述では n 個の個別署名を集約する場合を述べたが, 簡単な変更により, 任意の個別署名 $\sigma_{i_1}, \ldots, \sigma_{i_k}$ と, 独立な Aggregate 署名 σ とを集約することも可能である. また, 以下の墨塗り・削除署名の構成では, 個別署名 $\sigma_1, \ldots, \sigma_n$ から生成された Aggregate 署名 σ から, 個別署名 σ_i を容易に除去できるという性質を利用している: $\sigma \leftarrow \sigma/\sigma_i$ とすれば良い.

Aggregate 署名の安全性は, 群ペア $(\mathbb{G}_1, \mathbb{G}_2)$ における co-GDH 仮定に基づく [4]. ここで co-GDH 仮定とは, co-DDH 問題は簡単だが co-CDH 問題は困難という仮定である. ただし群ペア $(\mathbb{G}_1, \mathbb{G}_2)$ における co-DDH 問題とは g, $g^a \in \mathbb{G}_1$, h, $h^b \in \mathbb{G}_2$ から $a = b$ かを判定する問題, co-CDH 問題とは g, $g^a \in \mathbb{G}_1$, $h \in \mathbb{G}_2$ から $h^a \in \mathbb{G}_2$ を求める問題である. $\mathbb{G}_1 \times \mathbb{G}_2$ でペアリング e が定義されている場合, 群ペア $(\mathbb{G}_1, \mathbb{G}_2)$ における co-DDH 問題は容易である.

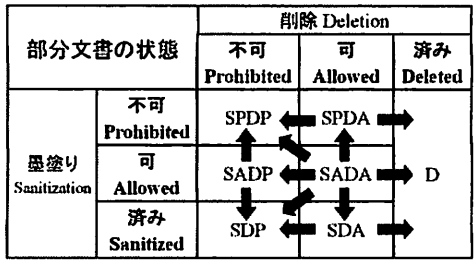
3. 佐野らによる墨塗り・署名方式

墨塗り署名と削除署名を統合する技術として, 佐野らは墨塗り・削除署名を提案するとともに, 2 種類の具体的な方式 (SIKOT-I, SIKOT-II) を示した. 本節はこれら方式の概要を説明する.

墨塗り・削除署名 SIKOT-I, SIKOT-II は, いずれも KeyGen (鍵生成), Sign (署名生成), Revise (改訂), Verify (検証) の 4 つのアルゴリズムから構成される. 鍵生成者は KeyGen を用いて署名者と各改訂者の秘密鍵・公開鍵ペアを生成する. 署名者は Sign を用いてオリジナル文書に対する署名を生成する. 改訂者は Revise を用いて部分情報の秘匿 (墨塗り・削除) を行い (本稿では複数人の改訂者を想定する), 検証者は Verify を用いて改訂文書の開示部分の完全性と秘匿の正当性を確認する. 墨塗り・削除署名の安全性として, 検証者は墨塗り部分からオリジナル文書に対する情報が得られないこと, また, どの部分文書が削除されたか, あるいは何箇所の部分文書が削除されたかを識別できないことが必要であり, SIKOT-I, SIKOT-II はこれら安全性を満たしている [3].

部分文書の墨塗り・削除機能に加え, 佐野らの墨塗り・削除署名 SIKOT-I, SIKOT-II は, 各部分文書に対して墨塗り・削除の制御することが可能である. この制御を行うために, 各部分文書には以下のような部分文書状態のいずれかが付与されている:

- SADA (Sanitization is Allowed, Deletion is Allowed): 現在は開示, 将来の墨塗りも削除も可能
- SADP (Sanitization is Allowed, Deletion is Prohibited): 現在は開示, 将来の墨塗りは可能だが削除は禁止
- SPDA (Sanitization is Prohibited, Deletion is Allowed): 現在は開示, 将来の墨塗りは禁止だが削除は可能
- SPDP (Sanitization is Prohibited, Deletion is Prohibited): 現在は開示, 将来の墨塗りも削除も禁止 (強制開示)



※ 矢印以外の状態遷移は起こらない.

図 1 部分文書状態の遷移

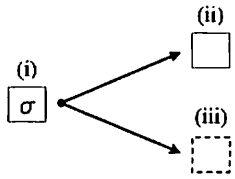


図 2 状態遷移のルール

- SDA (Sanitized, Deletion is Allowed): 現在は墨塗り, 将来の削除は可能
- SDP (Sanitized, Deletion is Prohibited): 現在は墨塗り, 将来の削除は禁止
- D (Deleted): 現在は削除

SIKOT-I, SIKOT-II はこれら部分文書状態の一部と有向的な状態遷移を実現することで, 部分文書状態を制御している. SIKOT-I と SIKOT-II の大きな違いは実現可能な部分文書状態の個数であり, SIKOT-I は SADA, SPDP, SDA, D の 4 状態を実現可能であるのに対し, SIKOT-II は SADA, SPDP, SPDA, SDA, SDP, D の 6 状態を実現可能となっている. このような状態を表現するために, SIKOT-I は 1 種類の Aggregate 署名を, SIKOT-II は 2 種類の Aggregate 署名を利用している.

4. Aggregate 署名の種類数と部分文書状態

佐野らによる墨塗り・削除署名 (SIKOT-I, SIKOT-II) では一部の部分文書状態が実現されていたが, SPDA という文書状態は実現できていなかった. また, Aggregate 署名を 1 種類または 2 種類に限定した場合に, 同様な墨塗り・削除署名が実現可能かも検討されていなかった. そこで本節では佐野らの墨塗り・削除署名を解析し, (本質的に) 別個の墨塗り・削除署名の実現可能性について検討する. Aggregate 署名を 1 種類用いた場合には, SIKOT-I とは (本質的に) 異なる墨塗り・削除署名が構成できることを示す. また Aggregate 署名を 2 種類用いた場合に, 全ての部分文書状態を実現する墨塗り・削除署名が構成できることを示す.

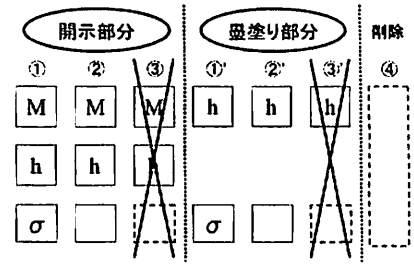


図 3 Aggregate 署名を 1 種類用いた場合に表現可能な全状態

4.1 アプローチ

本稿の最終的な目標は, 全ての部分文書状態を実現する墨塗り・削除署名を構成することである. このとき部分文書状態の可能な遷移を限定する必要があるため, 各部分文書状態の性質を考慮し, 以下では図 1 の状態遷移に場合に限定して考える (逆に図 1 の矢印以外の状態遷移は考えない). この状態遷移は, 佐野らの墨塗り・削除署名における状態遷移を自然に拡張したものとなっている.

次に部分文書状態の表現方法について考える. i 番目の部分文書に対する部分文書状態は, 部分文書 m_i, 個別署名 σ_i, Aggregate 署名 σ の保持方法と内容を用いて表現することになるため, 次のように図示して考える (図 2 参照):

- 個別署名 σ_i が実線で枠組みされているとき, 個別署名 σ_i は保持され, Aggregate 署名 σ の中に含まれているとする. → (*i*)
- 実線枠のみのとき (あるいは何も示されていないとき), 個別署名 σ_i は保持されていないが, Aggregate 署名 σ の中に含まれているとする. → (*ii*)
- 点線枠のみのときは個別署名 σ_i は保持されておらず, Aggregate 署名 σ の中にも含まれていないとする. → (*iii*)

以下, Aggregate 署名を 1 種類用いる場合と 2 種類用いる場合に分けて考察する.

4.2 Aggregate 署名が 1 種類の場合

Aggregate 署名を 1 種類用いた場合に表現できる状態は, 図 3 の 7 種類である. ここで状態 ③ と状態 ③' は部分文書に対する個別署名を保持しておらず, また個別署名が Aggregate 署名にも含まれていないため, 部分文書を検証することができない. つまりこれらの状態に部分文書状態を付与することは適切とはいえず, 除外する必要がある. 従って, Aggregate 署名が 1 種類の場合に表現できる状態数は最大でも 5 種類となり, 全ての部分文書状態を表現することは不可能である.

まず, 実現が容易な SADA と D の対応付けを考える. SADA からは全ての部分文書状態への遷移が必要となるため, 表現できる状態数を最大にするには, SADA を状態 ① に対応づける必要がある. 他方 D では全ての情報が削除されるので, 状態 ④ に対応づける必要がある.

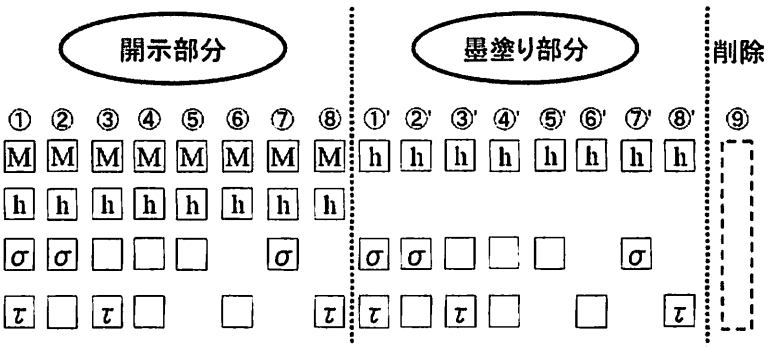


図 4　Aggregate 署名を 2 種類用いた場合に表現可能な全状態

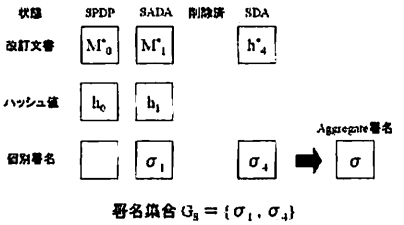


図 5　Aggregate 署名を 1 個用いた墨塗り・削除署名 (パターン 1-1)

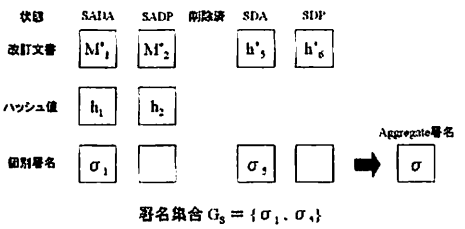


図 6　Aggregate 署名を 1 個用いた墨塗り・削除署名 (パターン 1-2)

次に DP (SPDP, SADP, SDP) について考える. DP からは D への遷移, つまり全情報の削除を禁止する必要があるため,

状態 ② または状態 ②' に対応づけなければならない (逆に状態 ②, ②' はこれらの部分文書状態しか表現できない). ここで状態 ② は状態 ②' に遷移可能なので, ② は (対応づけるならば) SADP となる. 従って, 状態 ②' は SPDP または SDP にしか対応づけられない.

他方で状態 ①' は, DP の場合の考察と部分文書がハッシュ値になっていることから, (対応づけるならば) SDA となる.

以上の議論より, Aggregate 署名を 1 種類用いた場合, 2 パターンの墨塗り・削除署名を構成することができる. 1 つ目のパターン [SADA, SPDP, SDA, D] を図 5 に, 2 つ目のパターン [SADA, SADP, SDA, SDP, D] を図 6 に示す.

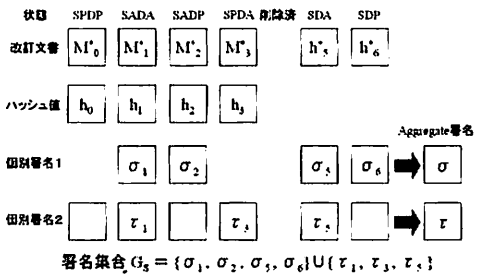


図 7　Aggregate 署名を 2 個用いた墨塗り・削除署名 (パターン 2-1)

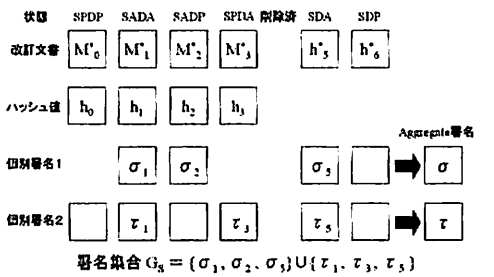


図 8　Aggregate 署名を 2 個用いた墨塗り・削除署名 (パターン 2-2)

ここでパターン 1-1 (図 5) は佐野らによる墨塗り・削除署名 SIKOT-I に一致しているが, パターン 1-2 (図 6) はこれとは本質的に異なっており, 別個の構成となっている. パターン 1-1 (図 5) とパターン 1-2 (図 6) を比べた場合, SADA, D, SDA は共通で実現されているが, それ以外の部分文書状態が異なっている. すなわち, パターン 1-1 は墨塗り禁止かつ削除禁止を設定ときに用いればよく, パターン 1-2 は墨塗り禁止を必要とせず開示部分と墨塗り部分にそれぞれ削除禁止を設定したいときに用いればよい.

4.3 Aggregate 署名が 2 種類の場合

Aggregate 署名を 2 種類用いた場合に表現できる状態は, 図 4 の 17 種類である. ただし Aggregate 署名が 1 種類の場合と同様に検証が不可能となる場合は除いている.

まず Aggregate 署名が 1 個の場合と同様に, SADA は状態 ① に, D は状態 ⑨ に対応づけて考える. 次に DP は状態 ②〜⑥ かつ ②'〜⑥' に対応づくことになるが, さらなる考察により

- SADP, SPDP は状態 ②〜⑥ のいずれかに限定される. これら状態間では状態 ② から状態 ④,⑥ または ③ から状態 ④,⑥ への遷移が可能なので, SADP は状態 ② または ③ に対応づけることになる. しかし, ② と ③ は σ, τ を入れ替えたものなので SADP は状態 ② に対応づける. これにより, SPDP は状態 ④,⑥ のどちらかに限定される.
- SDP は墨塗り済みのため, 状態 ②'〜⑥' のいずれかに限定されるが, SADP からの遷移を考えると状態 ②', ④', ⑥' に絞られる.

SPDA は状態 ①, ⑦, ⑧ のいずれかに限定されるが, 状態 ① は SADA に対応づけられているので, 状態 ⑦, ⑧ のどちらかに限定される. ここで, SPDA から SPDP への遷移を考えると, SPDP は状態 ⑥, SPDA は状態 ⑧ に対応づけられるので SDP は状態 ②', ④' のどちらでもよい. このとき, SDA は SDP への遷移を考えると状態 ①' に対応づけられる.

以上の議論により, Aggregate 署名を 2 種類用いた場合, SDP の候補が 2 個あるので 2 パターンの墨塗り・削除署名を構成することができる. 1 つ目のパターンを図 7 に, 2 つ目のパターンを図 8 に示す. いずれも 7 種類の部分文書状態を全て実現できており, 佐野らの墨塗り・削除方式とは異なる構成となっている. パターン 2-1 (図 7) とパターン 2-2 (図 8) は SDP の表現方法のみが異なっているが, Aggregate 署名 τ に関する表現を追加したという意味で, いずれもパターン 1-1 の自然な拡張となっている. また, パターン 2-2 (図 8) は SDP の個別署名 σ_i が保持されないのでパターン 2-1 (図 7) よりも署名長が短くなり、優れている。5 節でパターン 2-2 (図 8) を提案方式として説明する.

5. 提案墨塗り・削除署名

本節では, 全ての部分文書状態が実現可能な墨塗り・削除署名方式を提案する. 提案方式は前節で示したパターン 2-2 に基づいている. 以下では, 署名の対象となる文書は n 個の部分文書 $M_i \in \{0,1\}^*$ から構成される文書列 $(M_1, \ldots, M_n)$ として考える. また, 個別署名 σ_i が Aggregate 署名 σ に含まれていることを $\sigma_i \sqsubset \sigma$ とかくことにする.

5.1 方式の概要

前節で述べた通り, 状態表現は図 8 のようになっている. これに対する提案墨塗り・署名方式の処理を図 9 に示す.

まず, 本方式は KeyGen, Sign, Sanitize, Verify の 4 つのアルゴリズムから構成される. ここで署名集合は σ_i, (および τ_i) に対する σ, (および τ) からの除去の可否を決めるもので, ここに保持されていないものは $\sigma_i \not\sqsubset \sigma$(および $\tau_i \not\sqsubset \tau$) にすることができない. また, 添字集合 G_I は各部分文書の状態を登録するものである. 部分文書 M_i は文書識別子 ID と部分文書識別子 ID_i により $M_i^* \leftarrow ID||ID_i||M_i$ とコーディングされ, 個別署名はそれぞれ $\sigma_i \leftarrow H(ID||ID_i||H(M_i^*))^{sk}$, $\tau_i \leftarrow H(ID||ID_i||H(M_i^*))^{sk}$ と Aggregate 署名 $\sigma \leftarrow \prod_{i=0}^{n} \sigma_i$, $\tau \leftarrow \prod_{i=0}^{n} \tau_i$ が生成される. 部分文書の順序交換を防止する目的で, 文書識別子 ID と部分文書識別子 ID_i を用いているが, これは伊豆らの墨塗り署名方式の技術を流用している [5]. また, 2 種類の Aggregate 署名を生成するために公開パラメータ c を用いているが, これは同じ部分文書から異なる 2 種類の署名を求めるために導入した.

次に, 墨塗り・削除処理について説明する. 改訂者が, 墨塗り可能な箇所 (つまり $i \in SADA \cup SADP$ である部分文書 M_i^*) に対する墨塗りする場合を考える. 墨塗りする場合には, 部分文書を $M_i^* \leftarrow ID||ID_i||H(M_i^*)$ により更新し, 必要に応じて署名集合から個別署名を除去する. また, 削除可能な箇所 (つまり $i \in SADA \cup SPDA \cup SDA$ である部分文書 M_i^*) に対する削除する場合も考える. 削除する場合には, 部分文書, 各個別署名を削除し, 署名集合から各個別署名を除去することで行う. 検証者は文書識別子と部分文書識別子の正当性を確認した上で, 各 Aggregate 署名の検証を通じて文書の正当性を確認する.

ここで, 提案方式が図 1 の状態遷移 (以下, 正遷移とする) 以外は遷移しないことを示す. まず, 個別署名の変化は図 2 以外は行えない. すなわち, 正遷移以外の遷移を考えるとき改訂者は必ず個別署名を復元しなければならない. なぜなら個別署名は署名者の秘密鍵を用いて生成しているので, 署名者以外が個別署名を復元することができないからである. したがって, 正遷移以外は遷移しない.

最後に, SP (SPDP, SPDA) において墨塗りできない理由と DP (SPDP, SADP, SDP) において削除できない理由を図 8 を用いて説明する.

[SP (SPDP, SPDA) において墨塗りできない理由]
墨塗り部分は $\sigma_i \sqsubset \sigma$ であるのに対して, SP は $\sigma_i \not\sqsubset \sigma$ である. よって, SP に対して強制的に墨塗りを行うと検証に必要な σ_i を持っておらず σ の検証の際に invalid が出力されるので墨塗りが行えない.

[DP (SPDP, SADP, SDP) において削除できない理由]
DP は少なくとも 1 個は Aggregate 署名には含まれているが署名集合には保持されていない個別署名があり, 改訂者は Aggregate 署名からこの個別署名を削除できない. したがって, DP に対して強制的に削除を行うと余分な個別署名が Aggregate 署名に含まれることになり, σ(または τ) の検証の際に invalid が出力されるので削除が行えない.

5.2 安 全 性

提案方式は次の安全性を満たす.

- 個別署名の偽造不可能性
- 墨塗り処理の一方向性

(個別署名の偽造不可能性)
個別署名の偽造不可能性とは攻撃者が $\sigma_i \not\sqsubset \sigma$(または $\tau_i \not\sqsubset \tau$) となっているか, もしくは, 署名集合から除去されている σ_i(

表 1 提案方式と既存方式の比較

	Sanitization 機能				効率性	
	墨塗りのみ (SADP)	削除のみ (SPDA)	墨塗りまたは削除 (SADA)	強制開示 (SPDP)	個別署名生成回数	Aggregate 署名生成回数
SIT [6]	○	×	×	○	$n+1$	1
MHI [2]	×	○	×	○	n	1
SIKOT-I [3]	×	×	○	○	n	1
SIKOT-II [3]	○	×	○	○	$2n$	2
パターン 1-1 (図 5)	×	×	○	○	n	1
パターン 1-2 (図 6)	○	×	○	×	n	1
パターン 2-1 (図 7)	○	○	○	○	$2n$	2
提案方式 (パターン 2-2, 図 8)	○	○	○	○	$\mathbf{2n}$	**2**

または τ_i) を偽造または復元できないことを意味する．個別署名が偽造されると攻撃者が部分文書を任意の文書に置き換えることができ，削除された個別署名が復元されると前述の正遷移以外の遷移が行えてしまう．提案方式は co-GDH 仮定 (2 節参照) の下で個別署名の偽造不可能性を満たしている．

co-GDH 仮定のとき co-CDH 問題を解くのが困難であるので個別署名を偽造することも，削除された個別署名を復元することもできない．これは個別署名を偽造する場合，攻撃者が設定した部分文書 $\bar{M}_i^*$ のハッシュ値を $\bar{h}_i$ とすると，$\bar{h}_i \in \mathbb{G}_1$, $g_2, g_2^{sk}(=pk) \in \mathbb{G}_2$ から $\bar{h}_i^{sk}(=\bar{\sigma}_i) \in \mathbb{G}_1$ を求めることが困難であることを意味する．したがって，文書・署名ペア $(\bar{M}_i^*, \bar{\sigma}_i)$ を偽造することができない．また元の部分文書 M_i^* があり σ_i が削除されている場合，$h_i \in \mathbb{G}_1$, $g_2, g_2^{sk}(=pk) \in \mathbb{G}_2$ から $h_i^{sk}(=\sigma_i) \in \mathbb{G}_1$ を求めることが困難であることを意味している．したがって，元の部分文書 M_i^* から σ_i を復元することができない．

(墨塗り処理の一方向性)

墨塗り処理の一方向性とは攻撃者が墨塗り部分から元の部分文書を復元できないことを意味する．これは本方式が墨塗り部分をハッシュ値で表現しているので，ハッシュ関数の一方向性より，元の部分文書は復元できない．

5.3 比　　較

提案墨塗り・削除署名方式の性能を示すために，鈴木らによる墨塗り署名 (SIT [6]), 宮崎らによる削除署名 (MHI [2]), 佐野らによる墨塗り・削除署名 SIKOT-I, SIKOT-II [3], および提案方式の機能 (設定できる状態), 効率 (署名生成回数) に関する比較結果を表 1 に示す．ここでは本稿で示した類似方式 パターン 1-1 (図 5), パターン 1-2 (図 6), パターン 2-1 (図 7) との比較も行った．

表 1 からわかる通り，全ての部分文書状態を実現できているのはパターン 2-1, 提案方式 (パターン 2-2) だけであるが，SIT, MHI, SIKOT-I, パターン 1-1, パターン 1-2 は 1 種類の Aggregate 署名しか使用しない．その一方で SIKOT-II, パターン 2-1, 提案方式 (パターン 2-2) は 2 種類の Aggregate 署名を用いているため，個別署名と Aggregate 署名の生成回数が 2 倍となり，署名生成時，検証時の効率が低下している．

次に，何故 SIKOT-II よりもパターン 2-1, 提案方式 (パターン 2-2) の方が設定できる状態数が増えたのかを考える．その要因は墨塗り処理の違いにある．墨塗り処理の際に部分文書をハッシュ値に置き換えることは各方式で共通だが，SIKOT-II は個別署名を 1 個削除しているのに対して，パターン 2-1, 提案方式 (パターン 2-2) は個別署名を削除しない．これによりパターン 2-1, 提案方式 (パターン 2-2) は図 4 において SDA および SDP の候補数が SIKOT-II よりも増える．すなわち，SIKOT-II では SDA および SDP の候補が状態 ⑤'〜⑧' の 4 個であるのに対して，パターン 2-1, 提案方式 (パターン 2-2) は状態 ①'〜⑧' の 8 個となる．よって，パターン 2-1, 提案方式 (パターン 2-2) の設定できる状態数が増えた．

6. ま と め

墨塗り・削除署名において，全ての部分文書状態の表現と遷移を制御可能な新しい墨塗り・署名方式を提案した．提案方式は Aggregate 署名を 2 種類用いているが，1 種類では構成が不可能なこと，2 種類では同様な構成が可能であることも述べた．

今後は本方式に対する証明可能安全性について考えることが課題である．

文　　献

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

[2] K. Miyazaki, G. Hanaoka, H. Imai, "Digitally Signed Document Sanitizing Scheme from Bilinear Maps", *SCIS 2005*, 3E3-5, pp.1471-1476, 2005.

[3] 佐野, 伊豆, 國廣, 太田, 武仲, "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.

[4] D. Boneh, C. Gentry, B. Lynn, H. Shacham, "Aggregate and Verifiably Encrypted Signatures form Bilinear Maps", *EUROCRYPT 2003*, LNCS 2656, pp.416-432, Springer, 2003.

[5] 伊豆, 佐野, 國廣, 太田, 武仲, "Aggregate 署名を用いた墨塗り署名方式", *SCIS 2007*, 2007.

[6] M. Suzuki, T. Isshiki, K. Tanaka, "Sanitizable Signature with Secret Information", *SCIS 2006*, 4A1-2, pp.273, 2006.

◇ KeyGen

入力：群 $\mathbf{G}_2$ の生成元 g_2, 位数 p

（1） $\mathsf{sk} \leftarrow \mathbf{Z}/p\mathbf{Z}$ をランダムに生成し, $\mathsf{pk} \leftarrow g_2^{\mathsf{sk}}$ とする.

出力：署名者の秘密鍵・公開鍵ペア $(\mathsf{sk}, \mathsf{pk}) \in \mathbf{Z}/p\mathbf{Z} \times \mathbf{G}_2$

◇ Sign

入力：文書列 $(M_1, \ldots, M_n)$, 署名者の秘密鍵 sk

（1） 文書識別子 ID と部分文書識別子 ID_i をランダムに生成し, $M_0^* \leftarrow ID$, $M_i^* \leftarrow ID||ID_i||M_i$ $(i = 1, \ldots, n)$ とする. ただし ID_i は昇順となるように設定する.

（2） 各部分文書 M_i^* $(i = 0, \ldots, n)$ に対し, $h_i \leftarrow H(M_i^*)$, 個別署名 $\sigma_i \leftarrow H(ID||ID_i||h_i||0^c)^{\mathsf{sk}}$, $\tau_i \leftarrow H(ID||ID_i||h_i||1^c)^{\mathsf{sk}}$ を求める. ただし, 0^c とは全て 0 の c ビット列を表す.

（3） Aggregate 署名 $\sigma \leftarrow \prod_{i=0}^{n} \sigma_i$, $\tau \leftarrow \prod_{i=0}^{n} \tau_i$ を求める.

（4） 添字集合 $SADA \leftarrow \{1, \ldots, n\}$, $SADP \leftarrow \phi$, $SPDA \leftarrow \phi$, $SPDP \leftarrow \{0\}$, $SDA \leftarrow \phi$, $SDP \leftarrow \phi$ を設定し, 状態 $G_I = \{SADA, SADP, SPDA, SPDP, SDA, SDP\}$ とする.

（5） 署名集合 $G_S \leftarrow \{\sigma_i | i \in SADA \cup SADP \cup SDA\} \cup \{\tau_i | i \in SADA \cup SPDA \cup SDA\}$ を設定する.

出力：文書列 $(M_0^*, \ldots, M_n^*)$, 署名集合 G_S, Aggregate 署名 σ, τ, 状態 G_I

◇ Verify

入力：文書列 $(M_0^*, \ldots, M_k^*)$, Aggregate 署名 σ, τ, 状態 G_I, 署名者の公開鍵 pk

（1） 文書 ID の検証

各部分文書 M_i^* の文書識別子が M_0^* に等しいかを検証する. 失敗した場合は invalid を出力して終了する.

（2） 部分文書 ID の検証

各部分文書 M_i^* の部分文書識別子 ID_i が昇順になっているかを検証する. 失敗した場合は invalid を出力して終了する.

（3） Aggregate 署名 σ の正当性検証

$i \in SADA \cup SADP$ を満たす部分文書 M_i^* について, $x_i \leftarrow H(ID||ID_i||H(M_i^*)||0^c)$ を求める.

また, $i \in SDA \cup SDP$ を満たす部分文書 $M_i^*(= h_i^*)$ について, $x_i \leftarrow H(h_i||0^c)$ を求める. 次に $x \leftarrow \prod x_i$ を求め, $e(\sigma, g_2) = e(x, \mathsf{pk})$ が成立するかを検証する. 失敗した場合は invalid を出力して終了する.

（4） Aggregate 署名 τ の正当性検証

開示されている（$i \in SADA \cup SADP \cup SPDA \cup SPDP$ を満たす）部分文書 M_i^* について, $y_i \leftarrow H(ID||ID_i||H(M_i^*)||1^c)$ を求める.

墨塗りされている（$i \in SDA \cup SDP$ を満たす）部分文書 $M_i^*(= h_i^*)$ について, $y_i \leftarrow H(h_i||1^c)$ を求める. 次に $y \leftarrow \prod y_i$ を求め, $e(\tau, g_2) = e(y, \mathsf{pk})$ が成立するかを検証する. 失敗した場合は invalid を出力して終了する.

（5） valid を出力して終了する.

◇ Revise

入力：文書列 $(M_0^*, \ldots, M_k^*)$, 署名集合 G_S, Aggregate 署名 σ, τ, 状態 G_I

（1） 各部分文書 M_i^* $(i = 1, \ldots, k)$ の新しい状態を設定する. 状態は変更しなくても良い.

- $i \in SADA$ である部分文書 M_i^* の強制開示
  Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$ に, 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$, $\{\tau_i\}$ に, 添字集合を $SADA \leftarrow SADA \backslash \{i\}$, $SPDP \leftarrow SPDP \cup \{i\}$ に更新する.
- 部分文書 M_i^* の状態を $SADA \rightarrow SPDA$ に変更
  Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$ に, 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$ に, 添字集合を $SADA \leftarrow SADA \backslash \{i\}$, $SPDA \leftarrow SPDA \cup \{i\}$ に更新する.
- 部分文書 M_i^* の状態を $SADA \rightarrow SADP$ に変更
  署名集合を $G_S \leftarrow G_S \backslash \{\tau_i\}$ に, 添字集合を $SADA \leftarrow SADA \backslash \{i\}$, $SADP \leftarrow SADP \cup \{i\}$ に更新する.
- $i \in SADA$ である部分文書 M_i^* の墨塗り（SDA）
  部分文書を $M_i^* \leftarrow ID||ID_i||H(M_i^*)$ に更新する. 添字集合を $SADA \leftarrow SADA \backslash \{i\}$, $SDA \leftarrow SDA \cup \{i\}$ に更新する.
- 部分文書 M_i^* の状態を $SADA \rightarrow SDP$ に変更
  部分文書を $M_i^* \leftarrow ID||ID_i||H(M_i^*)$ に更新する. 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$, $\{\tau_i\}$ に, 添字集合を $SADA \leftarrow SADA \backslash \{i\}$, $SDP \leftarrow SDP \cup \{i\}$ に更新する.
- $i \in SADA$ である部分文書 M_i^* の削除
  部分文書 M_i^* を削除し, Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$, $\tau \leftarrow \tau/\tau_i$ に更新する. 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$, $\{\tau_i\}$ に更新する. 添字集合を $SADA \leftarrow SADA \backslash \{i\}$ に更新する.
- $i \in SPDA$ である部分文書 M_i^* の削除
  部分文書 M_i^* を削除し, Aggregate 署名を $\tau \leftarrow \tau/\tau_i$ に, 署名集合を $G_S \leftarrow G_S \backslash \{\tau_i\}$ に, 添字集合を $SPDA \leftarrow SPDA \backslash \{i\}$ に更新する.
- $i \in SPDA$ である部分文書 M_i^* の強制開示
  署名集合を $G_S \leftarrow G_S \backslash \{\tau_i\}$ に更新する. 添字集合を $SPDA \leftarrow SPDA \backslash \{i\}$, $SPDP \leftarrow SPDP \cup \{i\}$ に更新する.
- $i \in SADP$ である部分文書 M_i^* の強制開示
  Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$ に, 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$ に更新する. 添字集合を $SADP \leftarrow SADP \backslash \{i\}$, $SPDP \leftarrow SPDP \cup \{i\}$ に更新する.
- $i \in SADP$ である部分文書 M_i^* の墨塗り（SDP）
  部分文書を $M_i^* \leftarrow ID||ID_i||H(M_i^*)$ に更新する. 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$ に更新する. 添字集合を $SADP \leftarrow SADP \backslash \{i\}$, $SDP \leftarrow SDP \cup \{i\}$ に更新する.
- 部分文書 M_i^* の状態を $SDA \rightarrow SDP$ に変更
  署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$, $\{\tau_i\}$ に更新する. 添字集合を $SDA \leftarrow SDA \backslash \{i\}$, $SDP \leftarrow SDP \cup \{i\}$ に更新する.
- $i \in SDA$ である部分文書 M_i^* の削除
  部分文書 M_i^* を削除し, Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$, $\tau \leftarrow \tau/\tau_i$ に更新する. 署名集合を $G_S \leftarrow G_S \backslash \{\sigma_i\}$, $\{\tau_i\}$ に更新する. 添字集合を $SDA \leftarrow SDA \backslash \{i\}$ に更新する.

（2） 添字番号をリナンバリングする.

出力：文書列 $(M_0^*, \ldots, M_{k'}^*)$, 署名集合 G_S, Aggregate 署名 σ, τ, 状態 G_I

図 9 アルゴリズム