東海大学大学院2007年度修士論文

RSA Accumulatorを用いた開示条件付き墨塗り署名スキームに関する研究

RSA Accumulator based
Sanitizable Signature Scheme
with Condition to Disclose

指導教員 菊池 浩明 教授

東海大学 大学院 工学研究科
情報理工学専攻

6ADRM006 上山 真梨

目 次

記号一覧

m	: 部分文書
r, b	: 乱数
α, γ	: コミットメント
α^*, γ^*	: コミットメントの最終状態
M	: 条件　開示（強制開示も墨塗りもされていない状態）
S	: 条件　墨塗り
D	: 条件　強制開示
M^*, S^*, D^*	: 条件の最終状態
sk, pk	: 署名者の秘密鍵，公開鍵
p, q	: 素数
N	: RSA の公開鍵　$N = p \cdot q$
$\mathrm{Sign}_{sk}(\cdot)$	: 署名生成アルゴリズム
$\mathrm{Verify}_{pk}(\cdot)$	: 署名検証アルゴリズム
L	: 素数の集合
a, g	: 定数　$a, g \in Z_N$
δ	: 定数
$H(\cdot)$	: ハッシュ関数
h	: ハッシュ値
i	: インデックス番号
σ	: 署名
f	: 直線
e, c	: 素数
n, k, ℓ	: 総部分文書数，総墨塗り部分文書数，総強制開示部分文書数
$PK(\cdot)$	: 知識の証明
t_1, t_2, s	: ゼロ知識証明の検証値
A, B	: Accumulator
$commit$	: コミットメント
$Com(\cdot)$	: コミットメントスキーム
$CEAS$	: アクセス構造

第1章
序論

1.1 研究の背景と目的

近年，コスト削減や作業効率アップのため，文書の電子化が進んでいる．電子署名を用いることで，文書の作成者を証明し，改ざんされていないことを保証することができる．しかし，電子署名を施した文書を公開する際，プライバシーに関する部分などを秘匿（墨塗り）する必要性が出てきた．だが，電子署名の性質上，文書の一部を秘匿してしまうと，改ざんとみなされ検証出来ない．この問題を解決する方法として墨塗り署名が提案されている．

墨塗り署名は，署名された文書の一部分を墨塗り（秘匿）した後でも，開示部分の完全性を保証する署名方式である．墨塗り署名には，さまざまな方式が提案されている．開示条件を設定可能な方式には，多項式補間を用いた SUMI-5[1] や Aggregate 署名を用いた SUMI-6[2] がある．総部分文書数 n と墨塗り部分文書数 k に対して，$\mathcal{O}(n)$ の署名長が必要になる SUMI-5[1] に対して，Aggregate 署名を用いた SUMI-6[2] では $\mathcal{O}(n-k)$ の長さになり通信効率がよい．しかし，Aggregate 署名は楕円曲線上の双線形写像を実現する必要があり，現時点では通常の署名の数十倍の計算コストがかかると言われている．

そこで，本研究では，宮崎らが提案した SUMI-5[1] をベースに複数の素数を単一の証拠 (wittness) で効率よく証明可能な RSA Accumulator[3] を用いた開示条件付き墨塗り署名方式を提案する．この方式は Aggregate 署名（双線形写像）を用いることなく，SUMI-6[2] と同等の $\mathcal{O}(n-k)$ のサイズの効率的な署名長を実現する．

1.2 論文の構成

本論文の構成は，全 7 章の構成になっている．第 1 章では，序論として，現状の電子署名における問題点と解決方法である墨塗り署名の概要について述べる．そして，本研究における目的を述べる．第 2 章では，関連研究としていくつかの墨塗り署名方式について述べる．第 3 章では，提案方式における要素技術である RSA Accumulator と Lagrange の補間公式について述べる．第 4 章，第 5 章，第 6 章では，提案方式について述べる．第 4 章では，RSA Accumulator を用いた墨塗り署名について述べる．第 5 章では，RSA Accumulator を用いた開示条件付き墨塗り署名として 2 つの方式を提案し，それぞれの方式に対する問題点を挙

げる．第 6 章では，第 5 章の方式における問題点を解決し，安全性を高めた 2 つの改良方式について述べる．最後に第 7 章にて，本研究の結論と今後の課題について述べる．

第2章
関連研究

2.1 墨塗り署名

墨塗り署名は署名者，複数の墨塗り者，検証者の 3 者からなる．3 者モデルを図 2.1 に示す．署名者は，文書に自身の秘密鍵を用いて署名をする．墨塗り者は，文書の非開示をする部分を決め，墨塗り（秘匿）をする．そして検証者は，署名者の公開鍵を用い，開示部分の完全性を検証する．なお，墨塗り者は複数人いるものとする．

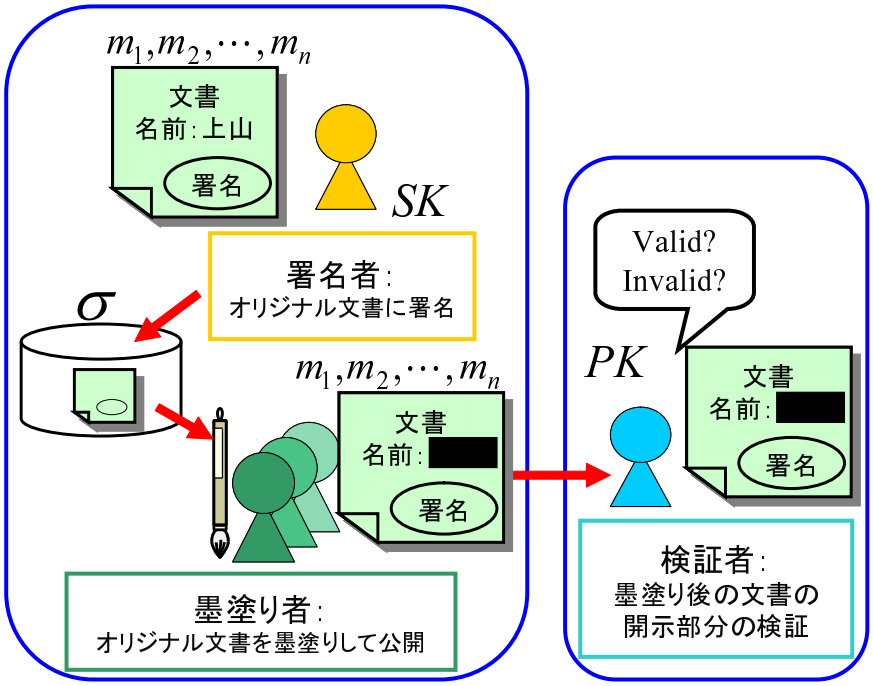

図 2.1: 3 者モデル

2.2 墨塗り署名の種類

墨塗り署名方式は様々な方式が提案されている．

Steinfeld によって提案された Content Extraction Signatures (CES)[4] は．成績証明書などの公式文書の所有者が，その文書の一部分だけを抽出して，文書の正当性を保証する技術

である．宮崎らによって提案された SUMI-4[5] は，CES における所有者を墨塗り者とみなした方式といえる．これらの方式は，i 番目の墨塗り者が，開示した部分を $i+1$ 番目以降の墨塗り者が墨塗りをするという追加墨塗りが可能である．

追加墨塗りにおける問題点を解決する方式として，宮崎らによって提案された開示条件を設定できる方式である SUMI-5[1] がある．また，追加墨塗りにおける問題点に対して，武仲らが提案した PIAT[6] は，墨塗り処理をする際に墨塗り者による署名を施すことにより，墨塗り者を特定することで解決している．

CES，SUMI-4，SUMI-5，PIAT はハッシュ値を用いた方式である．Aggregate 署名を用いた方式には宮崎らが提案した SUMI-6[2] や武仲らが提案した PIAT-A[7] がある．これらの方式は，ハッシュ値を用いた方式と比較して署名サイズを削減できる．

2.3 Steinfeld による墨塗り署名 (CES)

成績証明書などの公式文書の所有者が，その文書の一部分だけを抽出して，文書の正当性を保証する技術として Steinfeld によって提案された Content Extraction Signatures (CES)[4] がある．CES は署名者，所有者，検証者の 3 者からなる．この技術における所有者を墨塗り者とすることで，墨塗り署名方式とみなすことが出来る．特徴として，署名者が Content Extraction Access Structure(CEAS) と呼ばれる，抽出を許す部分文書の組合せを指定する情報を署名対象に入れることにより，指定された部分文書の組合せ以外の抽出を防止できることが挙げられる．CES では 4 つの方式が提案されている．ここでは CES の手順について説明する．

2.3.1 CommitVector

CES-CommitVector は，各部分文書をコミットし，そのコミットメントを連結し，署名を施す方式である．この方式は任意の署名アルゴリズムを使用することが可能である．SUMI-4[5] とほぼ同じ方式といえる．

【署名生成】

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk

Step 1: 抽出を許可する部分文書をアクセス構造 $CEAS$ を使って定める．

Step 2: 各部分文書 $m_i (i = 1, \ldots, n)$ に乱数 r_i を生成し，コミットメント $commit_i = Com(m_i||r_i)$ を求める．ここで Com はコミットメントスキームとする．

Step 3: 署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(CEAS||commit_1||\cdots||commit_n)$$

を生成する．

出力 署名 σ，部分文書 $m_1,\ldots,m_n$，乱数 $r_1,\ldots,r_n$，アクセス構造 $CEAS$，抽出インデックス集合 $M = \{1,\ldots,n\}$

【墨塗り】

入力 署名 σ，部分文書 $m_1,\ldots,m_n$，乱数 $r_1,\ldots,r_n$，アクセス構造 $CEAS$，抽出インデックス集合 M

Step 1: i 番目の部分文書を非抽出（墨塗り）する場合，コミットメント $commit_i = Com(m_i||r_i)$ を求め，部分文書 m_i と乱数 r_i を削除する．

Step 2: 抽出インデックス集合において $M \leftarrow M \setminus \{i\}$ と更新する．

出力 署名 σ，部分文書 $\{m_i|i \in M\}$，乱数 $\{r_i|i \in M\}$，コミットメント集合 $\{commit_i|i \notin M\}$，アクセス構造 $CEAS$，抽出インデックス集合 M

【署名検証】

入力 署名 σ，部分文書 $\{m_i|i \in M\}$，乱数 $\{r_i|i \in M\}$，コミットメント集合 $\{commit_i|i \notin M\}$，アクセス構造 $CEAS$，抽出インデックス集合 M，署名者の公開鍵 pk

Step 1: $i = 1,\ldots,n$ について抽出インデックス集合 M に i が含まれている場合は，i の部分文書のコミットメント $commit_i = Com(m_i||r_i)$ を求める．

Step 2: 署名者の公開鍵 pk を用いて，

$$\mathrm{Verify}_{pk}(\sigma) \stackrel{?}{=} (CEAS||commit_1||\cdots||commit_n)$$

を検証する．

Step 3: 抽出された文書がアクセス構造 $CEAS$ の要素であることを検証する．

2.3.2 HashTree

CES-HashTree は，CES-CommitVector の変形である．各部分文書のコミットメントをハッシュ関数を用い，出力されたハッシュ値を葉とした HashTree を構成する．署名者は CEAS を付加したルートのハッシュ値に署名を施す．HashTree を用いることにより CES-CommitVector よりデータサイズを削減することができる．

2.3.3 RSAProd

CES-CommitVector，CES-HashTree は任意の署名アルゴリズムを使用することが可能であったが CES-RSAProd は RSA 署名に固定される方式である．この方式は，RSA における batch（一括）検証を用いることで検証者に渡す抽出署名長を削減できる．

【署名生成】

署名者は，RSA の公開鍵 N, e，秘密鍵 d を用意する．

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 d

Step 1: 抽出を許可する部分文書をアクセス構造 $CEAS$ を使い定める．

Step 2: ランダムにタグ T を選び，部分文書 $m_i (i = 1, \ldots, n)$ に対し，ハッシュ値 $h_i = H(CEAS||T||i||m_i)$ を求める．ここで H はハッシュ関数とする．

Step 3: 署名者の秘密鍵 d を用いて，署名

$$\sigma_i = h_i^d \bmod N$$

を生成する．

出力 署名 $\sigma_1, \ldots, \sigma_n$，部分文書 $m_1, \ldots, m_n$，タグ T，アクセス構造 $CEAS$，抽出インデックス集合 $M = \{1, \ldots, n\}$

【墨塗り】

入力 署名 $\sigma_1, \ldots, \sigma_n$，部分文書 $m_1, \ldots, m_n$，タグ T，アクセス構造 $CEAS$，抽出インデックス集合 M

Step 1: 部分文書を抽出する場合，

$$\sigma_M = \Pi_{i \in M} \sigma_i \bmod N$$

を求める．

Step 2: 抽出インデックス集合において $M \leftarrow M \setminus \{i\}$ と更新する．

出力 署名 σ_M，部分文書 $\{m_i | i \in M\}$，タグ T，アクセス構造 $CEAS$，抽出インデックス集合 M

【署名検証】

入力 署名 σ_M，部分文書 $\{m_i | i \in M\}$，タグ T，アクセス構造 $CEAS$，抽出インデックス集合 M，署名者の公開鍵 e

Step 1: $i = 1, \ldots, n$ について抽出インデックス集合 M に i が含まれている場合は，ハッシュ値 $h_i = H(CEAS||T||i||m_i)$ を求める．

Step 2: 署名者の公開鍵 e を用いて，

$$\sigma_M^e \stackrel{?}{=} \Pi_{i \in M} h_i \bmod N$$

を検証する．

Step 3: 抽出された文書がアクセス構造 $CEAS$ の要素であることを検証する．

2.3.4 MERSAProd

CES-MERSAProd は CES-RSAProd の変形である．Multi-Exponent RSA における batch 処理をすることで，署名対象文書に対する署名サイズを削減する．

2.4 宮崎らによる墨塗り署名 (SUMI-4)

SUMI-4 は宮崎らによって提案された墨塗り署名方式である [5]．この方式は，各部分文書のハッシュ値の連結に署名を施す方式である．墨塗りを行う場合は，墨塗りをする部分文書をハッシュ値に置き換えて検証者に渡す．検証者は，墨塗りされていない部分文書からハッシュ値を求めたものと，墨塗りされている部分文書のハッシュ値を用いて検証する．SUMI-4 は，墨塗り者の匿名性が高いのが特徴である．ここでは SUMI-4 の手順について説明する．

2.4.1 署名生成

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk

Step 1: 部分文書 $m_i (i = 1, \ldots, n)$ に乱数 r_i を生成し，ハッシュ値 $h_i = H(m_i||r_i)$ を求める．ここで H は一方向性ハッシュ関数とする．

Step 2: 署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(h_1|| \cdots ||h_n)$$

を生成する．

出力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $r_1, \ldots, r_n$，墨塗りインデックス集合 $S = \{1, \ldots, n\}$

2.4.2 墨塗り

入力 署名 σ，部分文書 $\{m_i | i \notin S\}$，乱数 $\{r_i | i \notin S\}$，ハッシュ値集合 $\{h_i | i \in S\}$，墨塗りインデックス集合 S

Step 1: i 番目の部分文書を墨塗りする場合，ハッシュ値 $h_i = H(m_i||r_i)$ を求め，部分文書 m_i と乱数 r_i を削除する．

Step 2: 墨塗りインデックス集合 S に i を入れる．

出力 署名 σ，部分文書 $\{m_i | i \notin S\}$，乱数 $\{r_i | i \notin S\}$，ハッシュ値集合 $\{h_i | i \in S\}$，墨塗りインデックス集合 S

2.4.3 署名検証

入力 署名 σ，部分文書 $\{m_i | i \notin S\}$，乱数 $\{r_i | i \notin S\}$，ハッシュ値集合 $\{h_i | i \in S\}$，墨塗りインデックス集合 S，署名者の公開鍵 pk

Step 1: $i = 1, \ldots, n$ について墨塗りインデックス集合 S に i が含まれていない場合は，i の部分文書のハッシュ値 $h_i = H(m_i||r_i)$ を求める．

Step 2: 署名者の公開鍵 pk を用いて，

$$\mathrm{Verify}_{pk}(\sigma) \stackrel{?}{=} (h_1|| \cdots ||h_n)$$

を検証する．

2.5 宮崎らによる開示条件付き墨塗り署名（SUMI-5）

SUMI-5 は宮崎らによって提案された開示条件付き墨塗り署名方式である [1]．各部分文書に対して，開示（強制開示も墨塗りもされていない状態），墨塗り，強制開示の 3 つの開示条件を設定できる．この方式は，多項式補間を用いている．SUMI-4 と同様に墨塗り者の匿名性が高いのが特徴である．ここでは SUMI-5 の手順について説明する．

2.5.1 署名生成

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk

Step 1: 各部分文書 m_i に対し，乱数 r_i，b_i を生成し，2 点 $(1, H(b_i))$, $(2, H(m_i||r_i))$ を通る直線 f_i を求め，f_i 上の 2 点を $e_i = f_i(0)$, $c_i = f_i(3)$ とする．ここで直線 f_i は有限体上の一次多項式，H はハッシュ関数である．

Step 2: 署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(e_1||\cdots||e_n||c_1||\cdots||c_n)$$

を生成する．

出力 署名 σ，部分文書 $m_1,\ldots,m_n$，乱数 $r_1,\ldots,r_n$，$b_1,\ldots,b_n$，補助点 $c_1,\ldots,c_n$

2.5.2 墨塗り

開示条件を以下のように部分文書のインデックス集合で定める．例えば，初期状態では，$M=\{1,\ldots,n\}$, $S=D=\phi$ である．n は総部分文書数とする．

M： 開示かつ追加墨塗り可能

S： 墨塗り

D： 開示かつ追加墨塗り不可（強制開示）

入力 署名 σ，部分文書 $\{m_i|i\in M\cup D\}$，乱数 $\{r_i|i\in M\cup D\}$，$\{b_i|i\in M\cup S\}$，補助点 $c_1,\ldots,c_n$，条件 M, S, D

$i\in M$ について，条件の変化に応じ次の処理を行う．

- 墨塗り
  m_i, r_i を削除し，$M\leftarrow M\setminus\{i\}$，$S\leftarrow S\cup\{i\}$ と更新する．
- 強制開示
  b_i を削除し，$M\leftarrow M\setminus\{i\}$，$D\leftarrow D\cup\{i\}$ と更新する．

出力 署名 σ，部分文書 $\{m_i|i\in M\cup D\}$，乱数 $\{r_i|i\in M\cup D\}$，$\{b_i|i\in M\cup S\}$，補助点 $\{c_1,\ldots,c_n\}$，条件 M, S, D

墨塗りを繰り返した後，最終状態の条件を各々，M^*, S^*, D^* とおく．

2.5.3 署名検証

入力 署名 σ，部分文書 $\{m_i|i\in M^*\cup D^*\}$，乱数 $\{r_i|i\in M^*\cup D^*\}$，$\{b_i|i\in M^*\cup S^*\}$，補助点 $\{c_1,\ldots,c_n\}$，条件 M^*, S^*, D^*，署名者の公開鍵 pk

Step 1: $i=1,\ldots,n$ に対し，入力部分文書集合に m_i が含まれる場合，$(2, H(m_i||r_i))$ と補助点 c_i から直線 f_i と $e_i'=f_i(0)$ を求める．入力データに b_i が含まれる場合，$(1, H(b_i))$ と補助点 c_i から直線 f_i と $e_i'=f_i(0)$ を求める．

Step 2: 署名者の pk を用いて，

$$\mathrm{Verify}_{pk}(\sigma) \stackrel{?}{=} (e'_1||\cdots||e'_n||c_1||\cdots||c_n)$$

が成り立つか検証する．

2.6 宮崎らによる Aggregate 署名を用いた墨塗り署名（SUMI-6）

SUMI-6 は宮崎らによって提案された Aggregate 署名を用いた墨塗り署名方式である [2]. Aggregate 署名は，複数の署名者が各自の異なる文書に対して生成した署名を，個々の署名と同じ長さの Aggregate 署名と呼ばれる 1 個の署名を用いて，一括して検証することの出来る署名技術である．SUMI-6 では，Boneh らによる Aggregate 署名方式 [8] を用いている．

SUMI-6[2] は，各部分文書のハッシュ値を連結したものに署名を施すのではなく，各部分文書の個別署名と Aggregate 署名を生成することにより墨塗り署名を実現している．墨塗り処理を行う場合は，個別署名を用いて，Aggregate 署名を更新し，部分文書と個別署名を削除する．また，強制開示処理を行う場合は，個別署名のみを削除する．ここでは SUMI-6 の手順について説明する．

2.6.1 鍵生成

Step 1: $sk \in Z_p$ に対して，$pk \leftarrow g_2^{sk}$ とし，署名者の公開鍵と秘密鍵を生成する．

2.6.2 署名生成

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk

Step 1: 各部分文書 m_i に対し，個別署名 $\sigma_i \leftarrow H(m_i)^{sk}$ を求める．ここで H はハッシュ関数である．

Step 2: Aggregate 署名 $\sigma \leftarrow \prod_{i=0}^{n} \sigma_i$ を生成する．

出力 個別署名 $\sigma_1, \ldots, \sigma_n$，Aggregate 署名 σ，部分文書 $m_1, \ldots, m_n$

2.6.3 墨塗り

開示条件を以下のように部分文書のインデックス集合で定める．例えば，初期状態では，$M = \{1, \ldots, n\}$, $S = D = \phi$ である．n は総部分文書数とする．

M： 開示かつ追加墨塗り可能
S： 墨塗り
D： 開示かつ追加墨塗り不可（強制開示）

入力 個別署名 $\{\sigma_i | i \in M\}$，Aggregate 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，条件 M, S, D

$i \in M$ について，条件の変化に応じ次の処理を行う．

- 墨塗り
  Aggregate 署名を $\sigma \leftarrow \sigma/\sigma_i$ と更新する．m_i, σ_i を削除し，$M \leftarrow M \setminus \{i\}$，$S \leftarrow S \cup \{i\}$ と更新する．
- 強制開示
  σ_i を削除し，$M \leftarrow M \setminus \{i\}$，$D \leftarrow D \cup \{i\}$ と更新する．

出力 個別署名 $\{\sigma_i | i \in M\}$，Aggregate 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，条件 M, S, D

2.6.4 署名検証

入力 個別署名 $\{\sigma_i | i \in M\}$，Aggregate 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，条件 M, S, D，署名者の公開鍵 pk

Step 1: Aggregate 署名の検証を行う．

Step 2: 個別署名の検証を行う．

2.7 武仲らによる墨塗り署名（PIAT）

PIAT は武仲らによって提案された墨塗り署名方式である [6]．特徴として，墨塗り者と不正墨塗り者の特定が可能である，不正墨塗り箇所の特定が可能であることが挙げられる．墨塗り処理をするときに，墨塗り者の署名を施すことにより，墨塗り者の特定が可能であるため，SUMI-4 のように匿名性のある方式ではなく，墨塗り者が明示的であるのが大きな特徴である．

2.7.1 署名生成

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk

Step 1: 部分文書 $m_i (i = 1, \ldots, n)$ に乱数 r_i を生成し，$m_i^{(0)} \leftarrow r_i || m_i$ とする．ハッシュ値 $h_i^{(0)} = H(m_i^{(0)})$ を求める．ここで H は一方向性ハッシュ関数とする．

Step 2: $h^{(0)} = h_1^{(0)} || \cdots || h_n^{(0)}$ とし，署名者の秘密鍵 sk を用いて，署名

$$\sigma^{(0)} = \mathrm{Sign}_{sk}(h^{(0)})$$

を生成する．

出力 署名 $\sigma^{(0)}$，部分文書 $m_1^{(0)}, \ldots, m_n^{(0)}$，ハッシュ値集合 $h^{(0)}$

2.7.2 墨塗り

j 番目の墨塗り者は以下の処理を行い墨塗りをする．

入力 署名 $\sigma^{(0)}, \ldots, \sigma^{(j-1)}$，部分文書 $m_1^{(j-1)}, \ldots, m_n^{(j-1)}$，ハッシュ値集合 $h^{(0)}, \ldots, h^{(j-1)}$，$j$ 番目の墨塗り者の秘密鍵 sk_j

Step 1: 2 組のハッシュ値集合 $h^{(0)}, h^{(j-1)}$ を比較し，開示インデックス集合 M を求める．

Step 2: i 番目の部分文書を墨塗りする場合，開示インデックス集合を $M \leftarrow M \setminus \{i\}$ に更新する．

Step 3: 各部分文書 $m_1^{(j-1)}, \ldots, m_n^{(j-1)}$ から新たに部分文書

$$m_i^{(j)} = \begin{cases} m_i^{(j-1)}, & i \in M \\ r_i' || m_i', & i \notin M \end{cases}$$

を生成する．ここで，r_i' は墨塗り者が付加する乱数，m_i' は適当な墨塗り文字列である．ハッシュ値 $h_i^{(j)} = H(m_i^{(j)})$ を求める．

Step 4: $h^{(j)} = h_1^{(j)} || \cdots || h_n^{(j)}$ とし，j 番目の墨塗り者の秘密鍵 sk_j を用いて，署名

$$\sigma^{(j)} = \mathrm{Sign}_{sk_j}(h^{(j)})$$

を生成する．

出力 署名 $\sigma^{(0)}, \ldots, \sigma^{(j)}$，部分文書 $m_1^{(j)}, \ldots, m_n^{(j)}$，ハッシュ値集合 $h^{(0)}, \ldots, h^{(j)}$

2.7.3 署名検証

入力 署名 $\sigma^{(0)}, \ldots, \sigma^{(j)}$，部分文書 $m_1^{(j)}, \ldots, m_n^{(j)}$，ハッシュ値集合 $h^{(0)}, \ldots, h^{(j)}$，署名者の公開鍵 pk，墨塗り者の公開鍵 $pk_1, \ldots, pk_j$

Step 1:墨塗り文書の検証 各部分文書 $m_1^{(j)}, \ldots, m_n^{(j)}$ からハッシュ値 $h^{(j)\prime} = h_1^{(j)\prime} || \cdots || h_n^{(j)\prime}$ を求め，$h^{(j)} \stackrel{?}{=} h^{(j)\prime}$ か検証する．

Step 2:署名検証 署名者の公開鍵 pk ，墨塗り者の公開鍵 $pk_1, \ldots, pk_j$ ，ハッシュ値集合 $h^{(0)}, \ldots, h^{(j)}$ を用いて，署名 $\sigma^{(0)}, \ldots, \sigma^{(j)}$ を検証する．

Step 3:墨塗り箇所の特定 $h^{(0)}$ と $h^{(j)}$ の 2 組のハッシュ値集合を比較し，墨塗り部分文書を特定する．

Step 4:墨塗り者の検証 ハッシュ値集合 $h^{(0)}, \ldots, h^{(j)}$ を用いて，墨塗りされた i 番目の部分文書の墨塗り者を特定する．

2.8 武仲らによる Aggregate 署名を用いた墨塗り署名（PIAT-A）

PIAT-A は武仲らによって提案された，Aggregate 署名を用いた墨塗り署名方式である [7]. Aggregate 署名は，複数の署名者が各自の異なる文書に対して生成した署名を，個々の署名と同じ長さの Aggregate 署名と呼ばれる 1 個の署名を用いて，一括して検証することの出来る署名技術である．

PIAT[6] では，検証者は署名者と複数の墨塗り者の署名を検証の際に必要としていた．そこで Aggregate 署名を用いることにより必要な署名を 1 個に削減できる．しかし，PIAT の特徴のひとつである，不正墨塗り者の特定はできなくなる．これは Aggregate 署名の検証に失敗した場合，どの署名の検証に失敗したかが分からないためである．

第3章
要素技術

3.1 RSA Accumulator

RSA Accumulator は値の集合とそれらを累積した単一のアキュムレータ (accumulator) から成っており，累積された任意の値を証明することができる暗号要素技術である．Accumulator は Benaloh らによって初めて [3] で提案された技術であり，Camenisch らによって公開鍵証明書の失効に応用されたり [9]，Johnson らによって集合準同型性を満たす署名 [10] に拡張されたりしている．

最初の実現方法は，Benaloh と de Mare[3] によるもので，次に示すように RSA 暗号に基づいている．

また，RSA Accumulator の例を図 3.1 に示す．素数の集合 $L = \{e_1, \ldots, e_4\}$ における RSA Accumulator に e_3 が累積されていることを証明している．

Step 1: 署名者は安全な素数 p と q を選び $N = pq$ を公開する．素数の集合 $L = \{e_1, \ldots, e_n\}$ について，アキュムレータ A を

$$A = a^{e_1 e_2 \cdots e_n} \pmod{N},$$

と定める．ただしここで，a は N と互いに素な定数であり公開する．署名者は，適切な署名アルゴリズムを用いて A に署名を行う．例えば，RSA 署名を用いれば，$\sigma(A) = A^{1/e_1 1/e_2 \cdots 1/e_n}$ とする．それを $\sigma(A)$ とする．

Step 2: 署名者が，L の任意の要素 e_i が A に累積されていることを証明するには，L と a を用いて，

$$A_i = a^{e_1 \cdots e_{i-1} e_{i+1} \cdots e_n} \pmod{N}$$

で定められる証拠 (witness) A_i を計算する．

Step 3: 検証者は，

$$A_i^{e_i} \stackrel{?}{=} A \pmod{N}$$

により証拠を確かめる．強 RSA 仮定の下で，N の素因数を知らないプレーヤーが証拠 A_i を提示できるのは，$e_i \in L$ の時に限ることが証明されている．

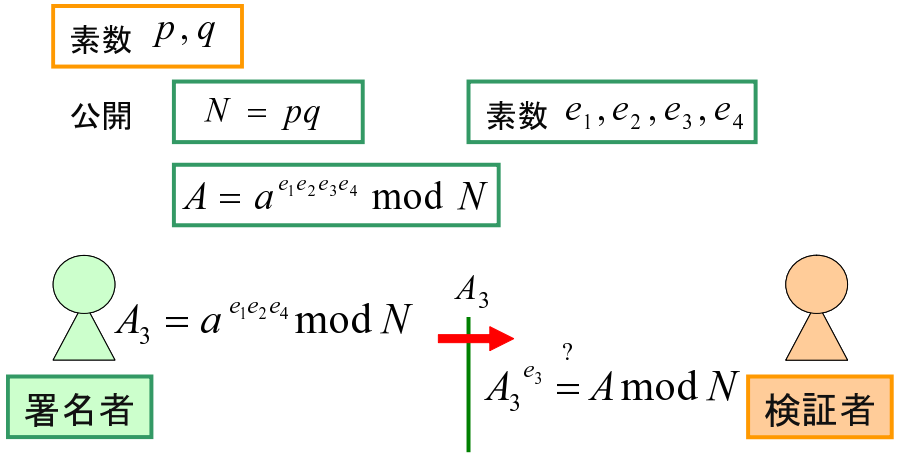

図 3.1: RSA Accumulator

3.2 Lagrange の補間公式

t 点から $t-1$ 次のある多項式 f を求めるには Lagrange の補間公式を用いることによって，容易に求めることができる．Lagrange の補間公式は，

$$f(x) = \sum_{j=1}^{t} \lambda_j(x) f(j),$$

$$\lambda_j(x) = \prod_{\ell \neq j} \frac{x-\ell}{j-\ell}$$

で表される．

第4章
RSA Accumulatorを用いた墨塗り署名

本章ではRSA Accumulator用いて署名長を削減した墨塗り署名方式を提案する．

宮崎らが提案した墨塗り者の匿名性の高い方式であるSUMI-4[5]は，各部分文書のハッシュ値の連結に署名を施すことにより墨塗り署名を実現している．この方式は，墨塗りを行った部分文書のハッシュ値を署名データとして保持する必要がある．

そこで，提案方式はRSA Accumulatorを使い，墨塗り部分文書のハッシュ値を束ねておくことにより，ハッシュ値を保持する必要がないため，署名長を削減することができる．これにより，墨塗り部分文書のハッシュ値に関するデータは，常に墨塗り部分文書数に依存せず1個となる．

4.1 提案方式

4.1.1 署名生成

署名者は，RSAの公開鍵Nと適切な署名アルゴリズムのpkとskを用意する．Nと互いに素な$a \in Z_N$を選び公開する．

入力 部分文書$m_1, \ldots, m_n$，署名者の秘密鍵sk，a

Step 1: 各部分文書m_iに対し，$e_i = H(m_i||r_i||i)$が素数になるように，乱数r_iを決める．ここで，Hはハッシュ関数とする．

Step 2: 署名者の秘密鍵skを用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(a^{e_1 \cdots e_n} \bmod N)$$

を生成する．

出力 署名σ，部分文書$m_1, \ldots, m_n$，乱数$r_1, \ldots, r_n$，a

4.1.2 墨塗り

入力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $r_1, \ldots, r_n$，a，墨塗りインデックス集合 S

Step 1: i 番目の部分文書を墨塗りする場合，$e_i = H(m_i||r_i||i)$ を求め，墨塗りインデックス集合 S に i を入れ，証拠

$$A_S = a^{\Pi_{j \in X} e_j} \bmod N$$

を求める．

Step 2: 部分文書 m_i と乱数 r_i を削除する．

出力 署名 σ，部分文書集合 $\{m_i | i \notin S\}$，乱数集合 $\{r_i | i \notin S\}$，証拠 A_S，墨塗りインデックス集合 S

4.1.3 署名検証

入力 署名 σ，部分文書集合 $\{m_i | i \notin S\}$，乱数集合 $\{r_i | i \notin S\}$，証拠 A_S，墨塗りインデックス集合 S

Step 1: 墨塗りインデックス集合 S に i が含まれていない場合は，i の部分文書に対する $e_i = H(m_i||r_i||i)$ を求める．

Step 2: 署名者の公開鍵 pk を用いて，以下を検証する．

$$\begin{aligned} \mathrm{Verify}_{pk}(\sigma) &\stackrel{?}{=} (a^{\Pi_{i \in X} e_i})^{\Pi_{i \notin X} e_i} \bmod N \\ &\stackrel{?}{=} a^{e_1 \cdots e_n} \bmod N \end{aligned}$$

4.1.4 計算例

提案方式の処理例を図 4.1 に示す．部分文書数 $n = 5$ ，墨塗りインデックス集合 $S = \{2, 3\}$ の場合の署名処理，墨塗り処理，及び，検証処理の手順を示す．

4.2 評価

4.2.1 署名長

提案方式における検証者が保持する署名長の比較を表 4.1 に示す．比較対象として，SUMI-4[5] をあげる．ここで，n は部分文書数，k は墨塗り部分文書数を表す．

表 4.1 から分かるように，RSA Accumulator を用いることにより，墨塗り部分文書のハッシュ値を束ねることが出来るため，従来方式と比較して，検証者が保持する署名長を削減することができた．

署名	部分文書 m_i	m_1 m_2 m_3 m_4 m_5
	乱数 r_i	r_1 b_2 b_3 r_4 b_5
	A_S	$A_S = a$
	インデックス集合	$S = \phi$
m_3 墨塗り	部分文書 m_i	m_1 m_2 $\not{m}_3$ m_4 m_5
	乱数 r_i	r_1 b_2 $\not{b}_3$ r_4 b_5
	A_S	$A_S = a^{e_3}$
	インデックス集合	$S = \{3\}$
m_2 墨塗り	部分文書 m_i	m_1 $\not{m}_2$ $\not{m}_3$ m_4 m_5
	乱数 r_i	r_1 $\not{b}_2$ $\not{b}_3$ r_4 b_5
	A_S	$A_S = a^{e_2 e_3}$
	インデックス集合	$S = \{2, 3\}$
検証	m_1, m_4, m_5	$a^{e_1 e_2 e_3 e_4 e_5} = (a^{e_2 e_3})^{e_1 e_4 e_5}$

図 4.1: 提案方式の計算例

表 4.1: 検証者が保持する署名長の比較

	SUMI-4[5]	提案方式
署名長	n	$n-k+1$
ハッシュ値の数	k	1
文書	$n-k$	$n-k$
署名長	$\mathcal{O}(n)$	$\mathcal{O}(n-k)$

第5章
RSA Accumulatorを用いた開示条件付き墨塗り署名

通常，墨塗り署名を使用するとき，墨塗り者は複数人いる．i 番目の墨塗り者が，開示した部分を $i+1$ 番目以降の墨塗り者が追加墨塗りする可能性がある．しかし，開示した部分に対し追加墨塗りを行われたくない場合が出てくる．そこで SUMI-5[1] では以降の墨塗りを禁止する設定を可能にしている．ここで，開示条件を決めるのは墨塗り者であり，署名者は開示条件を決めない．

開示条件を設定可能な方式には，多項式補間を用いた SUMI-5[1] や Aggregate 署名を用いた SUMI-6[2] がある．総部分文書数 n と墨塗り部分文書数 k に対して，$\mathcal{O}(n)$ の署名長が必要になる SUMI-5[1] に対して，Aggregate 署名を用いた SUMI-6[2] では $\mathcal{O}(n-k)$ の長さになり通信効率がよい．しかし，Aggregate 署名は楕円曲線上の双線形写像を実現する必要があり，現時点では通常の署名の数十倍の計算コストがかかると言われている．

本章では RSA Accumulator 用いた開示条件付き墨塗り署名方式 [11][12] を 2 つ提案する．提案方式は，宮崎らが提案した SUMI-5[1] をベースとし，複数の素数を単一の証拠 (wittness) で効率よく証明可能な RSA Accumulator[3] を用いた方式である．提案方式は Aggregate 署名（双線形写像）を用いることなく，SUMI-6[2] と同等の $\mathcal{O}(n-k)$ のサイズの効率的な署名長を実現する．また，提案方式 2（UK2）は，提案方式 1（UK1）に比べ，保持する署名データを少なくした改良方式である．従来方式と提案方式における計算コストと署名データにおける比較を表 5.1 に示す．

2 つの提案方式の特徴として，強制開示を設定した順序が検証可能であること，そして，墨塗り部分文書に関する署名データを束ねておくことにより，墨塗り部分文書に関するデータを保持する必要がないため，署名長を削減することができることが挙げられる．これにより，墨塗り部分文書に関するデータは，常に墨塗り部分文書数に依存せず 1 個となる．

表 5.1: 従来方式と提案方式における比較

	SUMI-5[1]	SUMI-6[2]	提案方式
計算コスト	○	× (ペアリング)	○ (RSA)
署名データ	× $\mathcal{O}(n)$	○ $\mathcal{O}(n-k)$	○ $\mathcal{O}(n-k)$

5.1 提案方式 1（UK1）

5.1.1 基本性質

一次多項式 $f(x)$ について,

$$\begin{cases} f(0) = e, \\ f(1) = H(b), \\ f(2) = H(m), \\ f(3) = c, \end{cases}$$

の関係がある時 (図 5.1 参照)，次の性質が成立する．ただし，ここで全ての演算は有限体の上で行われるとする．H はハッシュ関数である．

性質 1 $H(m)$ と c から,

$$\begin{aligned} e &= \frac{3}{3-2}f(2) + \frac{2}{2-3}f(3) \\ &= 3H(m) - 2c. \end{aligned}$$

性質 2 $a, a^c \bmod N$ と $H(m)$ から,

$$a^e = a^{3H(m)}/(a^c)^2 = a^{f(0)} \bmod N.$$

性質 3 $H(b)$ と $H(m)$ から,

$$\begin{aligned} e &= \frac{2}{2-1}f(1) + \frac{1}{1-2}f(2) \\ &= 2H(b) - H(m). \end{aligned}$$

性質 4 $a \bmod N$ と $H(m)$ と $H(b)$ から,

$$a^e = a^{2H(b)}/a^{H(m)} = a^{f(0)} \bmod N.$$

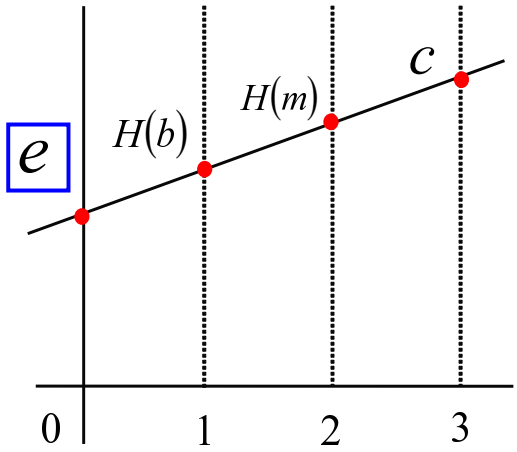

図 5.1: 提案方式 1 (UK1)

5.1.2 署名生成

署名者は，RSA の公開鍵 N と適切な署名アルゴリズムの pk と sk を用意する．N と互いに素な $a \in Z_N$ を選び公開する．

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk，a

Step 1: 各部分文書 m_i に対し，乱数 b_i を生成し，次の条件を満たす一次多項式 $f_i(x)$ を決める．

$$\begin{cases} f_i(0) = e_i, \\ f_i(1) = H(b_i), \\ f_i(2) = H(m_i||i), \\ f_i(3) = c_i, \end{cases}$$

ただし，e_i と c_i が素数となるまで乱数 b_i を繰り返し生成し求める．ここで，H はハッシュ関数である．

Step 2: $\alpha_i = a^{c_i} \bmod N$ を求める．

Step 3: 署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(a^{e_1 \cdots e_n} \bmod N)$$

を生成する．

出力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $b_1, \ldots, b_n$，コミットメント $\alpha_0 = a$，$\alpha_1, \ldots, \alpha_n$，条件 $M = \{1, \ldots, n\}$，$S = D = \phi$

ここで M, S, D は各部分文書の開示条件を表すインデックス集合であり，M は開示 (強制開示も墨塗りもされていない状態)，S は墨塗り，D は強制開示を表す．ただし，D は追加された順番も保持しているとする．

5.1.3 墨塗り

入力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in M \cup D \cup \{0\}\}$，条件 M, S, D

- i 番目を墨塗り

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$S \leftarrow S \cup \{i\}$ にする．

  Step 2: m_i と b_i から，性質 4 を用いて，全ての $j \in M \cup D \cup \{0\}$ について，$\alpha_j \leftarrow \alpha_j^{e_i}$ を求めて更新する．

  Step 3: m_i, b_i, α_i を削除する．

- i 番目を強制開示

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$D \leftarrow D \cup \{i\}$ にする．

  Step 2: m_i と b_i をから，性質 4 を用いて，全ての $j \in M$ について，$\alpha_j \leftarrow \alpha_j^{e_i}$ を求めて更新する．

  Step 3: b_i を削除する．

出力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント α_0，$\{\alpha_i | i \in M \cup D\}$，条件 M，S，D

墨塗りを繰り返した後，最終状態を M^*, S^*, D^*, α_i^* とおく．

5.1.4 署名検証

入力 署名 σ，部分文書 $\{m_i | i \in M^* \cup D^*\}$，乱数 $\{b_i | i \in M^*\}$，コミットメント $\{\alpha_i^* | i \in M^* \cup D^* \cup \{0\}\}$，条件 M^*，S^*，D^*，署名者の公開鍵 pk

Step 1: 検証用コミットメント α^* の初期値を α_0^* に設定する．$i \in D^*$ について，m_i と α_i^* と α^* を性質 2 に適用して，

$$\alpha^* \leftarrow (\alpha^*)^{e_i} = (\alpha^*)^{3H(m_i||i)} / (\alpha_i^*)^2 \bmod N$$

を求め，α^* を更新する．ただし，i は D に追加された順序で処理されていくとする．したがって，最終的には，

$$\alpha^* = \alpha^{\Pi_{i \in S^* \cup D^*} e_i}$$

が得られる．

Step 2: $i \in M^*$ について，m_i と b_i を用い，性質 4 より，

$$\alpha^* \leftarrow (\alpha^*)^{2H(b_i)}/(\alpha^*)^{H(m_i||i)}$$

を求めて，最終的に

$$\alpha^* = \alpha^{e_1 e_2 \cdots e_n} \bmod N$$

を得る．

Step 3: 署名者の公開鍵 pk を用い，

$$\mathrm{Verify}_{pk}(\sigma, a^*) \stackrel{?}{=} \mathrm{valid}$$

を検証する．

5.1.5 計算例

提案方式 1（UK1）の処理例を図 5.2 に示す．部分文書数 $n = 6$ ，強制開示 $D = (m_6, m_4)$，墨塗り $S = \{m_2, m_3\}$ の場合の署名処理，墨塗り処理，及び，検証処理の手順を示す．

署名	部分文書 m_i		m_1	m_2	m_3	m_4	m_5	m_6
	乱数 b_i		b_1	b_2	b_3	b_4	b_5	b_6
	コミットメント α_i	a	a^{c_1}	a^{c_2}	a^{c_3}	a^{c_4}	a^{c_5}	a^{c_6}
	インデックス集合	$M = \{1, 2, 3, 4, 5, 6\}$, $S = \phi$, $D = \phi$						
m_2 墨塗り	部分文書 m_i		m_1	$\not{m}_2$	m_3	m_4	m_5	m_6
	乱数 b_i		b_1	$\not{b}_2$	b_3	b_4	b_5	b_6
	コミットメント α_i	a^{e_2}	$a^{c_1 e_2}$	$\times$	$a^{c_3 e_2}$	$a^{c_4 e_2}$	$a^{c_5 e_2}$	$a^{c_6 e_2}$
	インデックス集合	$M = \{1, 3, 4, 5, 6\}$, $S = \{2\}$, $D = \phi$						
m_6 強制開示	部分文書 m_i		m_1	$\not{m}_2$	m_3	m_4	m_5	m_6
	乱数 b_i		b_1	$\not{b}_2$	b_3	b_4	b_5	$\not{b}_6$
	コミットメント α_i	a^{e_2}	$a^{c_1 e_2 e_6}$	$\times$	$a^{c_3 e_2 e_6}$	$a^{c_4 e_2 e_6}$	$a^{c_5 e_2 e_6}$	$a^{c_6 e_2}$
	インデックス集合	$M = \{1, 3, 4, 5\}$, $S = \{2\}$, $D = (6)$						
m_4 強制開示	部分文書 m_i		m_1	$\not{m}_2$	m_3	m_4	m_5	m_6
	乱数 b_i		b_1	$\not{b}_2$	b_3	$\not{b}_4$	b_5	$\not{b}_6$
	コミットメント α_i	a^{e_2}	$a^{c_1 e_2 e_6 e_4}$	$\times$	$a^{c_3 e_2 e_6 e_4}$	$a^{c_4 e_2 e_6}$	$a^{c_5 e_2 e_6 e_4}$	$a^{c_6 e_2}$
	インデックス集合	$M = \{1, 3, 5\}$, $S = \{2\}$, $D = (6, 4)$						
m_3 墨塗り	部分文書 m_i		m_1	$\not{m}_2$	$\not{m}_3$	m_4	m_5	m_6
	乱数 b_i		b_1	$\not{b}_2$	$\not{b}_3$	$\not{b}_4$	b_5	$\not{b}_6$
	コミットメント α_i	$a^{e_2 e_3}$	$a^{c_1 e_2 e_6 e_4 e_3}$	$\times$	$\times$	$a^{c_4 e_2 e_6 e_3}$	$a^{c_5 e_2 e_6 e_4 e_3}$	$a^{c_6 e_2 e_3}$
	インデックス集合	$M = \{1, 5\}$, $S = \{2, 3\}$, $D = (6, 4)$						
検証	m_6	$a^{e_2 e_3 e_6} = (a^{e_2 e_3})^{3H(m_6 \| 6)} / (a^{c_6 e_2 e_3})^2$						
	m_4	$a^{e_2 e_3 e_4 e_6} = (a^{e_2 e_3 e_6})^{3H(m_4 \| 4)} / (a^{c_4 e_2 e_3 e_6})^2$						
	m_1	$a^{e_1 e_2 e_3 e_4 e_6} = (a^{e_2 e_3 e_4 e_6})^{2H(b_1)} / (a^{e_2 e_3 e_4 e_6})^{H(m_1 \| 1)}$						
	m_5	$a^{e_1 e_2 e_3 e_4 e_5 e_6} = (a^{e_1 e_2 e_3 e_4 e_6})^{2H(b_5 \models)} / (a^{e_1 e_2 e_3 e_4 e_6})^{H(m_5 \| 5)}$						

図 5.2: 提案方式 1(UK1) の計算例

5.2 提案方式 2（UK2）

5.2.1 基本性質

一次多項式 $f(x)$ について，

$$\begin{cases} f(0) = e, \\ f(1) = H(b), \\ f(2) = H(m), \end{cases}$$

の関係がある時 (図 5.3 参照)，次の性質が成立する．ただし，ここで全ての演算は有限体の上で行われるとする．H はハッシュ関数である．

性質 3 $H(b)$ と $H(m)$ から，

$$\begin{aligned} e &= \frac{2}{2-1}f(1) + \frac{1}{1-2}f(2) \\ &= 2H(b) - H(m). \end{aligned}$$

性質 4 a と $H(m)$ と $H(b)$ から，

$$a^e = a^{2H(b)}/a^{H(m)} = a^{f(0)} \bmod N.$$

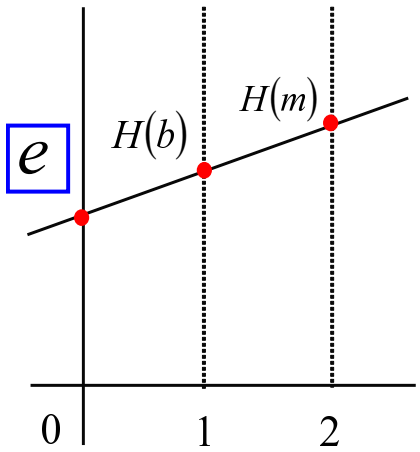

図 5.3: 提案方式 2(UK2)

5.2.2 署名生成

署名者は，RSA の公開鍵 N と適切な署名アルゴリズムの pk と sk を用意する．N と互いに素な $a \in Z_N$ を選び公開する．

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk，a

Step 1: 各部分文書 m_i に対し，乱数 b_i を生成し，次の条件を満たす一次多項式 $f_i(x)$ を決める．

$$\begin{cases} f_i(0) = e_i, \\ f_i(1) = H(b_i), \\ f_i(2) = H(m_i||i), \end{cases}$$

ただし，e_i は素数となるまで乱数 b_i を繰り返し生成し求める[1]．ここで，H はハッシュ関数である．

Step 2: $a^{e_1 \cdots e_n} \bmod N$ を求め，署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(a^{e_1 \cdots e_n} \bmod N)$$

を生成する．

出力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $b_1, \ldots, b_n$，コミットメント $\alpha_0 = a$，条件 $M = \{1, \ldots, n\}$，$S = D = \phi$

ここで M, S, D は各部分文書の開示条件を表すインデックス集合であり，M は開示 (強制開示も墨塗りもされていない状態)，S は墨塗り，D は強制開示を表す．ただし，D は追加された順番も保持しているとする．

5.2.3 墨塗り

入力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，条件 M, S, D

- i 番目を墨塗り

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$S \leftarrow S \cup \{i\}$ と更新する．

  Step 2: m_i と b_i から，性質 4 を用いて，

  $$\alpha_0 \leftarrow \alpha_0^{e_i}$$

  と更新する．

  Step 3: m_i, b_i を削除する．

- i 番目を強制開示

[1] 乱数 b_i を変化させたときに $f_i(0)$ の値がランダムにふるまうと仮定すると，e_i が素数となる確率は素数の分布定理により $p = 1/\ln 2^t$ となる．ここで t はハッシュ関数の出力ビット長であり，$t = 160$ ならば $p \approx 1/110.9$ となる．

Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$D \leftarrow D \cup \{i\}$ にする．

Step 2: 性質 4 を用いて，$\{m_j | j \in M\}$ と $\{b_j | j \in M\}$，α_0 から，

$$\alpha_i \leftarrow a^{\Pi_{j \in S \cup M} e_j H(b_i)}$$

を求める．

Step 3: b_i を削除する．

出力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，条件 M，S，D

墨塗りを繰り返した後，最終状態を M^*, S^*, D^*, ${\alpha_0}^*$ とおく．

5.2.4 署名検証

入力 署名 σ，部分文書 $\{m_i | i \in M^* \cup D^*\}$，乱数 $\{b_i | i \in M^*\}$，コミットメント ${\alpha_0}^*$, $\{\alpha_i | i \in D^*\}$，条件 M^*，S^*，D^*，署名者の公開鍵 pk

Step 1: 検証用コミットメント α^* の初期値を ${\alpha_0}^*$ に設定する．$i \in M^*$ について，m_i と b_i と α^* を用い，性質 4 より，

$$\alpha^* \leftarrow (\alpha^*)^{e_i} = (\alpha^*)^{2H(b_i)} / (\alpha^*)^{H(m_i||i)}$$

を求め，α^* を更新する．従って最終的には，$\alpha^* = a^{\Pi_{j \in M^* \cup S^*} e_j}$ が得られる．

Step 2: $i \in D^*$ について，m_i と α_i と α^* を性質 4 に適用して，

$$\alpha^* \leftarrow (\alpha_i)^2 / (\alpha^*)^{H(m_i||i)}$$

を求め，α^* を更新する．ただし，i は，D に追加された順序とは逆順序で処理されていくとする．したがって最終的に

$$\alpha^* = a^{e_1 e_2 \cdots e_n} \bmod N$$

を得る．

Step 3: 署名者の公開鍵 pk を用い，

$$\mathrm{Verify}_{pk}(\sigma, \alpha^*) \stackrel{?}{=} \mathrm{valid}$$

を検証する．

署名	部分文書 m_i	m_1 m_2 m_3 m_4 m_5 m_6
	乱数 b_i	b_1 b_2 b_3 b_4 b_5 b_6
	コミットメント α_i	a
	インデックス集合	$M = \{1, 2, 3, 4, 5, 6\}$, $S = \phi$, $D = \phi$
m_2 墨塗り	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 b_4 b_5 b_6
	コミットメント α_i	a^{e_2}
	インデックス集合	$M = \{1, 3, 4, 5, 6\}$, $S = \{2\}$, $D = \phi$
m_6 強制開示	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 b_4 b_5 $\not{b}_6$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 4, 5\}$, $S = \{2\}$, $D = (6)$
m_4 強制開示	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 $\not{b}_4$ b_5 $\not{b}_6$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 5\}$, $S = \{2\}$, $D = (6, 4)$
m_3 墨塗り	部分文書 m_i	m_1 $\not{m}_2$ $\not{m}_3$ m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ $\not{b}_3$ $\not{b}_4$ b_5 $\not{b}_6$
	コミットメント α_i	$a^{e_2e_3}$ $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 5\}$, $S = \{2, 3\}$, $D = (6, 4)$
検証	m_1	$a^{e_1e_2e_3} = (a^{e_2e_3})^{2H(b_1)}/(a^{e_2e_3})^{H(m_1\|\|1)}$
	m_5	$a^{e_1e_2e_3e_5} = (a^{e_1e_2e_3})^{2H(b_5)}/(a^{e_1e_2e_3})^{H(m_5\|\|5)}$
	m_4	$a^{e_1e_2e_3e_4e_5} = (a^{e_1e_2e_3e_5H(b_4)})^2/(a^{e_1e_2e_3e_5})^{H(m_4\|\|4)}$
	m_6	$a^{e_1e_2e_3e_4e_5e_6} = (a^{e_1e_2e_3e_4e_5H(b_6)})^2/(a^{e_1e_2e_3e_4e_5})^{H(m_6\|\|6)}$

図 5.4: 提案方式 2(UK2) の計算例

5.2.5 計算例

提案方式 2（UK2）の処理例を図 5.4 に示す．部分文書数 $n = 6$ ， 強制開示 $D = (m_6, m_4)$, 墨塗り $S = \{m_2, m_3\}$ の場合の署名処理，墨塗り処理，及び，検証処理の手順を示す．

5.3 評価

5.3.1 署名長

提案方式 1（UK1）と提案方式 2（UK2）における検証者が保持する署名データの比較を表 5.2 に示す．比較対象として，SUMI-5[1] と SUMI-6[2] をあげる．ここで n は部分文書数，k は墨塗り部分文書数，ℓ は強制開示部分文書数を表す．

提案方式 1（UK1）と提案方式 2（UK2）は，乱数の数では SUMI-6 と比較すると多いが，SUMI-5 とより少ない．また，署名の個数としては，SUMI-6 は 1 つの Aggregate 署名と $n-k-\ell$ 個の個別署名が必要なのに対して，提案方式 1（UK1）と提案方式 2（UK2）は 1 個の署名でよい．よって，$\mathcal{O}(n-k)$ の通信量となる．

表 5.2: 検証者が保持する署名データの比較

	SUMI-5[1]	SUMI-6[2]	提案方式 1（UK1）	提案方式 2（UK2）
文書	$n-k$	$n-k$	$n-k$	$n-k$
コミットメント	n	0	$n-k+1$	$\ell+1$
乱数	$n-k, n-\ell$	0	$n-k-\ell$	$n-k-\ell$
署名	1	$1+n-k-\ell$	1	1
署名長	$\mathcal{O}(n)$	$\mathcal{O}(n-k)$	$\mathcal{O}(n-k)$	$\mathcal{O}(n-k)$

5.4 問題点

伊豆らによって，提案方式 1 (UK1) と提案方式 2(UK2) は，攻撃者が墨塗り者になりすまし，部分文書を任意の内容に偽造した上で，強制開示の状態に変更することが可能であるという問題点が指摘された [13].

提案方式 1(UK1) では 4 点 $(0, e_i), (1, H(b_i)), (2, H(m_i||i)), (3, c_i)$ のうち，署名対象であるのは e_i のみである (図 5.5)．そのため， b_i, m_i, c_i の 3 点を偽造できる可能性がでてくる．偽造を行った場合，偽造部分文書 m'_i における 4 点は $(0, e_i), (1, H(b_i)'), (2, H(m'_i||i)), (3, c'_i)$ となる．ここで， $H(b_i)'$ となる点を求めることは可能であるが，ハッシュ関数の一方向性により b'_i 自体を求めることは難しい．よって，状態が開示 (強制開示も墨塗りもされていない状態) の場合は，b_i と m_i が必要となるため，部分文書を偽造し，状態を開示にすることは出来ない．しかし，提案方式 1(UK1) において，部分文書の状態が強制開示の場合は，乱数 b_i を保持しなくていい．よって，攻撃者が墨塗り者になりすまし，部分文書を任意の内容に偽造した上で，強制開示の状態に変更することが可能となる．

また，提案方式 2(UK2) でも 3 点 $(0, e_i), (1, H(b_i)), (2, H(m_i||i))$ のうち，署名対象であるのは e_i のみである (図 5.6)．よって，提案方式 1(UK1) と同様の攻撃を受ける．

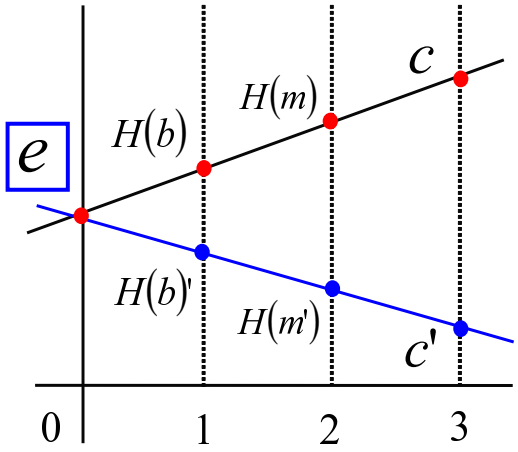

図 5.5: 提案方式 1 (UK1) における攻撃

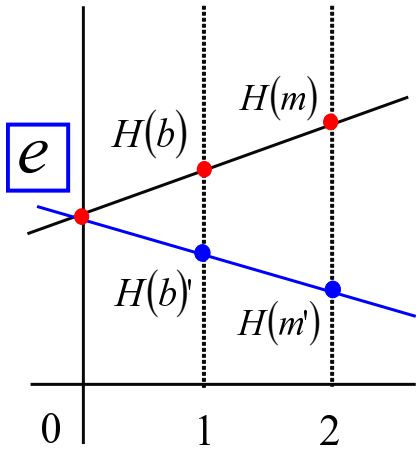

図 5.6: 提案方式 2(UK2) における攻撃

ここでは，具体的な攻撃手順について述べる．

5.4.1 提案方式 1（UK1）に対する攻撃

入力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in M \cup D \cup \{0\}\}$，条件 M, S, D

Step 1: 状態が開示である部分文書から偽造対象部分文書 m_j を選択する．部分文書 m_j と乱数 b_j から e_j を求めておく．

Step 2: 偽造部分文書 m'_j を生成し，

$$\begin{cases} f_j(0) = e_j, \\ f_j(1) = H(b_j)', \\ f_j(2) = H(m'_j||j), \\ f_j(3) = c'_j, \end{cases}$$

満たすように整数 c'_j を求める．

Step 3: 偽造コミットメント α'_j の初期値を α_0 とする．全ての $i \in D$ に対して，

$$\alpha'_j \leftarrow (\alpha'_j)^{3H(m_i||i)}/(\alpha_i)^2 \bmod N$$

を求める．ここで i は D に追加された順番で処理される．最後に

$$\alpha'_j \leftarrow (\alpha'_j)^{c'_j}$$

と更新する．全ての $i \in M$ に対して，

$$\alpha_i \leftarrow (\alpha_i)^{e_j}$$

と更新する．

Step 4: 条件を $M \leftarrow M \setminus \{j\}, D \leftarrow D \cup \{j\}$ と更新する．

Step 5: 部分文書を $m_j \leftarrow m'_j$，コミットメントを $\alpha_j \leftarrow \alpha'_j$ と更新し，乱数 b_j を削除する．

5.4.2 提案方式 2（UK2）に対する攻撃

入力 署名 σ，部分文書 $\{m_i|i \in M \cup D\}$，乱数 $\{b_i|i \in M\}$，コミットメント $\{\alpha_i|i \in M \cup D \cup \{0\}\}$，条件 M, S, D

Step 1: 状態が開示である部分文書から偽造対象部分文書 m_j を選択する．部分文書 m_j と乱数 b_j から e_j を求めておく．

Step 2: 偽造部分文書 m'_j を生成し，

$$\begin{cases} f_j(0) = e_j, \\ f_j(1) = H(b_j)', \\ f_j(2) = H(m'_j||j), \end{cases}$$

満たすように整数 $B'_j = H(b_j)'$ を求める．つまり，$B'_j = (e_j + H(m'_j||j))/2$ が整数となるように求める．(このような m'_j は 1/2 の確率で定まる．)

Step 3: 条件を $M \leftarrow M \setminus \{j\}, D \leftarrow D \cup \{j\}$ と更新する．

Step 4: 偽造コミットメント α'_j の初期値を α_0 とする．全ての $i \in M$ に対して，m_i と b_i から e_i を求め

$$\alpha'_j \leftarrow (\alpha'_j)^{e'_j} \bmod N$$

と更新する．最後に

$$\alpha'_j \leftarrow (\alpha'_j)^{B'_j}$$

と更新する．

Step 5: 部分文書を $m_j \leftarrow m'_j$，コミットメントを $\alpha_j \leftarrow \alpha'_j$ と更新し，乱数 b_j を削除する．

第6章
RSA Accumulatorを用いた開示条件付き墨塗り署名の改良

[11]，[12] にて，RSA Accumulator を用いた開示条件を設定可能な墨塗り署名方式を 2 つ提案した．この方式は Aggregate 署名（双線形写像）を用いることなく，SUMI-6[2] と同等の $\mathcal{O}(n-k)$ のサイズの効率的な署名長を実現している．

しかし，UK1[11]，UK2[12] は墨塗り者になりすました攻撃者が開示状態の部分文書の内容を任意の内容に変更 (偽造) した上で，その属性を強制開示に変更する攻撃が可能であることが指摘された [13]．

本章では，この攻撃を防止して安全性を高めた 2 つの改良方式である，Dual Accumulator (DA) 方式と δ 方式を提案する [14]．DA 方式は 2 本の RSA Accumulator を用いて，[13] の偽造を検出可能にする方式である．一方，δ 方式は，攻撃の対象になっていた線形式上の隣り合う 2 点間を，十分な長さ δ にする方式である．

6.1 Dual Accumulator方式（UK-DA）

6.1.1 概要

UK2[12] における $f_i(0) = e_i$ についてのアキュムレータに加えて，$f_i(3) = c_i$ のアキュムレータを追加することで，強制開示攻撃を防止する方式である．

しかし，強制開示の状態のときに，e_i のアキュムレータと c_i のアキュムレータに各々異なる偽造した $H(b_i)^*$，$H(b_i)_*$ を累積しても検証に成功してしまう (図 6.1 参照)．

そこで，2 つのアキュムレータ A と B に共通の指数 η が累積されていることを，Cramer[15] の選言のゼロ知識証明を用いて（墨塗り者が）証明すればよい．η の知識の証明を

$$PK(\eta, a, b) = (t_1, t_2, s)$$

と書く．p を素数，a,b を Z_p の位数が素数 q となる生成元とすると，t_1, t_2, s は，次の値より

成る．

$$\begin{cases} t_1 = a^r, t_2 = b^r \bmod p, \\ d = H(t_1||t_2||a||b), \\ s = r + d\eta \bmod q, \end{cases}$$

ただし，H はハッシュ関数，r は Z_q の一様な乱数とする．すなわち，

$$PK\{H(\eta)|A = a^\eta \wedge B = b^\eta\}$$

を証明している．この検証には，

$$a^s \stackrel{?}{=} t_1 \cdot A^d, b^s \stackrel{?}{=} t_2 \cdot B^d$$

を調べる．

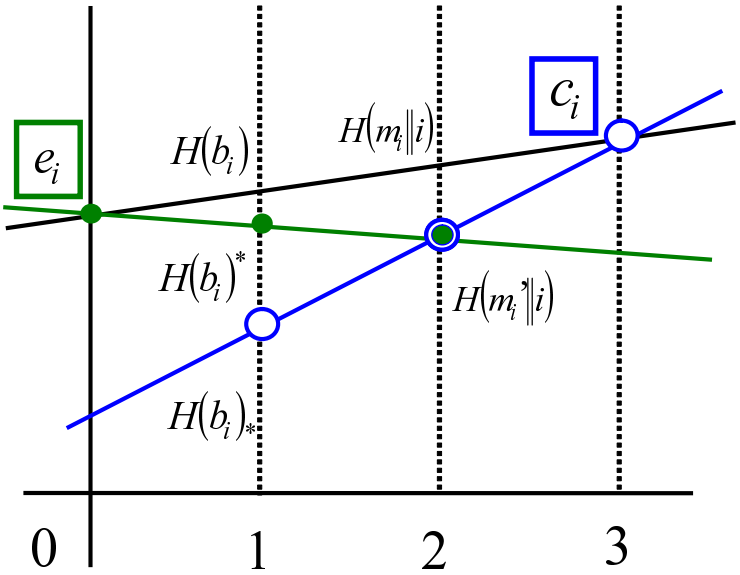

図 6.1: DA 方式の問題点

6.1.2 基本性質

一次多項式 $f(x)$ について，

$$\begin{cases} f(0) = e, \\ f(1) = H(b), \\ f(2) = H(m), \\ f(3) = c, \end{cases}$$

の関係がある時 (図 6.2 参照)，次の性質が成立する．ただし，ここで全ての演算は有限体の上で行われるとする．H はハッシュ関数である．

性質 3 $H(b)$ と $H(m)$ から，

$$\begin{aligned} e &= \frac{2}{2-1}f(1) + \frac{1}{1-2}f(2) \\ &= 2H(b) - H(m). \end{aligned}$$

性質 4 a と $H(m)$ と $H(b)$ から，

$$a^e = a^{2H(b)}/a^{H(m)} = a^{f(0)} \bmod N.$$

性質 5 $H(b)$ と $H(m)$ から，

$$\begin{aligned} c &= \frac{2-3}{2-1}f(1) + \frac{1-3}{1-2}f(2) \\ &= 2H(m) - H(b). \end{aligned}$$

性質 6 a と $H(m)$ と $H(b)$ から，

$$a^c = a^{2H(m)}/a^{H(b)} = a^{f(4)} \bmod N.$$

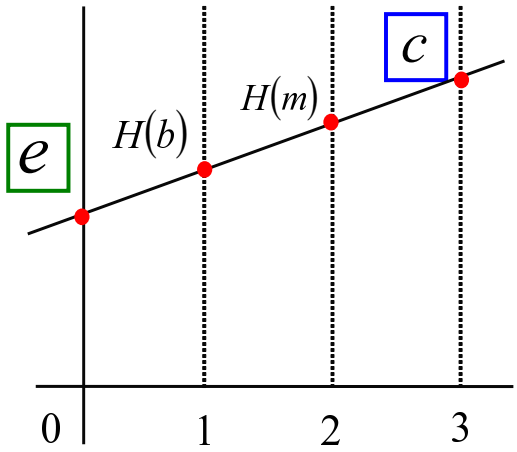

図 6.2: UK-DA

6.1.3 署名生成

署名者は，RSA の公開鍵 N と適切な署名アルゴリズムの pk と sk を用意する．N と互いに素な $a \in Z_N$ と $g \in Z_N$ を選び公開する．

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk，a，g

Step 1: 各部分文書 m_i に対し，乱数 b_i を生成し，次の条件を満たす一次多項式 $f_i(x)$ を決める．

$$\begin{cases} f_i(0) = e_i, \\ f_i(1) = H(b_i), \\ f_i(2) = H(m_i||i), \\ f_i(3) = c_i, \end{cases}$$

ただし，e_i と c_i は素数となるまで乱数 b_i を繰り返し生成し求める．ここで，H はハッシュ関数である．

Step 2: $a^{e_1 \cdots e_n} \bmod N$，$g^{c_1 \cdots c_n} \bmod N$ を求め，署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(a^{e_1 \cdots e_n}||g^{c_1 \cdots c_n})$$

を生成する．

出力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $b_1, \ldots, b_n$，コミットメント $\alpha_0 = a$，$\gamma_0 = g$，条件 $M = \{1, \ldots, n\}$，$S = D = \phi$

ここで M, S, D は各部分文書の開示条件を表すインデックス集合であり，M は開示 (強制開示も墨塗りもされていない状態)，S は墨塗り，D は強制開示を表す．ただし，D は追加された順番も保持しているとする．

6.1.4 墨塗り

入力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，$\{\gamma_i | i \in D \cup \{0\}\}$，$\{t_{1i} | i \in D\}$，$\{t_{2i} | i \in D\}$，$\{s_i | i \in D\}$，条件 M, S, D

- i 番目を墨塗り

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$S \leftarrow S \cup \{i\}$ と更新する．

  Step 2: m_i と b_i から，性質 4 を用いて，

  $$\alpha_0 \leftarrow \alpha_0^{e_i}$$

  と更新する．

  Step 3: m_i と b_i から，性質 6 を用いて，

  $$\gamma_0 \leftarrow \gamma_0^{c_i}$$

  と更新する．

Step 4: m_i, b_i を削除する．

- i 番目を強制開示

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$D \leftarrow D \cup \{i\}$ にする．

  Step 2: 性質 4 を用いて，$\{m_j | j \in M\}$ と $\{b_j | j \in M\}$，α_0 から，

  $$\alpha_i \leftarrow a^{\Pi_{j \in M \cup S} e_j H(b_i)}$$

  を求める．

  Step 3: 性質 6 を用いて，$\{m_j | j \in M\}$ と $\{b_j | j \in M\}$，γ_0 から，

  $$\gamma_i \leftarrow g^{\Pi_{j \in M \cup S} c_j H(b_i)}$$

  を求める．

  Step 4:

  $$\mathrm{PK}(H(b_i),\ a^{\Pi_{j \in M \cup S} e_j},\ g^{\Pi_{j \in M \cup S} c_j})$$

  を求める．

  Step 5: b_i を削除する．

出力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，$\{\gamma_i | i \in D \cup \{0\}\}$，$\{t_{1i} | i \in D\}$，$\{t_{2i} | i \in D\}$，$\{s_i | i \in D\}$，条件 M, S, D

墨塗りを繰り返した後，最終状態を M^*, S^*, D^*, ${\alpha_0}^*$, ${\gamma_0}^*$ とおく．

6.1.5 署名検証

入力 署名 σ，部分文書 $\{m_i | i \in M^* \cup D^*\}$，乱数 $\{b_i | i \in M^*\}$，コミットメント ${\alpha_0}^*$, $\{\alpha_i | i \in D^*\}$，${\gamma_0}^*$, $\{\gamma_i | i \in D^*\}$，$\{t_{1i} | i \in D *\}$，$\{t_{2i} | i \in D^*\}$，$\{s_i | i \in D^*\}$，条件 M^*，S^*，D^*，署名者の公開鍵 pk

Step 1: 検証用コミットメント α^* の初期値を ${\alpha_0}^*$，検証用コミットメント γ^* の初期値を ${\gamma_0}^*$ に設定する．

Step 2: $i \in M^*$ について，m_i と b_i と α^* を用い，性質 4 より，

$$\alpha^* \leftarrow (\alpha^*)^{e_i} = (\alpha^*)^{2H(b_i)} / (\alpha^*)^{H(m_i || i)}$$

を求め，α^* を更新する．従って最終的には，$\alpha^* = a^{\Pi_{j \in M^* \cup S^*} e_j}$ が得られる．

Step 3: $i \in M^*$ について，m_i と b_i と γ^* を用い，性質 6 より，

$$\gamma^* \leftarrow (\gamma^*)^{c_i} = (\gamma^*)^{2H(m_i||i)}/(\gamma^*)^{H(b_i)}$$

を求め，γ^* を更新する．従って最終的には，$\gamma^* = g^{\Pi_{j \in M^* \cup S^*} c_j}$ が得られる．

Step 4: $i \in D^*$ において，次式の PK の検証を行う．

$$\mathrm{PK}(H(b_i),\ a^{\Pi_{j \in M \cup S} e_j},\ g^{\Pi_{j \in M \cup S} c_j}) \stackrel{?}{=} (t_{1i}, t_{2i}, s_i)$$

Step 5: $i \in D^*$ について，m_i と α_i と α^* を性質 4 に適用して，

$$\alpha^* \leftarrow (\alpha_i)^2/(\alpha^*)^{H(m_i||i)}$$

を求め，α^* を更新する．ただし，i は，D に追加された順序とは逆順序で処理されていくとする．したがって最終的に

$$\alpha^* = a^{e_1 e_2 \cdots e_n} \bmod N$$

を得る．

Step 6: $i \in D^*$ について，m_i と γ_i と γ^* を性質 6 に適用して，

$$\gamma^* \leftarrow (\gamma^*)^{2H(m_i||i)}/(\gamma_i)$$

を求め，γ^* を更新する．ただし，i は，D に追加された順序とは逆順序で処理されていくとする．したがって最終的に

$$\gamma^* = g^{c_1 c_2 \cdots c_n} \bmod N$$

を得る．

Step 7: 署名者の公開鍵 pk を用い，

$$\mathrm{Verify}_{pk}(\sigma, \alpha^*||\gamma^*) \stackrel{?}{=} \mathrm{valid}$$

を検証する．

6.1.6 計算例

Dual Accumulator 方式（UK-DA）の処理例を図 6.3 に示す．部分文書数 $n = 6$ ，強制開示 $D = (m_6, m_4)$，墨塗り $S = \{m_2, m_3\}$ の場合の署名処理，墨塗り処理，及び，検証処理の手順を示す．

署名	部分文書 m_i	m_1 m_2 m_3 m_4 m_5 m_6
	乱数 b_i	b_1 b_2 b_3 b_4 b_5 b_6
	コミットメント α_i	a
	コミットメント γ_i	g
	インデックス集合	$M = \{1, 2, 3, 4, 5, 6\}$, $S = \phi$, $D = \phi$
m_2 墨塗り	部分文書 m_i	m_1 $\not{m_2}$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b_2}$ b_3 b_4 b_5 b_6
	コミットメント α_i	a^{e_2}
	コミットメント γ_i	g^{c_2}
	インデックス集合	$M = \{1, 3, 4, 5, 6\}$, $S = \{2\}$, $D = \phi$
m_6 強制開示	部分文書 m_i	m_1 $\not{m_2}$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b_2}$ b_3 b_4 b_5 $\not{b_6}$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_4e_5H(b_6)}$
	コミットメント γ_i	g^{c_2} $a^{c_1c_2c_3c_4c_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 4, 5\}$, $S = \{2\}$, $D = (6)$
m_4 強制開示	部分文書 m_i	m_1 $\not{m_2}$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b_2}$ b_3 $\not{b_4}$ b_5 $\not{b_6}$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	コミットメント γ_i	g^{c_2} $g^{c_1c_2c_3c_5H(b_4)}$ $g^{c_1c_2c_3c_4c_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 5\}$, $S = \{2\}$, $D = (6, 4)$
m_3 墨塗り	部分文書 m_i	m_1 $\not{m_2}$ $\not{m_3}$ m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b_2}$ $\not{b_3}$ $\not{b_4}$ b_5 $\not{b_6}$
	コミットメント α_i	$a^{e_2e_3}$ $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	コミットメント γ_i	$g^{c_2c_3}$ $g^{c_1c_2c_3c_5H(b_4)}$ $g^{c_1c_2c_3c_4c_5H(b_6)}$
	インデックス集合	$M = \{1, 5\}$, $S = \{2, 3\}$, $D = (6, 4)$
検証	m_1	$a^{e_1e_2e_3} = (a^{e_2e_3})^{2H(b_1)}/(a^{e_2e_3})^{H(m_1\|\|1)}$ $g^{c_1c_2c_3} = (g^{c_2c_3})^{2H(m_1\|\|1)}/(g^{c_2c_3})^{H(b_1)}$
	m_5	$a^{e_1e_2e_3e_5} = (a^{e_1e_2e_3})^{2H(b_5)}/(a^{e_1e_2e_3})^{H(m_5\|\|5)}$ $g^{c_1c_2c_3c_5} = (g^{c_1c_2c_3})^{2H(m_5\|\|5)}/(g^{c_1c_2c_3})^{H(b_5)}$
	m_4	$a^{e_1e_2e_3e_4e_5} = (a^{e_1e_2e_3e_5H(b_4)})^2/(a^{e_1e_2e_3e_5})^{H(m_4\|\|4)}$ $g^{c_1c_2c_3c_4c_5} = (g^{c_1c_2c_3c_5})^{2H(m_4\|\|4)}/(g^{c_1c_2c_3c_5H(b_4)})$
	m_6	$a^{e_1e_2e_3e_4e_5e_6} = (a^{e_1e_2e_3e_4e_5H(b_6)})^2/(a^{e_1e_2e_3e_4e_5})^{H(m_6\|\|6)}$ $g^{c_1c_2c_3c_4c_5c_6} = (g^{c_1c_2c_3c_4c_5})^{2H(m_6\|\|6)}/(g^{c_1c_2c_3c_4c_5H(b_6)})$

図 6.3: Dual Accumulator 方式（UK-DA）の計算例

6.2 δ方式（UK-δ）

6.2.1 概要

UK2[12] における線形式上の 2 点 $(1, H(b_i))$, $(2, H(m_i||i))$ を十分な大きさの δ について，$(\delta, H(b_i))$, $(\delta+1, H(m_i||i))$ とする方式である．

[13] において伊豆らによって指摘された，UK2[12] における強制開示攻撃は，偽造文書 $H(m_i'||i)$ について偽の乱数要素 $H(b_i)'$ が $H(b_i)' = (e_k + H(m_i'||i))/2$ が整数となる時に成功する．このような m_i' は 1/2 の確率で定まる．従って，δ 方式は十分な大きな δ について，攻撃を防止する．

6.2.2 基本性質

一次多項式 $f(x)$ について，

$$\begin{cases} f(0) = e, \\ f(\delta) = H(b), \\ f(\delta+1) = H(m), \end{cases}$$

の関係がある時 (図 6.4 参照)，次の性質が成立する．ただし，ここで全ての演算は有限体の上で行われるとする．H はハッシュ関数である．

性質 7 $H(b)$ と $H(m)$ から，

$$\begin{aligned} e &= \frac{(\delta+1)}{(\delta+1)-\delta} f(\delta) + \frac{\delta}{\delta-(\delta+1)} f(\delta+1) \\ &= (\delta+1)f(\delta) - \delta f(\delta+1) \\ &= (\delta+1)H(b) - \delta H(m). \end{aligned}$$

性質 8 a と $H(m)$ と $H(b)$ から，

$$a^e = a^{(\delta+1)H(b)} / a^{\delta H(m)} = a^{f(0)} \bmod N.$$

6.2.3 署名生成

署名者は，RSA の公開鍵 N と適切な署名アルゴリズムの pk と sk を用意する．N と互いに素な $a \in Z_N$ を選び公開する．定数 δ を定める．δ は公開情報である．

入力 部分文書 $m_1, \ldots, m_n$，署名者の秘密鍵 sk，a，定数 δ

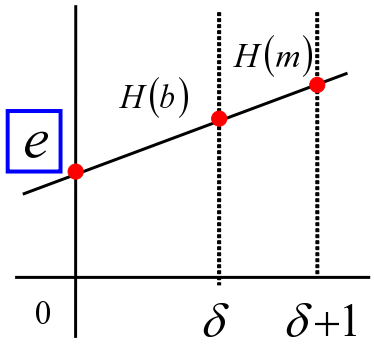

図 6.4: UK-δ

Step 1: 各部分文書 m_i に対し，乱数 b_i を生成し，次の条件を満たす一次多項式 $f_i(x)$ を決める．

$$\begin{cases} f_i(0) = e_i, \\ f_i(\delta) = H(b_i), \\ f_i(\delta+1) = H(m_i||i), \end{cases}$$

ただし，e_i は素数となるまで乱数 b_i を繰り返し生成し求めるここで，H はハッシュ関数である．

Step 2: $a^{e_1 \cdots e_n} \bmod N$ を求め，署名者の秘密鍵 sk を用いて，署名

$$\sigma = \mathrm{Sign}_{sk}(a^{e_1 \cdots e_n} \bmod N)$$

を生成する．

出力 署名 σ，部分文書 $m_1, \ldots, m_n$，乱数 $b_1, \ldots, b_n$，コミットメント $\alpha_0 = a$，定数 δ，条件 $M = \{1, \ldots, n\}$，$S = D = \phi$

ここで M, S, D は各部分文書の開示条件を表すインデックス集合であり，M は開示 (強制開示も墨塗りもされていない状態)，S は墨塗り，D は強制開示を表す．ただし，D は追加された順番も保持しているとする．

6.2.4 墨塗り

入力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，定数 δ，条件 M, S, D

- i 番目を墨塗り

  Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$S \leftarrow S \cup \{i\}$ と更新する．

Step 2: m_i と b_i から，性質 8 を用いて，

$$\alpha_0 \leftarrow \alpha_0^{e_i}$$

と更新する．

Step 3: m_i, b_i を削除する．

- i 番目を強制開示

Step 1: 条件を $M \leftarrow M \setminus \{i\}$，$D \leftarrow D \cup \{i\}$ にする．

Step 2: 性質 8 を用いて，$\{m_j | j \in M\}$ と $\{b_j | j \in M\}$，α_0 から，

$$\alpha_i \leftarrow a^{\Pi_{j \in M \cup S} e_j H(b_i)}$$

を求める．

Step 3: b_i を削除する．

出力 署名 σ，部分文書 $\{m_i | i \in M \cup D\}$，乱数 $\{b_i | i \in M\}$，コミットメント $\{\alpha_i | i \in D \cup \{0\}\}$，定数 δ，条件 M，S，D

墨塗りを繰り返した後，最終状態を M^*, S^*, D^*, ${\alpha_0}^*$ とおく．

6.2.5 署名検証

入力 署名 σ，部分文書 $\{m_i | i \in M^* \cup D^*\}$，乱数 $\{b_i | i \in M^*\}$，コミットメント ${\alpha_0}^*$, $\{\alpha_i | i \in D^*\}$，定数 δ，条件 M^*，S^*，D^*，署名者の公開鍵 pk

Step 1: 検証用コミットメント α^* の初期値を ${\alpha_0}^*$ に設定する．$i \in M^*$ について，m_i と b_i と α^* を用い，性質 8 より，

$$\alpha^* \leftarrow (\alpha^*)^{e_i} = (\alpha^*)^{(\delta+1)H(b_i)} / (\alpha^*)^{\delta H(m_i||i)}$$

を求め，α^* を更新する．従って最終的には，$\alpha^* = a^{\Pi_{j \in M^* \cup S^*} e_j}$ が得られる．

Step 2: $i \in D^*$ について，m_i と α_i と α^* を性質 8 に適用して，

$$\alpha^* \leftarrow (\alpha_i)^{(\delta+1)} / (\alpha^*)^{\delta H(m_i||i)}$$

を求め，α^* を更新する．ただし，i は，D に追加された順序とは逆順序で処理されていくとする．したがって最終的に

$$\alpha^* = a^{e_1 e_2 \cdots e_n} \bmod N$$

を得る．

Step 3: 署名者の公開鍵 pk を用い,

$$\mathrm{Verify}_{pk}(\sigma, \alpha^*) \stackrel{?}{=} \mathrm{valid}$$

を検証する.

6.2.6 計算例

δ 方式（UK-δ）の処理例を図 6.5 に示す．部分文書数 $n = 6$ ，強制開示 $D = (m_6, m_4)$, 墨塗り $S = \{m_2, m_3\}$ の場合の署名処理，墨塗り処理，及び，検証処理の手順を示す.

6.3 評価

6.3.1 パフォーマンス

Dual Accumulator 方式と δ 方式における検証者が保持する署名データサイズによる評価を表 6.1 に，べき乗計算の数による計算量の評価を表 6.2 に示す．比較対象として，SUMI-5[1] と SUMI-6[2]，UK2[12] を用いる．ここで n は部分文書数，k は墨塗り部分文書数，ℓ は強制開示部分文書数を表す．δ 方式は UK2[12] と同じ，Dual Accumulator 方式は δ 方式の 2 倍の通信量と計算量がかかることが示された.

表 6.1: 検証者が保持する署名データの比較

	SUMI-5[1]	SUMI-6[2]	UK2[12]	DA 方式	δ 方式
文書	$n-k$	$n-k$	$n-k$	$n-k$	$n-k$
コミットメント	n	0	$\ell+1$	$5\ell+2$	$\ell+1$
乱数	$n-k, n-\ell$	0	$n-k-\ell$	$n-k-\ell$	$n-k-\ell+1$
署名	1	$1+n-k-\ell$	1	1	1
署名長	$\mathcal{O}(n)$	$\mathcal{O}(n-k)$	$\mathcal{O}(n-k)$	$\mathcal{O}(n-k)$	$\mathcal{O}(n-k)$

署名	部分文書 m_i	m_1 m_2 m_3 m_4 m_5 m_6
	乱数 b_i	b_1 b_2 b_3 b_4 b_5 b_6
	コミットメント α_i	a
	インデックス集合	$M = \{1, 2, 3, 4, 5, 6\}$, $S = \phi$, $D = \phi$
m_2 墨塗り	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 b_4 b_5 b_6
	コミットメント α_i	a^{e_2}
	インデックス集合	$M = \{1, 3, 4, 5, 6\}$, $S = \{2\}$, $D = \phi$
m_6 強制開示	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 b_4 b_5 $\not{b}_6$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 4, 5\}$, $S = \{2\}$, $D = (6)$
m_4 強制開示	部分文書 m_i	m_1 $\not{m}_2$ m_3 m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ b_3 $\not{b}_4$ b_5 $\not{b}_6$
	コミットメント α_i	a^{e_2} $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 3, 5\}$, $S = \{2\}$, $D = (6, 4)$
m_3 墨塗り	部分文書 m_i	m_1 $\not{m}_2$ $\not{m}_3$ m_4 m_5 m_6
	乱数 b_i	b_1 $\not{b}_2$ $\not{b}_3$ $\not{b}_4$ b_5 $\not{b}_6$
	コミットメント α_i	$a^{e_2e_3}$ $a^{e_1e_2e_3e_5H(b_4)}$ $a^{e_1e_2e_3e_4e_5H(b_6)}$
	インデックス集合	$M = \{1, 5\}$, $S = \{2, 3\}$, $D = (6, 4)$
検証	m_1	$a^{e_1e_2e_3} = (a^{e_2e_3})^{(\delta+1)H(b_1)} / (a^{e_2e_3})^{\delta H(m_1\|\|1)}$
	m_5	$a^{e_1e_2e_3e_5} = (a^{e_1e_2e_3})^{(\delta+1)H(b_5)} / (a^{e_1e_2e_3})^{\delta H(m_5\|\|5)}$
	m_4	$a^{e_1e_2e_3e_4e_5} = (a^{e_1e_2e_3e_5H(b_4)})^{(\delta+1)} / (a^{e_1e_2e_3e_5})^{\delta H(m_4\|\|4)}$
	m_6	$a^{e_1e_2e_3e_4e_5e_6} = (a^{e_1e_2e_3e_4e_5H(b_6)})^{(\delta+1)} / (a^{e_1e_2e_3e_4e_5})^{\delta H(m_6\|\|6)}$

図 6.5: δ 方式 (UK-δ) の計算例

表 6.2: 計算量の比較

	UK2[12]	DA 方式	δ 方式
署名	$2n$	$4n$	$2n$
墨塗り	$2k$	$4k$	$2k$
開示	$(2n - 2k - \ell)\ell$	$(2n - 2k - \ell)2\ell + 4\ell$	$(2n - 2k - \ell)\ell$
検証	$2n - 2k - \ell$	$4n - 4k + 2\ell$	$2n - 2k - \ell$

第7章
結論と今後の課題

7.1 結論

本論文では，RSA Accumulator を用いた開示条件付き墨塗り署名について述べてきた．

第 4 章では RSA Accumulator を用いた墨塗り署名を提案した．RSA Accumulator を用いることにより，墨塗り部分文書のハッシュ値を束ねておくことが出来，ハッシュ値を保持する必要がないため，署名長を削減することができた．これにより，墨塗り部分文書のハッシュ値に関するデータは，常に墨塗り部分文書数に依存せず 1 個となる．

第 5 章では RSA Accumulator を用いた開示条件付き墨塗り署名を 2 つ提案した．従来のハッシュ値の連結に署名を施す方式や Aggregate 署名を使用する方式に比べ新しい要素技術である RSA Accumulator を用いた．強制開示の設定順序が検証可能であり，墨塗り部分文書のハッシュ値を束ねておくことが出来るため，署名長を削減できる．また第 5 章では，提案方式における問題点を指摘した．

第 6 章では，第 5 章で指摘した問題点を解決した方式を 2 つ提案した．また，パフォーマンスにおいて評価した．Dual Accumulator 方式は δ 方式の約 2 倍の通信量と計算量がかかることが分かった．δ 方式のパフォーマンスは第 5 章で提案した方式とほぼ同じである．

7.2 今後の課題

第 5 章において提案方式の問題点を挙げ，第 6 章では問題点を解決する方式を提案したが，そのほかに提案方式における問題点があるかどうかの検討や，詳細な安全性の評価が今後の課題となる．また，署名データサイズや計算量の点において，より一層効率のよい方式の提案や，実装を行い実際に計算にかかる時間の計測など有用性の検討も今後の課題として挙げられる．

参 考 文 献

[1] 宮崎 邦彦，岩村 充，松本 勉，佐々木 良一，吉浦 裕，手塚 悟，今井 秀樹．“開示条件を制御可能な電子文書墨塗り技術”．*2004* **年暗号と情報セキュリティシンポジウム** *(SCIS2004)*, pages 515–520, 2004.

[2] 宮崎 邦彦，花岡 悟一郎，今井 秀樹．“双線形写像を用いた電子文書墨塗り技術”．*2005* **年暗号と情報セキュリティシンポジウム** *(SCIS2005)*, 2005.

[3] J. Benaloh and M. de Mare. “One-way accumulators: A decentralized alternative to digital signatures”. *EUROCRYPT*, LNCS 765:274–285, 1994.

[4] Ron Steinfeld, Laurence Bull, and Yuliang Zheng. “Content Extraction Signatures”. *ICICS 2001*, LNCS 2288:285–304, 2001.

[5] 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”．**電子情報通信学会技術研究報告** *ISEC-20* **情報セキュリティ**, pages 61–67, 2003.

[6] 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識別可能な電子文書の墨塗り方式”．**コンピュータセキュリティシンポジウム** *2004 (CSS2004)*, pages 475–480, 2004.

[7] 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性保証方式の改良 (その 2)”．*2006* **年暗号と情報セキュリティシンポジウム** *(SCIS2006)*, 2006.

[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”. *EUROCRYPT*, LNCS 2656:416–432, 2003.

[9] J. Camenisch and A. Lysyanskaya. “Dynamic accumulators and application to efficient revocation of anonymous credentials”. *CRYPTO 2002*, 2002.

[10] Robert Johnson, David Molnar, Dawn Song, and David Wagner. “Homomorphic Signature Schemes”. *CT-RSA 2002*, LNCS 2271:244–262, 2002.

[11] 上山 真梨，菊池 浩明．“RSA Accumulator を用いた開示条件付き墨塗り署名方式”．**コンピュータセキュリティシンポジウム** *2007 (CSS2007)*, 2007.

[12] 上山 真梨，菊池 浩明. “RSA Accumulator を用いた開示条件付き墨塗り署名方式の改良”, 2007. preprint,(http://www.cs.dm.u-tokai.ac.jp/~ ueyama/css2007uk2.pdf).

[13] 伊豆 哲也，武仲 正彦，上山 真梨，菊池 浩明. “RSA Accumulator を用いた墨塗り署名方式について”. *2008* **年暗号と情報セキュリティシンポジウム** *(SCIS2008)*, 2008.

[14] 上山 真梨，菊池 浩明，伊豆 哲也，武仲 正彦. “RSA Accumulator を用いた開示条件付き墨塗り署名方式の改良”. *2008* **年暗号と情報セキュリティシンポジウム** *(SCIS2008)*, 2008.

[15] R. Cramer, I. Damgård, and B. Schoenmakers. “Proofs of Partial Knowledge and Simplefied Design of Witness Hiding Protocols”. *Advances in Cryptology CRYPTO'94*, LNCS 839:174–187, 1994.

謝辞

本論文を遂行するにあたり多くの方々から多大なる御指導と御援助を賜りました.

特に，本論文を完成に導いていただき，また，研究にかかわらず私を導いて下さった東海大学情報理工学部情報メディア学科 菊池 浩明 教授には深く感謝を申し上げます.

また，本研究を推進するにあたって，御親切なる御教示ならびに御激励を賜りました東海大学情報理工学部情報科学科 中西 祥八郎 教授，東海大学情報理工学部情報科学科 内田 理 准教授に厚く感謝申し上げます.

さらに，2 年間共に楽しみ，苦しみ，励まし合い，時には研究に対して有益な意見を与えてくれた東海大学大学院工学研究科情報理工学専攻 石原 進一 氏，菅 秀俊 氏，甲田 博揮 氏，篠原 学 氏，富沢 和也 氏，永井 慧 氏，松本 幹 氏，吉村 武 氏に深く感謝致します.

その他にも，様々な行事など，共に協力してきた電気第 5 研究室の一同に感謝致します.

また，活発な議論，技術的な御教示を頂いた富士通研究所 伊豆 哲也 氏，武仲 正彦 氏に深く感謝を申し上げます.

最後に，両親に心より感謝致します.