

RSA Accumulator を用いた開示条件付き墨塗り署名方式の改良

上山 真梨†

菊池 浩明‡

† 東海大学大学院工学研究科
259-1292 神奈川県平塚市北金目 1117

‡ 東海大学情報理工学部情報メディア学科
259-1292 神奈川県平塚市北金目 1117

あらまし 墨塗り署名は、文書の一部分を墨塗り（秘匿）した電子署名の完全性を保証する署名方式である。文書の指定された部分の墨塗りを禁止する方式として、宮崎らが提案した SUMI-5[2] がある。本稿では、RSA Accumulator を用いた開示条件付きの墨塗り署名方式を提案する。署名長が墨塗りされた部分文書数に依存しないため、提案方式は従来方式より効率的である。

Improvement of Sanitizable Signature Scheme with Condition to Disclose Using RSA Accumulator

Mari Ueyama†

Hiroaki Kikuchi‡

† Graduate School of Engineering, Tokai University
1117 Kitakanam, Hiratsuka, Kanagawa 259-1292, Japan

‡ Department of Information Media Technology, Tokai University
1117 Kitakanam, Hiratsuka, Kanagawa 259-1292, Japan

Abstract Sanitizable signature is a signature scheme which guarantees the integrity of partially disclosed document. Miyazaki et al. proposed a scheme, called SUMI-5[2], that allows denying specified parts of document. We propose a new sanitizable signature scheme with condition to disclose using RSA accumulator. Our scheme is more efficient than the conventional scheme in a sense that the size of signature is independent from a number of sanitized subdocuments.

1 はじめに

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 [5][1]。

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 [5][1][4][6]。開示条件を設定可能な方式には、多項式補間を用いた [2] や Aggregate 署名を用いた [3][6] 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 [2] に対して、Aggregate 署名を用いた [3] 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の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、本稿では、宮崎らが提案した SUMI-

5[2] をベースに、複数の素数を単一の証拠 (witness) で効率よく証明可能な RSA Accumulator[8] を用いて開示条件を設定可能な墨塗り署名方式を提案する。Accumulator は Benaloh らによって初めて [8] 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 [7] に拡張されたりしている。提案方式は、この技術を応用することで、Aggregate 署名 (双線形写像) を用いることなく、[3] と同等の $O(n-k)$ のサイズの効率的な署名長を実現している。

2 墨塗り署名

2.1 モデル

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。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、本稿では墨塗り者は複数人いるので、 i 番目の墨塗り者が開示した部分を $i+1$ 以降の墨塗り者が追加墨塗りする可能性がある。そこで SUMI-5[2] では以降の墨塗りを禁止する設定を可能にしている。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。

2.2 従来方式 SUMI-5[2]

2.2.1 署名生成

入力：文書 m_1, \dots, m_n 、署名者の秘密鍵 sk

- 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¹ 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 H はハッシュ関数である。
- 署名者の秘密鍵 sk を用いて、署名 $\sigma = \text{Sign}_{sk}(e_1 || \dots || e_n || c_1 || \dots || c_n)$ を生成する。

¹有限体上の一次多項式

出力：署名 σ 、文書 m_1, \dots, m_n 、乱数 r_1, \dots, r_n 、 b_1, \dots, b_n 、補助点 c_1, \dots, c_n

2.2.2 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、 $M = \{1, \dots, n\}$ 、 $S = D = \phi$ である。 n は総部分文書数とする。

S ： 墨塗り

M ： 開示かつ追加墨塗り可能

D ： 開示かつ追加墨塗り不可 (強制開示)

入力：署名 σ 、文書集合 $\{m_i | i \in M \cup D\}$ 、乱数集合 $\{r_i | i \in M \cup D\}$ 、 $\{b_i | i \in S \cup M\}$ 、補助点 c_1, \dots, c_n 、条件 S, M, D

$i \in M$ について、条件の変化に応じ次の処理を行う。

1. 墨塗り

m_i, r_i を削除し、 $M' = M \setminus \{i\}$ 、 $S' = S \cup \{i\}$ と更新する。

2. 強制開示

b_i を削除し、 $M' = M \setminus \{i\}$ 、 $D' = D \cup \{i\}$ と更新する。

最終状態の条件を各々、 S^*, M^*, D^* とおく。また、墨塗り文書数 $k = |S^*|$ 、強制開示文書数 $\ell = |D^*|$ とおく。

出力：署名 σ 、文書 $\{m_i | i \in M^* \cup D^*\}$ 、乱数 $\{r_i | i \in M^* \cup D^*\}$ 、 $\{b_i | i \in S^* \cup M^*\}$ 、補助点 $\{c_1, \dots, c_n\}$ 、条件 S^*, M^*, D^*

2.2.3 署名検証

入力：墨塗り処理の出力と同一

- $i = 1, \dots, n$ に対し、入力文書集合に 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。

2. 署名者の pk を用いて, $\text{Verify}_{pk}(\sigma) \stackrel{?}{=} (e'_1 || \dots || e'_n || c_1 || \dots || c_n)$ が成り立つか検証する.

SUMI-5 は乱数が文書と無関係に生成されることから墨塗り秘匿性と, (元の) 署名方式の適応的選択文書攻撃に対して存在的偽造不能とハッシュ関数の衝突困難性の仮定の下で, 墨塗り偽造不能性を満たすことが示されている [2].

3 提案方式

3.1 概要

提案方式は, SUMI-5 をベースに, RSA Accumulator を用いた開示条件を設定可能な墨塗り署名方式である. SUMI-5 は, 2点から e_i を求め, 検証を行っているため, 必ず2点を保持しなければならない. 加えて, 検証時に $H(m_i || r_i)$ も開示するので, 墨塗り秘匿性を満たすためには, 辞書攻撃を受けないように乱数 r_i が必要である.

そこで提案方式では, 墨塗り指定されている文書集合に対応する全ての e_i を RSA Accumulator を用い, 束ねておくことにより, 署名長の削減を試みる. また SUMI-5 で2組必要だった乱数 r_i, b_i を単一に減らす.

3.2 RSA Accumulator

RSA アキュムレータ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. 最初の実現方法は, Benaloh と de Mare [8] によるもので, 次に示すように RSA 暗号に基づいている.

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 $L = \{e_1, \dots, e_n\}$ について, アキュムレータ A を

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

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²を用いて A に署名を行ない, それを $\sigma(A)$ とする. 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,

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

で定められる証拠 (witness) A_i を計算する. 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.

3.3 基本性質

一次多項式 $f(x)$ について, $f(0) = e, f(1) = H(b), f(2) = H(m)$ の関係がある時, 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.

性質 1 $H(b)$ 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 2 $a \pmod{N}$ と $H(m)$ と $H(b)$ から,

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

3.4 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: 文書 m_1, \dots, m_n , 署名者の秘密鍵 sk

- 各部分文書 m_i に対し, 2点 $(1, H(b_i)), (2, H(m_i || i))$ を通る一次多項式 $f_i(x)$ が $e_i = f_i(0)$ を素数となるように, 乱数 b_i を決める. ここで, H はハッシュ関数である.

²例えば, RSA 署名を用いれば, $\sigma(A) = A^{1/e_1 1/e_2 \dots 1/e_n}$

2. 署名者の秘密鍵 sk を用いて, 署名

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

を生成する.

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

3.5 墨塗り

強制開示の順番を保持するインデックス X を定義する.

入力: 署名 σ , 文書 $\{m_i | i \in M \cup D\}$, 乱数 $\{b_i | i \in M\}$, コミットメント集合 $\{\alpha_i | i \in D \cup \{0\}\}$, 条件を表すインデックス S, M, D , 強制開示の順番 $X = (x_1, \dots, x_\ell)$

- i 番目を墨塗り

条件を $M' = M \setminus \{i\}$, $S' = S \cup \{i\}$ にする.
 m_i と b_i から, 性質 2 を用いて, $\alpha'_0 = \alpha_0^{e_i}$
を求めて更新する. m_i, b_i を削除する.

- i 番目を強制開示

条件を $M' = M \setminus \{i\}$, $D' = D \cup \{i\}$ にする.
性質 2 を用いて, $\{m_j | j \in M'\}$ と $\{b_j | j \in M'\}$, α_0 から, $\alpha_i = a^{\prod_{j \in S \cup M'} e_j H(b_j)}$
を求める. 強制開示の順番を保持するインデックス $X = (x_1, \dots, x_\ell)$ を $x_{\ell+1} = i$
として, $X' = (x_1, \dots, x_\ell, x_{\ell+1})$ と更新する.
 b_i を削除する.

出力: 署名 σ , 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 , $\{\alpha_i | i \in D'\}$, 条件 S', M', D' , 強制開示の順番 X'

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

3.6 署名検証

入力: 署名 σ , 文書 $\{m_i | i \in M^* \cup D^*\}$, 乱数 $\{b_i | i \in M^*\}$,
コミットメント集合 α_0 , $\{\alpha_i | i \in D^*\}$, 条件 S^*, M^*, D^* , 強制開示の順番 $X^* = (x_1, \dots, x_\ell)$

1. 検証用コミットメント α^* の初期値を α_0 に設定する. $i \in M^*$ について, m_i と b_i と α^* を用い, 性質 2 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, $\alpha^* = a^{\prod_{j \in M^* \cup S^*} e_j}$ が得られる.

2. X^* を用い, 強制開示の設定を行った順序の逆順序, $i = \ell, \dots, 1$ について, m_i と α_i と α^* を性質 2 に適用して,

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

を求め, α^* を更新する. したがって最終的に $a^{e_1 e_2 \cdots e_n} \bmod N$ を得る.

3. 署名者の公開鍵 pk を用い, $\text{Verify}_{pk}(\sigma, a^{e_1 \cdots e_n} \bmod N) \stackrel{?}{=} \text{valid}$ を検証する.

4 計算例

$n = 6$, 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, 検証処理の手順を図 1 に示す.

5 評価

5.1 署名データ

提案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行う (表 1). 比較対象として, SUMI-5[2] と SUMI-6[3] を用いる. また, 開示条件の設定は出来ないが基本的な SUMI-4[1] も参照する. 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.

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. 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ものに対して, 提案方式は 1 個の署名でよい.

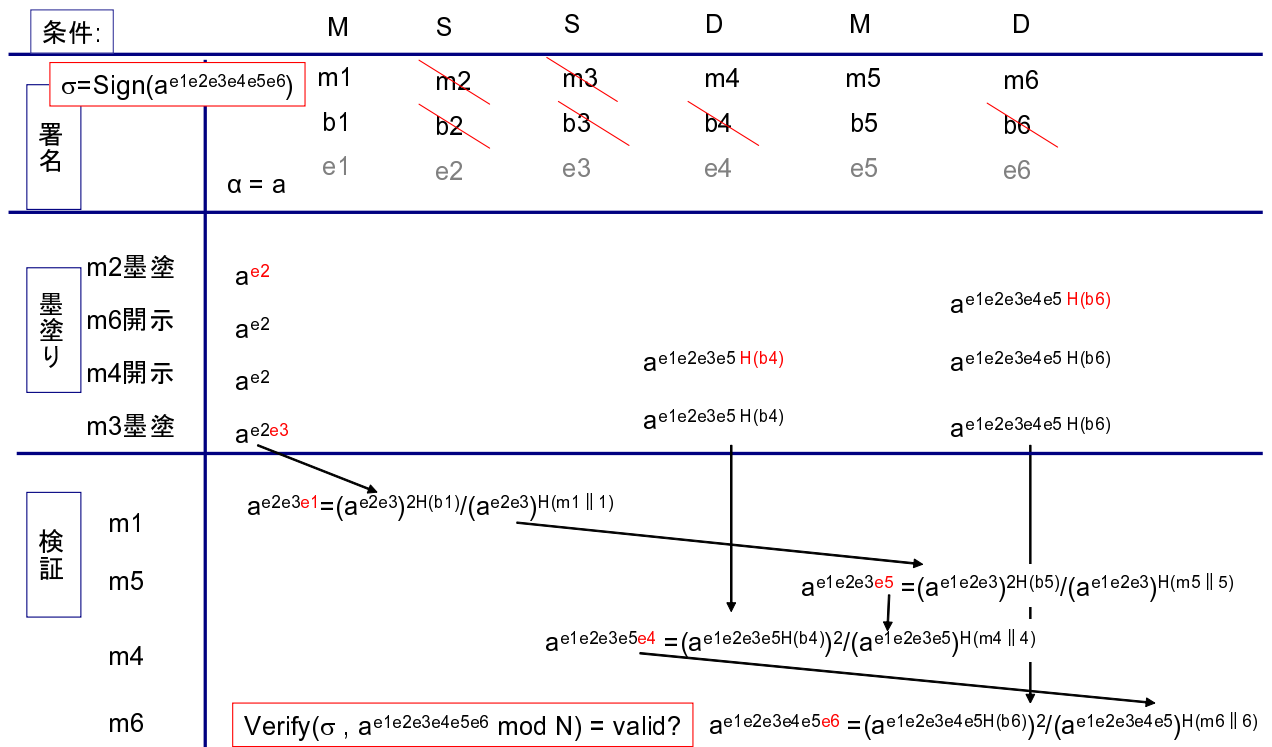


図 1: 提案方式の計算例

表 1: 検証者が保持する署名データの比較

	SUMI-4[1]	SUMI-5[2]	SUMI-6[3]	提案方式
文書	$n - k$	$n - k$	$n - k$	$n - k$
コミットメント (ハッシュ値)	k	n	0	$\ell + 1$
乱数	$n - k$	$n - k, n - \ell$	0	$n - k - \ell$
署名	1	1	$1 + n - k - \ell$	1
署名長	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n - k)$	$\mathcal{O}(n - k)$

6 おわりに

RSA Accumulator を用いた開示条件を設定可能な墨塗り署名方式を提案した。また、提案方式において署名データの点において評価をした。今後の課題は、詳細な安全性の評価を行うことである。

参考文献

[1] 宮崎, 洲崎, 岩村, 松本, 佐々木, 吉浦, “電子文書墨塗り問題”, 信学技報, ISEC 2003-20, pp. 61-67, 2003.

[2] 宮崎, 岩村, 松本, 他, “開示条件を制御可能な電子文書墨塗り技術”, SCIS 2004, 2D3-2, pp. 515-520, 2004.

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

[4] T. Izu, N. Kanaya, M. Takenaka, and T. Yoshioka, “PIATS: A Partially Sanitizable Signature Scheme,” ICICS 2005, LNCS 3783, pp. 72-83, 2005.

[5] R. Steinfeld, L. Bull, and Y. Zheng, “Content Extraction Signatures”, ICICS 2001, LNCS 2288, pp. 285-304, 2001.

- [6] 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, 2C4-3, 2007.
- [7] R. Johnson, D. Molnar, D. Song, and D. Wagner, “Homomorphic Signature Schemes”, CT-RSA 2002, LNCS 2271, pp. 244-262, 2002.
- [8] J. Benaloh, and M. de Mare, “One-way accumulators: A decentralized alternative to digital signatures,” EUROCRYPT, LNCS 765, Springer, pp. 274-285, 1994.
- [9] J. Camenisch, and A. Lysyanskaya, “Dynamic accumulators and application to efficient revocation of anonymous credentials,” CRYPTO 2002, 2002.
- [10] 伊豆, 國廣, 太田, 武仲, “墨塗り署名方式の安全性について”, SCIS 2006, 4A1-4, 2006.