SCIS 2007 The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

メッセージ制限付き署名の新しいモデルと構成法

Restricted message signing : a new security model and construction

小山 拓郎* 尾形 わかは*
Takuro Oyama Wakaha Ogata

あらまし ディジタル署名は，電子情報に対して正当性を保障する機能を有するだけでなく，署名者の匿名性を保障する方式や，複数人の署名を効率的に扱うことができる方式など，様々な付加機能を持つ署名が提案されている．近年，松尾によりディジタル署名の新しい概念として，メッセージ制限付き署名が提案された．メッセージ制限付き署名では，署名者は特定のメッセージに対して署名をすることができない．また，第三者は公開鍵から署名を制限されているメッセージに関して何の情報も得ることはできない．松尾の方式では多項式と Boneh, Boyen のテクニックを用いてメッセージ制限付き署名を実現させている．本稿では，メッセージ制限付き署名の新しいモデルを提案し，提案するメッセージ制限付き署名が通常のディジタル署名と，ある条件を満たすキーワード検索付き公開鍵暗号を組み合わせて構成できることを示す．また，本方式と松尾の方式を比較し，考察する．

キーワード メッセージ制限付き署名，キーワード検索付き公開鍵暗号

1 はじめに

ディジタル署名には，電子情報に対して正当性を保障する機能を有するだけでなく，署名者の匿名性を保障する方式や，複数人の署名を効率的に扱うことができる方式など，様々な付加機能を持つ署名が存在する．近年，松尾により新しい概念として，メッセージ制限付き署名が提案された [6]．メッセージ制限付き署名では，署名者は特定のメッセージに対して署名をすることができない．また，第三者は公開鍵から署名を制限されているメッセージに関して何の情報も得ることはできない．[6] の方式では多項式と [1] のテクニックを用いてメッセージ制限付き署名を実現させている．

本稿ではメッセージ制限付き署名の新しいモデルを提案し，提案するメッセージ制限付き署名が，通常のディジタル署名と，ある条件を満たすキーワード検索付き公開鍵暗号を組み合わせて構成できることを示す．[6] の方式では信頼できる第三者 (CA) がメッセージ制限付き署名の公開鍵と秘密鍵を生成し，署名者に渡す．提案する方式では，署名者と CA が協力して鍵生成を行う．提案方式では，署名者は通常のディジタル署名の署名とキーワード検索付き公開鍵暗号の落とし戸のペアをメッセージ制限付き署名の署名とする．ただし，キーワード検索付き公開鍵暗号に対して署名者が落とし戸を正しく生成したことを効率良く検証できる方法が存在することが条件となる．本稿の構成は，まず 2 節でディジタル署名の安全性，キーワード検索付き公開鍵暗号の構成とその安全性について紹介する．3 節でメッセージ制限付き署名のモデルとその安全性について述べる．4 節で提案するメッセージ制限付き署名を構成する方法を示し，その安全性を証明する．5 節で，[6] の方式と提案方式について比較，考察する．6 節で提案方式に具体的なキーワード検索付き公開鍵暗号とディジタル署名を適用した例を示し，効率性を比較する．

2 準備

2.1 ディジタル署名の偽造不可能性

ディジタル署名 $\mathcal{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ の安全性は敵 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ による以下のゲームによって定義される．

準備. $\mathcal{C}$ は，$\mathsf{KeyGen}(1^k)$ を実行し，公開鍵と秘密鍵 $(\mathsf{pk}, \mathsf{sk})$ を得て，pk を $\mathcal{A}$ に与える．

署名オラクル. $\mathcal{A}$ はメッセージ M_i を出力する．$\mathcal{C}$ は $\mathsf{Sign}(\mathsf{sk}, M_i)$ を実行して署名 σ_i を生成し，$\mathcal{A}$ に σ_i を返す．

偽造. $\mathcal{A}$ は偽造 (M', σ') を出力する．

$\mathcal{A}$ が有効な偽造を出力すれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}^{uf-cma}_{\mathcal{DS}}(\mathcal{A}) = \Pr[M' \neq M_i, \mathsf{Verify}(\mathsf{pk}, M', \sigma') = 1]$$

* 東京工業大学，〒 152-8550 東京都目黒区大岡山 2-12-1, Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8550, Japan, taku-zy@crypt.ss.titech.ac.jp, wakaha@mot.titech.ac.jp

定義 2.1. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}_{\mathcal{DS}}^{uf-cma}(\mathcal{A})$ が無視できるほど小さいならば，$\mathcal{DS}$ は偽造不可能である．

2.2 キーワード検索付き公開鍵暗号

キーワード検索付き公開鍵暗号 (以下，PEKS)[2][4][5] は，受信者が秘密鍵とキーワード M から作った秘密情報 (落とし戸) をサーバに渡しておくことにより，暗号化されたキーワード M' が M と一致しているかをチェックする能力をサーバに持たせることができる．$M \neq M'$ ならば，サーバは M' についてそれ以上の情報を得ることができない．

PEKS は，以下の 4 つのアルゴリズムから構成される．

- $\mathsf{KeyGen}^{\mathrm{PEKS}}$: セキュリティパラメータ k を入力とし，公開鍵と秘密鍵のペア $(\mathsf{pk}^{\mathrm{PEKS}}, \mathsf{sk}^{\mathrm{PEKS}})$ を生成する．
- Trapdoor : 受信者の秘密鍵 $\mathsf{sk}^{\mathrm{PEKS}}$ とキーワード M を入力とし，落とし戸 T_M を出力する．
- PEKS : 受信者の公開鍵 $\mathsf{pk}^{\mathrm{PEKS}}$ とキーワード M を入力とし，M の検索可能な暗号文 C を出力する．
- Test : 受信者の公開鍵 $\mathsf{pk}^{\mathrm{PEKS}}$ と暗号文 $C = \mathsf{PEKS}(\mathsf{pk}^{\mathrm{PEKS}}, M')$ と落とし戸 $T_M = \mathsf{Trapdoor}(\mathsf{sk}^{\mathrm{PEKS}}, M)$ を入力とし，$M = M'$ ならば 1 を，そうでなければ 0 を出力する．

PEKS の安全性は [2] において，選択キーワード攻撃に対する識別不可能性が定義されている．選択キーワード攻撃に対する識別不可能性は敵 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ による以下のゲームによって定義される．

準備. $\mathcal{C}$ は，$\mathsf{KeyGen}(1^k)$ を実行し，$(\mathsf{pk}^{\mathrm{PEKS}}, \mathsf{sk}^{\mathrm{PEKS}})$ を得て，$\mathsf{pk}^{\mathrm{PEKS}}$ を $\mathcal{A}$ に与える．

フェーズ 1. $\mathcal{A}$ は任意のキーワード $M_i \in \{0,1\}^*$ を $\mathcal{C}$ に送る．$\mathcal{C}$ は M_i に対する落とし戸 $T_{M_i} = \mathsf{Trapdoor}(\mathsf{sk}^{\mathrm{PEKS}}, M_i)$ を生成し，$\mathcal{A}$ に返す．

チャレンジ. $\mathcal{A}$ は 2 つのキーワード M_0', M_1' を $\mathcal{C}$ に送る．$\mathcal{C}$ は $b \in \{0,1\}$ をランダムに選び，M_b' に対する暗号文 $C = \mathsf{PEKS}(\mathsf{pk}^{\mathrm{PEKS}}, M_b')$ を生成し，$\mathcal{A}$ に送る．ただし，M_0', M_1' はフェーズ 1 でその落とし戸を得ていないものとする．

フェーズ 2. フェーズ 1 と同様に，$\mathcal{A}$ は $M_i \in \{0,1\}^*$ を送り，$\mathcal{C}$ からその落とし戸 T_{M_i} を得ることができる．ただし，$M_i \notin \{M_0', M_1'\}$．

出力. $\mathcal{A}$ は b の推測値 $b' \in \{0,1\}$ を出力する．

$\mathcal{A}$ が b を正しく推測できれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}_{\mathrm{PKES}}^{ind-cka}(\mathcal{A}) = |\Pr[b'=1|b=1] - \Pr[b'=1|b=0]|$$

定義 2.2. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}_{\mathrm{PEKS}}^{ind-cka}(\mathcal{A})$ が無視できるほど小さいならば，PEKS は選択キーワード攻撃に対して識別不可能である．

3 メッセージ制限付き署名

3.1 モデル

メッセージ制限付き署名 (以下，RMS) は [6] によって提案された．RMS は公開鍵と秘密鍵を生成する CA と，秘密鍵を用いて署名を生成する署名者と，公開鍵を用いて署名の検証を行う検証者で構成される．署名者が，自分が任意に選んだメッセージの集合に対して生成した公開鍵を署名禁止メッセージに対する公開鍵であるとみせかけるの防ぐために，CA は独自にディジタル署名 $\mathcal{DS}^{\mathrm{CA}} = (\mathsf{KeyGen}^{\mathrm{CA}}, \mathsf{Sign}^{\mathrm{CA}}, \mathsf{Verify}^{\mathrm{CA}})$ を持ち，RMS の公開鍵に対して CA の署名 θ をつける．
RMS は $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ から構成される．

- KeyGen : セキュリティパラメータと署名を禁止するメッセージの集合 S を用いて RMS の公開鍵と秘密鍵 $(\mathsf{pk}, \mathsf{sk})$ と pk に対する CA の署名 θ を生成する．署名者は $(\mathsf{pk}, \mathsf{sk}, \theta)$ を得る．
- Sign : sk とメッセージ M を入力として，署名 σ を出力するアルゴリズム．
- Verify : $\mathsf{pk}^{\mathrm{CA}}$, pk,$\theta$, M,σ を入力とし，1 または 0 を出力するアルゴリズム．

CA は前もって $\mathsf{KeyGen}^{\mathrm{CA}}(1^k)$ を実行し，自身の署名の鍵 $(\mathsf{pk}^{\mathrm{CA}}, \mathsf{sk}^{\mathrm{CA}})$ を生成する．まず CA は KeyGen を実行し，RMS の鍵 $(\mathsf{pk}, \mathsf{sk})$ を生成する．また CA は $\mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$ を実行して署名 θ を生成する．署名者は sk を用いて $\mathsf{Sign}(\mathsf{sk}, M)$ を実行し，署名 σ を生成する．検証者は $\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}$ を用いて $\mathsf{Verify}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta, M, \sigma)$ を実行し，署名を検証する．

3.2 安全性の定義

[6] において，メッセージ制限付き署名の安全性は以下の 3 つが定義されている．

- **プライバシー**

  CA と署名者以外の誰も，公開鍵から署名を制限されているメッセージに関する情報を得ることができない．

- **署名者に対する偽造不可能性**

  署名者は制限されているメッセージに対して，有効な署名を生成することができない．

- **偽造不可能性**

  署名者以外の誰も，有効な署名とメッセージのペアを生成することができない．

3.2.1 プライバシー

RMS のプライバシーは以下のゲームによって定義される．

準備． $\mathcal{C}$ は $\mathsf{KeyGen}^{\mathrm{CA}}(1^k)$ を実行し，$(\mathsf{pk}^{\mathrm{CA}}, \mathsf{sk}^{\mathrm{CA}})$ を得て，$\mathsf{pk}^{\mathrm{CA}}$ を $\mathcal{A}$ に与える．

チャレンジ． $\mathcal{A}$ は同じ要素数を持つ２つのメッセージの集合 $S_0 = \{M_1, \ldots, M_n\}$，$S_1 = \{M'_1, \ldots, M'_n\}$ を選び，(S_0, S_1) を $\mathcal{C}$ に送る．$\mathcal{C}$ は $b \in \{0,1\}$ をランダムに選び，KeyGen を実行し，S_b に対する $(\mathsf{pk}, \mathsf{sk})$ と $\theta = \mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$ を生成し，(pk, θ) を $\mathcal{A}$ に送る．

署名オラクル． $\mathcal{A}$ はメッセージ $M_i \notin S_0 \cup S_1$ を $\mathcal{C}$ に送る．$\mathcal{C}$ は M_i に対する署名 $\sigma_i = \mathsf{Sign}(\mathsf{sk}, M_i)$ を生成し，$\mathcal{A}$ に返す．

出力． $\mathcal{A}$ は b の推測値 $b' \in \{0,1\}$ を出力する．

$\mathcal{A}$ が b を正しく推測できれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}^{privacy}_{\mathrm{RMS}}(\mathcal{A}) = |\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]|$$

定義 3.1. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}^{privacy}_{\mathrm{RMS}}(\mathcal{A})$ が無視できるほど小さいならば，RMS はプライバシーを満たす．

3.2.2 署名者に対する偽造不可能性

RMS の署名者に対する偽造不可能性は以下のゲームによって定義される．

準備． $\mathcal{C}$ は $\mathsf{KeyGen}^{\mathrm{CA}}(1^k)$ を実行し，$(\mathsf{pk}^{\mathrm{CA}}, \mathsf{sk}^{\mathrm{CA}})$ を得て，$\mathsf{pk}^{\mathrm{CA}}$ を $\mathcal{A}$ に与える．

チャレンジ． $\mathcal{A}$ は署名を禁止するメッセージの集合 S を選び，S を $\mathcal{C}$ に送る．$\mathcal{C}$ は KeyGen から $(\mathsf{pk}, \mathsf{sk})$ と署名 $\theta = \mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$ を生成し，$\mathcal{A}$ は $(\mathsf{sk}, \mathsf{pk}, \theta)$ を得る．

偽造． $\mathcal{A}$ は偽造 $(\mathsf{pk}', \theta', M', \sigma')$ を出力する．

$\mathcal{A}$ が有効な偽造を出力すれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}^{ufs}_{\mathrm{RMS}}(\mathcal{A}) = \Pr[\mathsf{Verify}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}', \theta', M', \sigma') = 1] \quad \text{ただし } M' \in S$$

定義 3.2. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}^{ufs}_{\mathrm{RMS}}(\mathcal{A})$ が無視できるほど小さいならば，RMS は署名者に対して偽造不可能である．

3.2.3 偽造不可能性

RMS の偽造不可能性は以下のゲームによって定義される．

準備． $\mathcal{C}$ は $\mathsf{KeyGen}^{\mathrm{CA}}(1^k)$ を実行し，$(\mathsf{pk}^{\mathrm{CA}}, \mathsf{sk}^{\mathrm{CA}})$ を得る．さらにメッセージ $\{M'_i\}_{1 \le i \le n}$ をランダムに選び，署名を禁止するメッセージの集合を $S = \{M'_1, \ldots, M'_n\}$ とする．$\mathcal{C}$ は KeyGen を実行し，$(\mathsf{pk}, \mathsf{sk})$ と署名 $\theta = \mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$ を生成し，$\mathcal{A}$ は $(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta)$ を得る．

署名オラクル． $\mathcal{A}$ はメッセージ M_i を $\mathcal{C}$ に送る．$\mathcal{C}$ は M_i に対する署名 $\sigma_i = \mathsf{Sign}(\mathsf{sk}, M_i)$ を生成し，$\mathcal{A}$ に返す．

偽造． $\mathcal{A}$ は偽造 $(\mathsf{pk}', \theta', M', \sigma')$ を出力する．

$\mathcal{A}$ が有効な偽造を出力すれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}^{gen}_{\mathrm{RMS}}(\mathcal{A}) = \Pr[\mathsf{Verify}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}', \theta', M', \sigma') = 1] \quad \text{ただし } M' \neq M_i$$

定義 3.3. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}^{gen}_{\mathrm{RMS}}(\mathcal{A})$ が無視できるほど小さいならば，RMS は偽造不可能である．

3.3 従来の方式

[6] の方式では多項式を利用して，署名者が署名できないメッセージを持つ機能を果たしている．また，ID-Based 暗号 [1] の証明のテクニックを利用して，公開鍵から署名を制限されているメッセージに関して情報が漏れないことを証明している．

[6] の方式では信頼できる第三者 (CA) が RMS の公開鍵と秘密鍵を生成し，署名者に渡す．そのため CA は RMS の署名を生成することができ，悪用できてしまうという問題点がある．

また，署名を制限するメッセージの数のオーダーで，秘密鍵と公開鍵の数が多くなってしまうため，署名を制限するメッセージの数が多い場合は好ましくない．

4 提案するメッセージ制限付き署名

本稿では，まず RMS の新しいモデルを提案する．提案するモデルでは，CA も偽造不可能性に対する敵とみなす．このために，CA と署名者が協力して鍵を生成するモデルとし，KeyGen は署名者と CA によって実行されるプロトコルとして考える．

新しい定義を以下に示す．プライバシーと署名者に対する偽造不可能性の定義は [6] と同様であるが，署名者に対する偽造不可能性において，KeyGen は $\mathcal{C}$ と $\mathcal{A}$ が協力して行う．ただし，$\mathcal{C}$ が CA の役割をし，$\mathcal{A}$ が署名者の役割をする．

偽造不可能性

準備． $\mathcal{A}$ は $\mathsf{pk}^{\mathrm{CA}}$ を $\mathcal{C}$ に送る．$\mathcal{A}$ と $\mathcal{C}$ は KeyGen を実行し，$\mathcal{C}$ は $(\mathsf{pk}, \mathsf{sk}, \theta)$ を得る．ただし，$\mathcal{C}$ は署名者の役割をし，$\mathcal{A}$ は CA の役割をする．

署名オラクル． $\mathcal{A}$ はメッセージ M_i を $\mathcal{C}$ に送る．$\mathcal{C}$ は M_i に対する署名 $\sigma_i = \mathsf{Sign}(\mathsf{sk}, M_i)$ を生成し，$\mathcal{A}$ に返す．

偽造． $\mathcal{A}$ は偽造 $(\mathsf{pk}', \theta', M', \sigma')$ を出力する．

$\mathcal{A}$ が有効な偽造を出力すれば，$\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下の式で定義する．

$$\mathrm{Adv}^{gen}_{\mathrm{RMS}}(\mathcal{A}) = \Pr[\mathsf{Verify}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}', \theta', M', \sigma') = 1] \quad \text{ただし } M' \neq M_i$$

定義 4.1. 任意の多項式時間の敵 $\mathcal{A}$ に対して，$\mathrm{Adv}^{gen}_{\mathrm{RMS}}(\mathcal{A})$ が無視できるほど小さいならば，RMS は偽造不可能である．

4.1 提案方式のアイデア

ディジタル署名と PEKS を用いると上述の RMS を実現することができる．提案方式では通常のディジタル署名の署名と PEKS の落とし戸のペアをメッセージ制限付き署名の署名とする．この時，PEKS には署名者が落とし戸を正しく生成したことを効率良く検証できる方法が存在することが条件となる．そのような関数を以下のように定義する．

- $\mathsf{Verify}^{\mathsf{TD}}$**:** メッセージ M と PEKS の公開鍵 $\mathsf{pk}^{\mathrm{PEKS}}$ と PEKS の落とし戸 T_M を入力として，T_M が正しい落とし戸ならば 1 を，そうでなければ 0 を出力する．

4.2 提案方式

図 1 に，提案するディジタル署名と PEKS を用いた RMS を示す．ここで，

- $\mathcal{DS}^{\mathrm{S}} = (\mathsf{KeyGen}^{\mathrm{S}}, \mathsf{Sign}^{\mathrm{S}}, \mathsf{Verify}^{\mathrm{S}})$: 署名者 S の任意の署名方式
- $(\mathsf{KeyGen}^{\mathrm{PEKS}}, \mathsf{Trapdoor}, \mathsf{PEKS}, \mathsf{Test})$: $\mathsf{Verify}^{\mathsf{TD}}$ を持つ PEKS
- $\{M_1, M_2, \ldots, M_n\}$: 署名できないメッセージの集合

とする．

—KeyGen—
S: $(\mathsf{pk}^{\mathrm{S}}, \mathsf{sk}^{\mathrm{S}}) \leftarrow \mathsf{KeyGen}^{\mathrm{S}}(1^{k'})$
S: send pk^{S} to CA
CA: $(\mathsf{pk}^{\mathrm{PEKS}}, \mathsf{sk}^{\mathrm{PEKS}}) \leftarrow \mathsf{KeyGen}^{\mathrm{PEKS}}(1^k)$ $C_i \leftarrow \mathsf{PEKS}(\mathsf{pk}^{\mathrm{PEKS}}, M_i)$ for $1 \le i \le n$ $\mathsf{pk}^{\mathrm{PEKS,C}} = (\mathsf{pk}^{\mathrm{PEKS}}, \{C_i\}_{1 \le i \le n})$ $\mathsf{pk} = (\mathsf{pk}^{\mathrm{S}}, \mathsf{pk}^{\mathrm{PEKS,C}})$ $\theta \leftarrow \mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$
CA: send $(\mathsf{sk}^{\mathrm{PEKS}}, \mathsf{pk}, \theta)$ to S
S: $\mathsf{sk} = (\mathsf{sk}^{\mathrm{S}}, \mathsf{sk}^{\mathrm{PEKS}})$
—$\mathsf{Sign}(\mathsf{sk}, M)$—
$\sigma_0 \leftarrow \mathsf{Trapdoor}(\mathsf{sk}^{\mathrm{PEKS}}, M)$
$\sigma_1 \leftarrow \mathsf{Sign}^{\mathrm{S}}(\mathsf{sk}^{\mathrm{S}}, M)$
output $\sigma = (\sigma_0, \sigma_1)$
—$\mathsf{Verify}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta, M, \sigma)$—
If $\mathsf{Verify}^{\mathrm{CA}}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta) = 1$ and $\mathsf{Verify}^{\mathrm{S}}(\mathsf{pk}^{\mathrm{S}}, M, \sigma_1) = 1$ and $\mathsf{Verify}^{\mathsf{TD}}(M, \mathsf{pk}^{\mathrm{PEKS}}, \sigma_0) = 1$ and $\mathsf{Test}(\mathsf{pk}^{\mathrm{PEKS}}, \sigma_0, C_i) = 0$ for all i, then output 1 else output 0

図 1: 提案するメッセージ制限付き署名

4.3 提案方式の安全性

4.3.1 プライバシー

定理 4.1. PEKS が選択キーワード攻撃に対して安全ならば，提案するメッセージ制限付き署名方式はプライバシーを満たす．

定理 4.1 を証明するためにまず，チャレンジとして $S_0 = \{M_1, M_2, \ldots, M_n\}, S_1 = \{M'_1, M_2 \ldots, M_n\}$ のような一つの要素だけ異なるものを出力する $\mathcal{A}'$ を考える．

補題 4.2. $\mathcal{A}'$ をメッセージ制限付き署名方式に対して，$\mathrm{Adv}^{privacy}_{\mathrm{RMS}}(\mathcal{A}') = \epsilon$ を持つ敵とする．$\mathcal{A}'$ から，PEKS の安全性を少なくとも確率 ϵ で破る $\mathcal{B}$ を構成することができる．

[証明] $\mathcal{B}$ を以下のように構成する．
準備． $\mathcal{B}$ への入力として $\mathsf{pk}^{\mathrm{PEKS}}$ が与えられる．$\mathcal{B}$ は $\mathsf{KeyGen}^{\mathrm{CA}}, \mathsf{KeyGen}^{\mathrm{S}}$ を実行し，$(\mathsf{pk}^{\mathrm{CA}}, \mathsf{sk}^{\mathrm{CA}}), (\mathsf{pk}^{\mathrm{S}}, \mathsf{sk}^{\mathrm{S}})$ を生成する．$\mathsf{pk}^{\mathrm{CA}}$ を $\mathcal{A}$ に送る．
チャレンジ． $\mathcal{A}'$ から $S_0 = \{M_1, M_2, \ldots, M_n\}$ と，$S_1 = \{M'_1, M_2 \ldots, M_n\}$ が送られてくる．$\mathcal{B}$ は (M_1, M'_1) を自身のチャレンジとして出力し，C_1 を得る．$M_2, \ldots, M_n$ に対しては，$\mathcal{B}$ が $\mathsf{pk}^{\mathrm{PEKS}}$ を用いて $C_2, \ldots, C_n$ を生成する．$\mathsf{pk}^{\mathrm{PEKS,C}} = (\mathsf{pk}^{\mathrm{PEKS}}, \{C_i\}_{1 \le i \le n})$ とする．$\mathcal{B}$ は $\mathsf{sk}^{\mathrm{CA}}$ を用いて $\mathsf{pk} = (\mathsf{pk}^{\mathrm{S}}, \mathsf{pk}^{\mathrm{PEKS,C}})$ に対する署名 θ を生成し (pk, θ) を $\mathcal{A}'$ に返す．
署名オラクル． $\mathcal{A}'$ からメッセージ M_i が送られてくる．$M_i \notin S_0 \cup S_1$ なので，$\mathcal{B}$ の持つ落とし戸を生成するオラクルに M_i を送り，$\sigma_{i,0} = \mathsf{Trapdoor}(\mathsf{sk}^{\mathrm{PEKS}}, M_i)$ を得ることができる．また $\mathsf{Sign}^{\mathrm{S}}$ を実行し $\sigma_{i,1} = \mathsf{Sign}^{\mathrm{S}}(\mathsf{sk}^{\mathrm{S}}, M_i)$ を生成して，$\sigma_i = (\sigma_{i,0}, \sigma_{i,1})$ とし，σ_i を $\mathcal{A}'$ に返す．
出力． $\mathcal{A}'$ は b' を出力する．$\mathcal{B}$ は b' を出力する．
$\mathcal{B}$ は $\mathcal{A}'$ の環境を正しくシミュレートできている．よって $\mathcal{A}'$ が b を正しく推測すれば $\mathcal{B}$ も b を正しく推測できる．(証明終)．

[定理 4.1 の証明]
補題 4.2 から，

$$|\Pr[b' = 1 | S_b = \{M'_1, \ldots, M_n\}] - \Pr[b' = 1 | S_b = \{M_1, \ldots, M_n\}]| \le \epsilon$$

が成り立つ．同様に，任意の $1 \le i \le n$ に対して，

$$|\Pr[b' = 1 | S_b = \{\ldots, M_i, M_{i+1}, \ldots, M_n\}] - \Pr[b' = 1 | S_b = \{\ldots, M'_i, M_{i+1}, \ldots, M_n\}]| \le \epsilon$$

が成り立つ．よって，

$$|\Pr[b' = 1 | S_b = \{M'_1, \ldots, M'_n\}] - \Pr[b' = 1 | S_b = \{M_1, \ldots, M_n\}]| \le n\epsilon$$

となり，定理 4.1 が成り立つ．

4.3.2 署名者に対する偽造不可能性の証明

定理 4.2. CA の署名方式 $\mathcal{DS}^{\mathrm{CA}}$ が偽造不可能ならば，提案するメッセージ制限付き署名方式は署名者に対して偽造不可能である．

[**証明**] $\mathcal{A}$ をメッセージ制限付き署名方式に対して，$\mathrm{Adv}^{uns}_{\mathrm{RMS}}(\mathcal{A}) = \epsilon$ を持つ敵とする．$\mathcal{A}$ から，$\mathcal{DS}^{\mathrm{CA}}$ の偽造不可能性を少なくとも確率 ϵ で破る $\mathcal{B}$ を構成する．

準備． $\mathcal{B}$ への入力として $\mathsf{pk}^{\mathrm{CA}}$ が与えられる．$\mathcal{B}$ は $\mathcal{A}$ へ $\mathsf{pk}^{\mathrm{CA}}$ を送る．

チャレンジ． $\mathcal{A}$ から $(\mathsf{pk}^{\mathrm{S}}, S)$ が送られてくる．
$\mathcal{B}$ は PEKS の $\mathsf{KeyGen}^{\mathrm{PEKS}}(1^k)$ と PEKS を実行して，$(\mathsf{sk}^{\mathrm{PEKS}}, \mathsf{pk}^{\mathrm{PEKS,C}})$ を生成する．$\mathcal{B}$ は $\mathcal{DS}^{\mathrm{CA}}$ の署名オラクルに $\mathsf{pk} = (\mathsf{pk}^{\mathrm{S}}, \mathsf{pk}^{\mathrm{PEKS,C}})$ を送り，θ を得る．
$(\mathsf{sk}^{\mathrm{PEKS}}, \mathsf{pk}, \theta)$ を $\mathcal{A}$ に返す．

偽造． $\mathcal{A}$ は偽造 $(\mathsf{pk}', \theta', M', \sigma')$ を出力する．$\mathcal{B}$ は (pk', θ') を偽造として出力する．

$\mathcal{A}$ が $M' \in S$ かつ有効な偽造を出力したとする．

まず，$\mathsf{pk}' = \mathsf{pk}$ の場合を考える．$\sigma' = (\sigma'_0, \sigma'_1)$ とすると，Verify の定義から σ'_0 は M' に対する正しい落とし戸である．この時，ある $j \in \{1, \ldots, n\}$ に対して，$\mathsf{Test}(\mathsf{pk}^{\mathrm{PEKS}}, \sigma_0, C_j) = 1$ となる．これは，すべての j に対して $\mathsf{Test}(\mathsf{pk}^{\mathrm{PEKS}}, \sigma_0, C_j) = 0$ となる定義に矛盾する．よって，$\mathsf{pk}' \neq \mathsf{pk}$.

$\mathsf{pk}' \neq \mathsf{pk}$ の時，定義から，$\mathsf{Verify}^{\mathrm{CA}}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}', \theta') = 1$ となる．また，$\mathsf{pk}' \neq \mathsf{pk}$ なので，$\mathcal{B}$ は $\mathcal{DS}^{\mathrm{CA}}$ の署名オラクルに pk' を聞いていない．よって (pk', θ') は $\mathcal{B}$ の有効な偽造となり，定理 4.2 が成り立つ．

4.3.3 偽造不可能性の証明

定理 4.3. 署名者の署名方式 $\mathcal{DS}^{\mathrm{S}}$ が偽造不可能ならば，提案するメッセージ制限付き署名方式は偽造不可能である．

[**証明**] $\mathcal{A}$ をメッセージ制限付き署名方式に対して，$\mathrm{Adv}^{gen}_{\mathrm{RMS}}(\mathcal{A}) = \epsilon$ を持つ敵とする．$\mathcal{A}$ から，$\mathcal{DS}^{\mathrm{S}}$ の安全性を少なくとも確率 ϵ で破る $\mathcal{B}$ を構成する．

準備． $\mathcal{B}$ への入力として pk^{S} が与えられる．$\mathcal{B}$ は $\mathcal{A}$ から $\mathsf{pk}^{\mathrm{CA}}$ を得る．$\mathcal{B}$ は $\mathcal{A}$ へ pk^{S} を送り，$(\mathsf{sk}^{\mathrm{PEKS}}, \mathsf{pk}, \theta)$ を得る．

署名オラクル． $\mathcal{A}$ からメッセージ M_i が送られてくる．$\mathcal{B}$ は $\mathsf{sk}^{\mathrm{PEKS}}$ を用いて σ_0 を計算する．また M_i を $\mathcal{DS}^{\mathrm{S}}$ の署名オラクルに送り，σ_1 を得る．$\sigma = (\sigma_0, \sigma_1)$ を $\mathcal{A}$ に返す．

偽造． $\mathcal{A}$ は偽造 $(\mathsf{pk}', \theta', M', \sigma' = (\sigma'_0, \sigma'_1))$ を出力する．$\mathcal{B}$ は (M', σ'_1) を自身の偽造として出力する．

$\mathcal{B}$ が $\mathcal{A}$ から得る $\mathsf{pk}^{\mathrm{CA}}, (\mathsf{sk}^{\mathrm{PEKS}}, \mathsf{pk}^{\mathrm{PEKS,C}}, \theta)$ は $\mathcal{DS}^{\mathrm{S}}$ とは独立したものである．$\mathcal{A}$ が有効な偽造を出力したならば，Verify の定義から，$\mathsf{Verify}^{\mathrm{S}}(\mathsf{pk}^{\mathrm{S}}, M', \sigma'_1) = 1$．また $M' \neq M_i$ なので，$\mathcal{B}$ は，$\mathcal{DS}^{\mathrm{S}}$ の署名オラクルに M' を聞いていない．よって，(M', σ'_1) は $\mathcal{B}$ の有効な偽造となり，定理 4.3 が成り立つ．

5 比較，考察

[6] の方式では，RMS の鍵生成を全て CA に依存しているので CA も RMS の署名を生成することができる．

提案方式では，署名者が RMS の鍵の一部 ($\mathcal{DS}^{\mathrm{S}}$ の鍵) を生成しているため，CA は RMS の署名を生成することができない．署名者が鍵の全部 (PEKS の鍵と暗号文も) を生成することも可能だが，署名者が正しく鍵を生成したかどうかをチェックする必要があるので，効率が悪くなる．

なお，提案方式では，署名者がキーワード検索付き公開鍵暗号の落とし戸を正しく生成したものを署名としたかどうかを効率良く検証できることが条件となる．この条件を満たす PEKS として，[2] や [4] の方式がある．[2] はキーワード M と公開鍵 (g, h) と落とし戸 T_M を用いて Diffie-Hellman tuple になっているか $(\hat{e}(g, T_M) = \hat{e}(h, H(M))$ をチェックすることで T_M が正しい落とし戸かどうかを検証できる．[4] はキーワード M と公開鍵 (g, h) と落とし戸 T_M を用いて，
$g^{H(M)} \cdot h \cdot T_M = 1$ かどうかをチェックすることで検証できる．

また，[6] の方式はランダムオラクルモデルで安全性が証明されている．提案方式は，適用するディジタル署名と PEKS がスタンダードモデルで安全であるならば，スタンダードモデルで安全性を示すことができる．

また提案方式は，署名を制限するメッセージの数が多くなると公開鍵は [6] と同様に多くなるが，秘密鍵の数は増えない．

6 構成例

$\mathbb{G}, \mathbb{G}_1$ を位数が q の巡回群とし，g を $\mathbb{G}$ の生成元とする．また $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ は以下の性質を満たす双線形写像とする．

- **双線形性**: 任意の $u, v \in \mathbb{G},\ a, b \in \mathbb{Z}_q$ に対して，
$$\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$$
- **Non-degenerate**: $\hat{e}(g, g) \neq 1$

ハッシュ関数 H_1, H_2 をそれぞれ，$H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G} \to \log q$ とする．提案方式に PEKS として [2] を，ディジタル署名 $\mathcal{DS}^{\mathrm{S}}$ として [3] を適用した RMS を図 2 に示す．$\mathcal{DS}^{\mathrm{CA}}$ は任意とする．

図 2 の方式は $\mathcal{DS}^{\mathrm{CA}}$ が偽造不可能で，かつ Gap DH 仮定 [3] と BDH 仮定 [2] が成り立つならば，RMS の 3 つの安全性を満たす．

また，提案したモデルでは安全性は成り立たないが，図 2 の方式の x と y を等しくすると，[6] のモデルで

は安全で，かつ効率を良くすることができる．$x = y$ とすると，$h_1 = h_2$ なので，KeyGen の計算量が減り，pk も短くなる．署名においては，$\sigma_0 = \sigma_1$ となるので，計算量が減り，署名長が半分になる．また Verify において，署名検証と落とし戸の検証を分ける必要がなく，$\hat{e}(g, \sigma_0) = \hat{e}(h_2, H_1(M))$ の計算一回で済む．

—KeyGen—
S: $x \leftarrow \mathbb{Z}_q{}^*, h_1 \leftarrow g^x$
S: send h_1 to CA
CA: $y \leftarrow \mathbb{Z}_q{}^*,\ h_2 \leftarrow g^y$
$r_i \leftarrow \mathbb{Z}_q{}^*,\ t_i = \hat{e}(H_1(M_i), h^{r_i})$ for $1 \le i \le n$
$f_i = (g^{r_i}, H_2(t_i)) = (A_i, B_i)$ for $1 \le i \le n$
$\mathsf{pk} = (h_1, h_2, \{f_i\}_{1 \le i \le n})$
$\theta \leftarrow \mathsf{Sign}^{\mathrm{CA}}(\mathsf{sk}^{\mathrm{CA}}, \mathsf{pk})$
$\mathsf{pk} = (h_1, h_2, \{f_i\}_{1 \le i \le n}),\ \mathsf{sk} = (x, y)$
S: gets $(\mathsf{pk}, \mathsf{sk}, \theta)$

—Sign(sk, M)—
$\sigma_0 \leftarrow H_1(M)^y$
$\sigma_1 \leftarrow H_1(M)^x$
output $\sigma = (\sigma_0, \sigma_1)$

—Verify($\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta, M, \sigma$)—
If $\mathsf{Verify}^{\mathrm{CA}}(\mathsf{pk}^{\mathrm{CA}}, \mathsf{pk}, \theta) = 1$
and $\hat{e}(g, \sigma_0) = \hat{e}(h_2, H_1(M))$
and $\hat{e}(g, \sigma_1) = \hat{e}(h_1, H_1(M))$
and $H_2(\hat{e}(\sigma_0, A_i)) \neq B_i$ for all i,
output 1
else
output 0

図 2: [2] と [3] の方式を適用したメッセージ制限付き署名

6.1 効率

署名を制限されるメッセージの数を n，$\mathbb{G}$ 上のべき乗演算一回の計算量を E，ペアリング一回の計算量を P とする．[6] の方式 (以下，M)，図 2 の方式 (以下，OO)，及び図 2 の方式で PEKS と $\mathcal{DS}^{\mathrm{S}}$ の秘密鍵を一致させた方式 (以下，OO') の計算量を表 1 に示す．CA の署名方式については共通なのでここでは考えないものとする．

表 1: 計算量

	KeyGen	Sign	Verify
M	$(2n+4)E$	$(n+4)E$	$nE+2P$
OO'	$(2n+1)E+nP$	E	$(n+1)P$
OO	$(2n+2)E+nP$	$2E$	$(n+2)P$

M と OO では，安全性のモデルが異なるため，以下ではモデルが等しい M と OO' の比較を行う．

鍵生成アルゴリズムでは，M は $O(n)$ 回の E を必要とするのに対し，OO' では，$O(n)$ 回の E と P を必要とするので，[6] の方が効率が良い．

署名アルゴリズムでは，M は $O(n)$ 回の E を必要とするのに対し，OO' では，2 回の E の計算をするだけでよい．

検証アルゴリズムでは，M は $O(n)$ 回の E を必要とするのに対し，OO' では，$O(n)$ 回の P を必要とする．共に $O(n)$ で計算量は増えるが，一般にペアリングは計算量が多くかかると言われているので，M の方が効率が良いと言える．

また，M の署名は $\mathbb{G}$ 上の要素 2 つであるが，OO' は $\mathbb{G}$ 上の要素 1 つで済む．OO はモデルは異なるが，M と等しい署名長で済む．

OO に PEKS として [2] の代わりに [4] を用いると，鍵生成の計算量を [6] と同程度にすることができる．署名と検証の計算量，及び署名長は OO と同程度である．

7 まとめ

本稿では，メッセージ制限付き署名について，新しいモデルを提案した．ディジタル署名と PEKS を用いて，提案するメッセージ制限付き署名を構成する方法を示し，その安全性を証明した．また，[6] の方式と提案方式を比較，考察した．

参考文献

[1] D.Boneh, X. Boyen, “Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles,” In Proc. of EUROCRYPT ’04, LNCS3027, pp.223-238, 2004.

[2] D.Boneh, G.Crescenzo, R.Ostrovsky and G.Persiano, “Public Key Encryption with Keyword Search,” In Proc. of EUROCRYPT ’04, LNCS3027, pp.56-73, 2004.

[3] D.Boneh, H.Shacham and B.Lynn, “Short signatures from the Weil pairing,” In Proc. of ASIACRYPT ’01, LNCS2248, pp.514-532, 2001.

[4] C.Gu, Y.Zhu and Y.Zhang, “Efficient Public Key Encryption with Keyword Search Schemes from Pairings,” Cryptology ePrint Archive, Report 2006/108, 2006. http://eprint.jacr.org/2006/108/.

[5] D.Khader, “Public Key Encryption with Keyword Search based on K-Resilient IBE,” Cryptology ePrint Archive,Report 2006/358, 2006. http://eprint.jacr.org/2006/358/.

[6] T.Matsuo, “Restricted Message Signing,” ASIACCS ’06, p.367, 2006.