# 匿名のままの権利行使を可能とした認証方式

佐藤直之† 鈴木英明†

本稿では，ユーザが匿名のまま権利を主張しこれを行使でき，またサービスを提供する側は匿名者であってもその不正行為の取締りが可能な，新たな認証方式を提案する．多くの場合，認証の目的は，利用者が誰であるかを特定することではなく，利用者がその権利を持つかどうかを確認することである．本方式は，このような認証場面に適用することを想定している．本方式では，裁判官と名付けた1つの特殊機関を設置した世界モデルにおいて，ブラインド署名や確率的暗号など種々の暗号技術を応用して作成した電子証明書を利用する．これによって，個人情報の保護を実現しつつ，不正者への容易な対応を可能とした認証方式を実現している．

# An Authentication System that Can Verify an Anonymous Person's Rights

Naoyuki Sato† and Hideaki Suzuki†

Our system enables an anonymous person to prove his rights and a verifier who provides services or resources for his users to trace and catch illegal anonymous users. In many cases, the purpose of authentication is not to identify a user, but to verify the user's rights. Our system is suitable for these situations. In our system, no privacy information is acquired by a verifier, unless the user performs an illegal action. We define a model that includes a 'Judge' as a special authority, and make and use special electronic certificates by using some cryptographic techniques (e.g., a blind signature and probabilistic cryptosystem) in that model.

## 1. まえがき

近年，情報の電子化が進み，また電子ネットワークなどの情報流通の手段が普及して，より簡単かつ手軽に情報を交換することが可能になってきた．これにともなって，個人情報を保護する必要性が強く認識されるようになった．電子ネットワークでは，簡単に情報を配布しまた収集できるので，不必要に個人情報が流通することは，個人の社会生活に甚大な被害を及ぼす可能性も高い．

このような背景の中，現在，電子ネットワーク利用時の匿名性を実現するための手法が数多く提案されている[4)]．匿名性とは，資源やサービスの利用者が，名前や住所，年齢などの自分の特徴を表す個人情報（特徴情報）と，過去に行った行為についての個人情報（行動情報）の2つを隠蔽する性質のことである．

しかし，単に匿名性を提供するだけの個人情報保護手法には大きな欠点がある．匿名者は自分の素性を明らかにしないため，本来自分が持っている権利から導かれる利益を享受できない．匿名者が自分の権利を主張したとしても，その権利は匿名がゆえに容易には認められない．また，一般に匿名者は自分の行動についての証明を行うことができない．このため匿名者は，お得意様向けの特典など，過去の行動に起因したサービスを利用できない．さらに，資源やサービスを提供する側にとっては，匿名者の行った犯罪等違反行為への対応が難しいという問題もある．

本稿では，これらの問題を解決し，利用者が匿名のまま権利を主張しこれを行使でき，またサービス提供者側は匿名者であってもその不正行為の取締りが可能な，新たな認証方式を提案する．多くの場合，認証の目的は，利用者が誰であるかを特定することではなく，利用者がその権利を持つかどうかを確認することである．本方式は，このような認証場面に適用することを想定している．

提案する方式では，ユーザの権利を証明する道具として，各種発行局から事前に発行される特殊な電子証明書を利用する．現実世界における運転免許証やパスポートなどは，これらが所有者の権利を証明する道具

† NTT 情報流通プラットフォーム研究所
NTT Information Sharing Platform Laboratories

であるという点で，本稿での証明書と対応する．また，裁判官と名付けた特殊機関を設け，これによって通常時のユーザの匿名性は保証したまま，不正者への対応を可能とする．

本研究では，これまで述べてきた個人情報の保護の必要性に加えて，ユーザの利便性およびシステムの安全性などの多くの課題を考慮し，以下の要求を設定した．本稿では，これらを満たす認証方式を提案する．

**（1）認証時の特徴情報の流出制御**
認証実施時において，どのような特徴情報が検証者側に伝わるかを，ユーザ自身が制御できる．

**（2）行動の追跡不能性とその制御**
行動が把握されうる範囲をユーザが制御できる．

**（3）証明書の再利用性**
証明書は何度でも利用できる．

**（4）証明書の利用者制限**
証明書はその所有者以外は利用できない．

**（5）証明書の無効化**
発行局は任意の証明書を無効にできる．

**（6）ユーザの特定**
（不正発覚時などに）ユーザを特定する方法がある．

**（7）ユーザの無力化**
任意のユーザに発行されたすべての証明書を同時に無効にする方法がある．

上記の（1）と（2）は個人情報保護のための要求である．本稿で提案する認証方式では，ユーザが自分の名前やIDなどを明かさずに，ある権利を持っていることのみを証明できる．通常，認証時に検証者側が得られる情報は，端末の前にいるユーザが，ある権利を持っているという情報のみである．検証者側は，ユーザが『誰』であるかはもちろん,『誰』であるかに容易に結びつく情報を得ることもできない．ユーザの行動に結びつくすべての情報は，ユーザ自身の管理下にあり，ユーザが望まない限り，名前不詳の匿名者としてでもその行動を監視することはできない．

（3）と（4）は利便性および安全性からの要求である．本稿の方式では，通常の電子チケットとは違って，ユーザは自分に対して発行された証明書を何度でも繰り返し利用できる．しかし同時に，コピーなどによって他人の証明書を不正に入手したとしても，これを利用することはできない．

（5),（6）および（7）は不正行為への対策として有効な要求である．これらは認証システムに当然必要なものであるが，上記の個人情報保護の要求の実現との両立が直観的には簡単でない．本稿で提案する方式では，暗号技術を応用しこれらの両立を実現した．一般に，匿名性が保証されると，不正を試みる人の数は増える傾向にある．このため，本研究では特に不正行為への対策に重点を置いている．

著者らはすでに，文献16)および文献14)などにおいて，耐タンパ性を持つ個人携帯端末の存在を前提として，上記の要求を満たす認証方式の概要を提示している．本稿では，これと同等の機能を持ち，耐タンパ性端末を必要としない汎用的な方式を提案する．これらの違いの詳細については，5章で簡単に説明する．

次章では，本稿で提案する認証方式の基本的な設計方針と，その基盤となる技術を説明する．3章では提案するプロトコルを説明する．4章で安全性などの特徴について検討し，5章で関連研究を，6章でまとめを行う．

## 2. 基本方針と数学的準備

### 2.1 基本方針と基盤技術

本稿では，4種類の存在から成る世界モデルを想定して利用する（**図1**参照）．証明書発行局（**Certificate Issuer** or **CI**）はユーザ（**User**）の権利を保証する証明書を発行する．ユーザは，自分の証明書を適切に利用し資源を利用する．資源管理者（**Resource Manager** or **RM**）は資源やサービスをユーザに提供する存在で，認証における検証者となる．裁判官（**Judge**）は調停役であり，主に不正に対する防衛のために存在する．裁判官は，他の三者が信頼する特別な存在である．発行局，資源管理者，ユーザは複数存在することを想定している．現在一般的な，資源ごとに個人パスワードなどを設定する手法は，図中右側の $CI$ と $RM$ がペアになった場合に相当する．

本稿で提案する方式は，以下の基本的な方針に従って動作する．

(1) ユーザは鍵ペア $(S, P)$ を複数個（増減可）持つ．
(2) 発行局は，証明書を1つの公開鍵 $P$ と関連づけて発行する．証明書を利用するには，$P$ に対応した秘密鍵 $S$ が必要である．

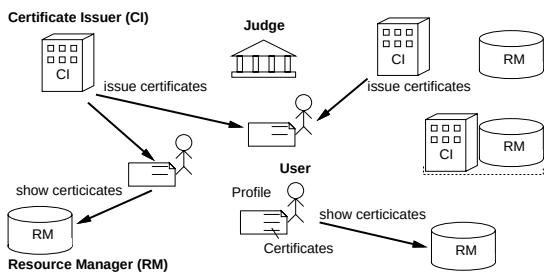


**図1** 世界モデル
Fig. 1 World model

(3) 証明書は，その証明対象であるユーザについての情報を必要以上に含まない．通常証明書の情報からは，そのユーザを一意に識別できない．
(4) ユーザ自身が証明書および鍵を選択し利用する．
(5) 同一ユーザの持つ証明書は，すべて同一の識別情報を含む．ただし，特殊な鍵を持つ裁判官しかこの情報を読み取れない．
(6) 上記の識別情報が既知であれば，誰でもそのユーザに対して発行された証明書を識別できることとする．証明書はこのために必要なデータを含む．

上記の (2) は，証明書を利用できるのが，その所有者だけであることを実現するために設定した．(1), (3) および (4) は個人情報保護のための方針である．(5) および (6) は不正者の特定や証明書の無効化のための方針である．

基盤的技術として，(3) を実現するためにブラインド署名技術[2]を，(5) のためにいくつかの条件を満たした確率的暗号系[9]を，(6) のために特殊な符号系を定義して利用する．他の項目については，証明書の発行や検証のプロトコル自体で実現する．

以下には，数学的準備として確率的暗号技術と符号技術について述べる．

### 2.2 確率的暗号系

本提案方式では，発行局を問わず，同一ユーザに対して発行する証明書には，すべて同じ識別情報を埋め込む．この情報は不正発覚時などに，ユーザを特定するために利用する．このために，意味的安全（強秘匿）な確率的暗号系を利用する．本稿ではこの関数を以下のように記述する．

- 暗号生成関数 $c = E_{crypt}(e, m, r)$
  公開鍵 $e$ を用いて文書 $m$ の暗号文 $c$ を作成する関数．確率的暗号系では，1 つの平文 $m$ の暗号文は多数存在するが，$r$ によって一意に決定する．
- 復号関数 $m = D_{crypt}(d, c)$
  暗号文 $c$ を秘密鍵 $d$ で復号し，文書 $m$ を取り出す関数．

ただし，本稿の方式で利用する暗号系は，以下の 2 つの条件を満たさなければならない．

**条件 1：** 任意の暗号文について，これと同じ平文を持つ別の暗号文を公開情報のみから容易に生成できる．また，同一平文を持つどの暗号文も，等しく生成可能とする．

本稿では，上記の暗号文を変換する関数を $c' = Cv_{crypt}(e, c, t)$ (ただし $c = E_{crypt}(e, m, r)$) と記述する．この関数は，公開鍵 $e$ と暗号文 $c$ より，$c$ と同じ平文を持つ別の暗号文 $c'$ を生成する．$t$ は $c'$ を一意に指定する．秘密鍵を知らなければ，意味的安全性より $(c, c')$ と $(c, E_{crypt}(e, m', r'))$ $(m \neq m')$ の区別は容易でない．

**条件 2：** $c_1 = Cv_{crypt}(e, c_0, t_{01})$，$c_2 = Cv_{crypt}(e, c_1, t_{12})$ のとき，$c_2 = Cv_{crypt}(e, c_0, t_{02})$ となる $t_{02}$ が存在し，$t_{02}$ は $(t_{01}, t_{12})$ より容易に計算できる．このとき逆に，$(c_2, c_0, t_{02})$ からは $(c_1, t_{01}, t_{12})$ を一意に導けない．これはすなわち，暗号文変換の合成は可能であるが，その分解は一意的に決定できないことを意味する．

上記の 2 つの条件を同時に満たす暗号系としては，ElGamal 暗号[6),15)]，Okamoto らの暗号[11)]などがある☆．

### 2.3 特殊な符号系

本提案方式では，あるユーザに対して発行されたすべての証明書を同時に無効にできるようにするために，以下の 5 つの特徴を持つ特殊な符号系を利用する．

**特徴 1：** 1 つの元に対して複数の符号がある．

**特徴 2：** 符号からは，その元についての情報をほとんど得られない．特に，ある符号 $c_0$ に対して，2 つの符号 $(c_1, c_2)$ があり，このどちらかが $c_0$ と同じ元を持っていることが分かっているとしても，$c_1$ と $c_2$ のどちらが同じ元を持つか容易には判別できない．

**特徴 3：** 符号に対して，任意に選ばれたある値が元であるかどうかは容易に判別できる．

**特徴 4：** 符号に対して，その元を知らなくても，同じ元を持つ別の符号を容易に生成できる．また，同一の元を持つどの符号も，等しく生成可能とする．

上記の特徴 1 で定義される符号生成関数を $E_{code}(x, r)$ とおく．$c = E_{code}(x, r)$ は，元 $x$ の符号 $c$ を作成する．$r$ は $c$ を一意に決定するための情報である．特徴 3 で定義される判定関数は $D_{code}(x, c)$ と記す．$c = E_{code}(x', r)$ の場合以下のようになる．

$$D_{code}(x, c) = \begin{cases} 1 & (x = x' \text{ の場合}) \\ 0 & (\text{上記以外の場合}) \end{cases}$$

また，特徴 4 で定義される符号変換関数を

☆ ElGamal 暗号を用いた関数群の構成例を以下に示す．$p$ を大きな素数．$a$ を $mod\ p$ のもとでの原始根．$y = a^x \ mod\ p$ で，公開鍵が $(y, p, a)$，秘密鍵が $(x, p)$ とする．計算はすべて $mod\ p$ のもとで行う．

$E_{crypt}((y, p, a), m, r) = (a^r,\ m \cdot y^r)$
$D_{crypt}((x, p), (c_L, c_R)) = (c_L^x)^{-1} \cdot c_R$
$Cv_{crypt}((y, p, a), (c_L, c_R), t) = (c_L \cdot a^t,\ c_R \cdot y^t)$

$Cv_{code}(c,t)$ とおく．$t$ は変換を一意に指定する．

**特徴 5：** $c_1 = Cv_{code}(c_0, t_{01})$，$c_2 = Cv_{code}(c_1, t_{12})$ のとき，$c_2 = Cv_{code}(c_0, t_{02})$ となる $t_{02}$ が存在し，$t_{02}$ は $(t_{01}, t_{12})$ より容易に計算できる．このとき逆に，$(c_2, c_0, t_{02})$ からは $(c_1, t_{01}, t_{12})$ を一意に導けない．

$D_{code}(x, E_{code}(x,r)) = 1$ である．上記の特徴より，元 $x$ を知らなければ，2 つの符号 $c$ と $c'$ が同一の元を持つかどうかすら分からない．この関数の具体例としては，Diffie-Hellman 決定問題の困難性[5]に根拠をおいた方法などがある☆．

## 3. 提案方式

### 3.1 各構成要素の役割と初期状態

以降において，"$_{CI}$" は証明書発行局に関係するデータに，"$_J$" は裁判官に関係するデータにインデックスとして付与する．

**証明書発行局 CI：** 証明書に対して RSA 署名[12]を基盤としたブラインド署名[2]を行う．各 $CI$ は RSA 署名で利用する鍵ペア $((e_{CI}, n_{CI}), d_{CI})$ を持ち，公開鍵 $(e_{CI}, n_{CI})$ を公開している．また，安全パラメータ $(N, R)$ を決定し公開している．$N$ と $R$ は小さな整数で $0 < R < N$ である．

**資源管理者 RM：** $RM$ は認証時の検証者となる．

**ユーザ：** ユーザは自分の証明書を管理して利用する．ユーザは初期化時に 1 枚の証明書を受け取る☆☆．この証明書にはユーザごとに異なった一意的な *ID*が埋め込まれている．ただし，*ID*は裁判官の公開鍵で暗号化されており，裁判官以外は読み取れない（後述参照）．*ID*は，これ自身からユーザの特定が可能であるように設定する．たとえば，ユーザの名前や生年月日などの特徴情報と，適当な長さの乱数を組み合わせて *ID*を作成する．

また，ユーザは署名を行うために鍵ペア $(P, S)$ を複数セット持つ．この署名の具体的手法については，任意の手法を採用してよい．鍵ペアの数はユーザの意志で自由に増減することができる．ユーザは公開鍵部分を前もって広く公開する必要はなく，また秘密鍵は誰にも教えない．

**裁判官：** 裁判官は 2.2 節で定義した暗号系で使用するための鍵ペア $(e_J, d_J)$ を持ち，公開鍵 $e_J$ を公開している．また,『無効 *ID* リスト』と『無効証明書リスト』（ともに初めは空）を持ち，公開している．

### 3.2 証明書の定義

本方式では，証明書は 6 つの要素から成る．

i. 署名 $s$：証明書発行局の発行した署名．$s$ は紙の証明書における『印』に対応する．

ii. 公開鍵 $P$：$P$ は証明書の利用者を制限する．この鍵と対応した秘密鍵を持った者のみが，この証明書を利用できる．$P$ は紙の証明書での『顔写真』などと対応する．ユーザは 1 つの鍵を複数の証明書の利用条件としてもよい．

iii. 証明事項 $M$：証明書が保証する内容．紙の証明書での文書部分と同じ．

iv. 暗号文 $E$：識別子 *ID* の裁判官の鍵 $e_J$ による暗号文．$E = E_{crypt}(e_J, ID, r)$．

v. 符号 $D$：識別子 *ID* の符号．$D = E_{code}(ID, r')$．

vi. 補足情報 $A$：$A$は 3 種類の情報 $((U_2, \ldots, U_R), (V_2, \ldots, V_R), (c_1, \ldots, c_R))$ より成る．ここで，$R$ はこの証明書を発行した $CI$ が決定し公開している．詳細は下記プロトコルを参照のこと．

### 3.3 発行プロトコル

ユーザが新しい証明書 $C = (s, P, M, E, D, A)$ を獲得する際のプロトコルを示す．プロトコル開始前に，ユーザと発行局 $CI$ の間で，証明事項 $M$ に合意しているとする．プロトコル中でユーザが利用する証明書 $C_o = (s_o, P_o, M_o, E_o, D_o, A_o)$ は，ユーザが以前に獲得した証明書の 1 枚である．これは，初期化時に与えられたものでもよい．関数 $g_S$ と $g_K$ は適当な一方向性ハッシュ関数とし，また "||" は結合を表す．

（1）ユーザは乱数 $(r_k, u_k, v_k, w_k, x_k)$ $(1 \le k \le N)$ を適当に選び，以下の値 $(\alpha_k, \beta_k)$ を計算する．

$$\alpha_k = r_k^{e_{CI}} \cdot g_S(Z_k) \ (mod\ n_{CI}) \quad (1)$$

$$\beta_k = E_{crypt}(e_J, r_k || Z_k, x_k) \quad (2)$$

ただし，

$$X_k = Cv_{crypt}(e_J, X_{k-1}, u_k), \quad X_0 = E_o \quad (3)$$

$$Y_k = Cv_{code}(Y_{k-1}, v_k), \quad Y_0 = D_o \quad (4)$$

$$Z_k = g_K(P||w_k)||X_k||Y_k||M. \quad (5)$$

ユーザは，公開鍵 $P_o$ に対応する秘密鍵 $S_o$ を用いて，式 (6) の文に対する署名 $s_U$ を作成し，$s_U$ と $C_o$ およびすべての（$\alpha_k, \beta_k$）を $CI$ に送る．

$$\alpha_1||\alpha_2||\ldots||\alpha_N||\beta_1||\beta_2||\ldots||\beta_N \quad (6)$$

（2）$CI$ は，$C_o$ の発行局の公開鍵と裁判官の公開する 2 つのリストを用いて，$C_o$ の有効性を確認する（詳細は下記検証プロトコル（2）の第 1 項および 3.5

☆ $p$ を素数，$1 < x < p-1$，$t$ を $p-1$ と素な乱数とする．以下の計算はすべて $mod\ p$ のもとで行う．
$E_{code}(x,r) = (a, a^x)$($r$ を用いて原始根 $a$ を生成)
$D_{code}(x,(c_L,c_R)) = 1$($c_L^x = c_R$ のとき) $or$ 0(左記以外)
$Cv_{code}((c_L,c_R),t) = (c_L^t, c_R^t)$.
注意：$c_L^t$ は原始根となる．

☆☆ この証明書は裁判官自身によって発行されるのが理想的である．

節を参照)．また，$CI$ は $C_o$ からユーザの公開鍵 $P_o$ を取り出し，これを用いて署名 $s_U$ を確認する．すべてに合格した場合，$CI$ は $N$ 個の候補の中から適当に $(N-R)$ 個の候補を選び，このインデックス番号のサブセット $Ind$ をユーザに送る．

$$Ind=\{k_i\},\ 1\leq k_i \leq N \text{ for } 1\leq i\leq N-R \quad (7)$$

(3) ユーザは，$Ind$ に含まれるすべての $k_i$ に対して，$(U_{0k_i}, V_{0k_i}, r_{k_i}, x_{k_i}, g_K(P||w_{k_i}))$ を $CI$ に送る．ここで，$U_{ij}$ は $X_j = Cv_{crypt}(e_J, X_i, t)$ となる値 $t$ であり，$V_{ij}$ は $Y_j = Cv_{code}(Y_i, t)$ となる値 $t$ である．2.2 節および 2.3 節の定義より，$U_{ij}$ は $(u_{i+1}, u_{i+2}, \ldots, u_j)$ から，$V_{ij}$ は $(v_{i+1}, v_{i+2}, \ldots, v_j)$ から容易に計算できる．

(4) $CI$ は，手順 (3) で受け取った情報を用いて，以下の $(\alpha'_k, \beta'_k)$ $(k \in Ind)$ を計算し，これらが手順 (1) で受け取っていた $(\alpha_k, \beta_k)$ と一致することを確かめる．

$$\alpha'_k = r_k^{e_{CI}} \cdot g_S(Z'_k) \ \ (mod\ n_{CI}) \quad (8)$$

$$\beta'_k = E_{crypt}(e_J, r_k||Z'_k, x_k) \quad (9)$$

ただし，

$$X'_k = Cv_{crypt}(e_J, E_o, U_{0k}) \quad (10)$$

$$Y'_k = Cv_{code}(D_o, V_{0k}) \quad (11)$$

$$Z'_k = g_K(P||w_k)||X'_k||Y'_k||M. \quad (12)$$

これに合格した場合，$CI$ は以下の値 $s'$ を計算し，これをユーザに返す．また，$CI$ は $(s', E_o, \{\beta_k\})$ $(k \notin Ind)$ を保存する．

$$s' = \prod_{k\notin Ind} (\alpha_k)^{d_{CI}} \ \ (mod\ n_{CI})$$
$$= \prod_{k\notin Ind} r_k \cdot (g_S(Z_k))^{d_{CI}} \ \ (mod\ n_{CI}) \quad (13)$$

(5) ユーザは，新しい証明書 $C=(s, P, M, E, D, A)$ を以下のように計算する．ここで，$k_i$ は級数 $(1, 2, 3, \ldots, N)$ から $Ind$ に含まれる項を除いた $R$ 個の要素を持つ数列の $i$ 番目の値を示す．

- $s = s'/(\prod_{k\notin Ind} r_k) \ \ (mod\ n_{CI})$
  $= \prod_{k_i} (g_S(Z_{k_i}))^{d_{CI}} \ \ (mod\ n_{CI})$
- $E = X_{k_1}$
- $D = Y_{k_1}$
- $(U_2, U_3, \ldots, U_R) = (U_{k_1k_2}, U_{k_1k_3}, \ldots, U_{k_1k_R})$
- $(V_2, V_3, \ldots, V_R) = (V_{k_1k_2}, V_{k_1k_3}, \ldots, V_{k_1k_R})$
- $(c_1, c_2, \ldots, c_R) = (w_{k_1}, w_{k_2}, \ldots, w_{k_R})$

### 3.4 検証プロトコル

ユーザは以下のようにして証明書を利用する．

(1) ユーザは証明書 $C$ を資源管理者 $RM$ に送る．

(2) $RM$ は $C$ について 3 つの検証を行う．

- 以下の式が成り立つことを確認する．

$$s^{e_{CI}} = \prod_{1\leq i\leq R} g_S(Z''_i) \ \ (mod\ n_{CI}) \quad (14)$$

ただし，

$$X''_1 = E \quad (15)$$

$$X''_i = Cv_{crypt}(e_J, E, U_i)\ (i \geq 2) \quad (16)$$

$$Y''_1 = D \quad (17)$$

$$Y''_i = Cv_{code}(D, V_i)\ (i \geq 2) \quad (18)$$

$$Z''_i = g_K(P||c_i)||X''_i||Y''_i||M. \quad (19)$$

この検証の合格は，『公開鍵 $P$ と対応した秘密鍵を持った存在について，$CI$ が事項 $M$ を保証している』ことを意味する．

- 既存の公開鍵署名を利用した認証手順を用いて，ユーザが公開鍵 $P$ に対応した秘密鍵 $S$ を保持していることを確認する．
- 裁判官の公開する 2 つのリストを用いて $C$ の有効性を調べる．この詳細は次節で説明する．

### 3.5 取消し処理とユーザの特定

証明書を無効にする方法は 3 通りある．この 3 番目の手法では，ユーザを特定することができる．

#### 3.5.1 証明書の無効化 (1)

発行時点において，証明事項 $M$ 内に有効期限などの条件を明記する．この条件を満たさなくなった場合，証明書は無効と考える．条件は，期限などのように容易に判別がつくものであることが望まれる．

#### 3.5.2 証明書の無効化 (2)

無効としたい証明書が既知であるなら，その証明書に含まれる署名 $s$ を『無効証明書リスト』に登録する．このリストに登録された $s$ が含まれる証明書は無効と解釈する．

ただし，$CI$ はブラインド署名の技術を用いて署名を作成しているので，通常自分が発行したどの証明書をどのユーザが持つか分からない．この場合，$CI$ は無効としたい証明書に対応した $(s', E_o, \{\beta_k\})$ $(k \notin Ind)$ を裁判官に送る．裁判官は，秘密鍵 $d_J$ を用いて，$\{\beta_k\}$ から $\{r_k||Z_k\}$ を取り出す．裁判官は以下の式 (20) を検証し，これが成立するなら署名 $s$ を $s = s'/(\prod_{k\notin Ind} r_k)(mod\ n_{CI})$ として計算し，『無効証明書リスト』に登録する．

$$(s')^{e_{CI}} = \prod_{k\notin Ind} r_k^{e_{CI}} \cdot g_S(Z_k) \ \ (mod\ n_{CI}) \quad (20)$$

式 (20) が成立しない場合，保存されていた $\beta_k$ に偽りの情報が含まれていたことになる．この場合，$E_o$ を用いて，次項で説明するようにユーザの無力化などの処置を行うことができる．

#### 3.5.3 ユーザの特定と無力化

裁判官は，証明書自体から，そのユーザを識別し，

このユーザの持つすべての証明書を同時に無効にすることができる．裁判官は，証明書に含まれる情報 $E$ $(= E_{crypt}(e_J, ID, r))$ より，秘密鍵 $d_J$ を用いて，ユーザの識別子 $ID$を導出する．3.1 節の前提により，$ID$からユーザを特定できる．また，無力化のためには，裁判官は $ID$を『無効 $ID$リスト』に登録する．

ある証明書において，証明書に含まれる $D$ $(= E_{code}(ID, r'))$ が，無効 $ID$リストに登録された $ID$の 1 つと対応している場合，すなわち以下の式が成り立つとき，この証明書は無効である．

$$\exists ID_k \text{ in 無効 } ID \text{ リスト} : D_{code}(ID_k, D) = 1 \tag{21}$$

## 4. 検討・評価

本提案方式の特徴を（1）証明書の偽造,（2）証明書の不正利用,（3）証明書の運ぶ情報,（4）負荷の 4 つの点から検討する．

### 4.1 証明書の偽造

証明書の偽造の方法は大きく分けて 3 種類考えられる．1 つは，公開されている情報を用いて，不正者が自分で偽の証明書を作成する方法である．2 つめは，何らかの手段を用いて，3.3 節で説明した発行プロトコル実施時に，発行局をだまして偽の情報を含んだ証明書に署名をさせる方法である．3 つめは，過去の証明書発行処理で得た情報をもとに偽造を行う方法である．

公開情報のみを用いて偽の証明書を作成する場合，作成された証明書は 3.4 節に示した検証プロトコルを通過できなければならない．これは，式 (14) より次の式を満たす値 $s$ を得ることを要求する．

$$s = \left( \prod_{1 \le i \le R} g_S(Z_i'') \right)^{d_{CI}} \quad (mod\ n_{CI}) \tag{22}$$

現在のところ，任意の $Z_i''$ と適切な値 $n_{CI}$ に対して，秘密鍵 $d_{CI}$ を知らずに上記の値 $s$ を計算する実用的な手法は発見されていない．したがって，公開情報のみを用いて偽の証明書を作成することは難しい．

2 つめの手法として，発行プロトコル実施時に，発行局 $CI$ をだまして偽の情報に署名させる場合，以下の情報の偽造を目的とすることが考えられる．

- $M$：任意の内容を保証させる証明書を獲得
- $E$ と $(U_2, \ldots)$：不正ユーザの摘発を妨害
- $D$ と $(V_2, \ldots)$：ユーザ単位での無効化を妨害
- $\beta_k$：証明書単位での無効化を妨害

上記にあげていない $P$ および $(c_1, \ldots)$ はもともとユーザが指定する値であるので偽造する価値がない．

発行プロトコルでは，cut-and-choose の手法を用いて，これらの偽造の成立を防いでいる．上記の中で $\beta_k$ 以外の情報については，集合 $Ind$ に含まれないすべての候補について整合性が維持されるように偽造が行われていなければ意味がない．たとえば，$M$ は $k \notin Ind$ のすべての候補で一致していなければ検証プロトコルを通過できない．したがって，偽造が成功する確率 $p$ は $p = 1/{}_NC_R$ である．逆に，この偽造工作が発行プロトコル実施時に発覚する確率は $1 - p$ である．$p = 1/{}_NC_R = (R! \cdot (N - R)!)/N!$ なので，$p$ は $N$ および $R$ に従って急速に減少する．

次に，$\beta_k$ は証明書に含まれる情報ではなく，発行局が保存しておく情報である．この偽造が成功した場合，その対応する証明書を証明書単位で無効にすることができなくなるが，不正ユーザを特定することや当該証明書を含むそのユーザの持つすべての証明書を無効にすることにはまったく支障を与えない．式 (20) の検証は，このような不正が行われているかを調べている．$\beta_k$ の偽造については，$Ind$ に含まれない 1 つの $\beta_k$ に対して成功すれば，証明書の無効化ができなくなるので，$p = R/N$ の確率で成功する．

3 つめの偽造方法として，過去に行った証明書発行処理で得た情報をもとに，証明書を偽造する方法がある．この手法は，式 (22) の右辺が掛け算の形になっているのを利用する手法である．たとえば，同一内容を保証する 3 枚の証明書をユーザが保持しているとし，これらに対応する署名が各々 $s_1 = abc$, $s_2 = abd$, $s_3 = dxy$ の形であったとする．この場合，ユーザは $s_{illegal} = s_1 s_2^{-1} s_3 = cxy$ を計算して，発行局の意図しない新たな署名 $s_{illegal}$ を構築できる．この手法は，上述のように，発行局が同一要素 $(a, b, d)$ に対する署名を複数回発行したために可能となっている．したがって，同一要素に対する 2 度以上の署名を回避する対策を導入すればこの偽造法は実行不可能である．具体的には，発行プロトコルの手順 (4) において，過去に利用された $Z_k'$ を拒否することや，ユーザの匿名性を侵害しない程度で証明事項 $M$ の中にユーザの意志で決定できない要素（1 万枚単位の既発行枚数カウンタなど）を挿入すること，またこの両方を組み合わせる方法などが簡単で有効である．

### 4.2 証明書の不正利用

ユーザが証明書を利用するためには，これに含まれる公開鍵 $P$ と対応した秘密鍵 $S$ が必要である．したがって，証明書の本来の所有者が秘密鍵 $S$ を秘密に保管しておくのならば，証明書はその所有者しか利用できないことは明らかである．

ただし，本来あってはならないことだが，もし証明

書とそれに対応した秘密鍵が公開されていた場合，これらを入手した者は証明書を不正に利用することができてしまう．この場合，上記の事実が発覚した段階で，裁判官がその証明書の本来の所有者を特定することができる．また，無効証明書リストを用いて証明書を無効にしたり，ユーザ自身を無力化したりすることもできる．

### 4.3 証明書の運ぶ情報

証明書 $C = (s, P, M, E, D, A)$ の持つ情報について考える．まず，署名 $s$ 以外のすべての要素は，誰でも自由に作成できるので，正しい署名が含まれない証明書はまったく情報がない．正しい署名を含む証明書は，各要素の結び付きを，署名者 $CI$ が保証することを示す．

ところで，$CI$ は，ユーザについての多くの特徴情報をあらかじめ知っているかもしれない．名前や電話番号，支払い能力などの情報は，証明書を発行するために $CI$ がユーザに要求する情報であることが予想できる．だが $CI$ は，これらの特徴情報を，証明書と結び付けて不当に利用することができない．$CI$ は証明書発行時に目隠しをしたまま（ブラインドで）署名を行うので，同じ内容 $M$ を保証する証明書を複数のユーザに対して発行した場合，自分が作成した署名 $s$（正確には $s'$）がどの証明書と対応するのかが分からない．関数 $g_K$ の一方向性と乱数 $w_k$ は，cut-and-choose の処理によってユーザの公開鍵 $P$ が $CI$ に知れてしまわないためにある．このため $CI$ は $P$ をユーザ判別のための道具として利用できない．また，式 (3) の $X_k$ および式 (4) の $Y_k$，乱数 $w_k$ の値は $N$ 個の要素で各々異なり，2.2 節と 2.3 節の定義により，開示を行わない $k \notin Ind$ の項について $CI$ はこれらの値を知ることができない．したがって，これらをもとに作成される証明書内の情報 $(E, D, (U_2, \ldots), (V_2, \ldots), (c_1, \ldots))$ は $CI$ にまったく情報を与えない．もちろん，$E$ および $D$ はユーザの識別子 $ID$についての情報を含んでいるが，秘密鍵 $d_J$ を知らない者は，これを利用できない．$CI$ は $\beta_k$ も知っているが，$d_J$ を知らないのでこれを直接には利用できない．

以上の考察より，裁判官を除いたすべての存在に対して，証明書がもたらす情報は，『$CI$ が公開鍵 $P$ に対応した秘密鍵を持った存在について，事項 $M$ を保証している』という内容だけであることが分かる．証明書に署名を行った $CI$ 自身に対しても，この性質が成り立つことは重要である．この性質より，任意の数の $CI$ と $RM$ が結託しても，ユーザの特徴情報と行動情報とを結び付けることは困難となる．

ただし，ユーザが同一証明書を複数回利用した場合は，証明書自体をインデックスとして，$RM$ や $CI$ がユーザの行動の一部を監視できる．この問題は，ユーザが，同一内容を保証する証明書を複数枚保持し，適宜に使い分けることによって容易に解決できる．また，同一鍵ペア $(P, S)$ を複数の証明書に対する条件として利用した場合，$P$ をインデックスとしてユーザの行動を監視できるようになる．このため，ユーザは鍵の利用範囲を適宜に調節しなければならない．また，このようにすることによって，ユーザは自分についての情報の公開度合を制御することができる．

以上のすべての考察より，本提案方式は，1 章での要求を満たしていることは明らかである．

### 4.4 計算負荷の削減：サード パーティの導入

提案方式では，発行プロトコル実施時，検証プロトコル実施時，証明書の無効化実施時にそれぞれ計算が行われる．ただし，本稿では，プロトコル内部で利用する関数 $E_{crypt}$ や $E_{code}$ を具体的に指名せずプロトコルに汎用性を持たせているので，計算負荷について定量的に評価することはできない．ここでは，定性的な負荷について若干の考察を行う．

上記の 3 つの負荷の中でシステム全体に対して最も影響が大きいのは，証明書の検証プロトコルにおける負荷である．発行プロトコルおよび証明書の無効化は，1 つの証明書のライフサイクルにおいて 1 度しか行われない処理である．これに対して，検証プロトコルはユーザが証明書を利用するたびに実行される．このため，検証プロトコルの負荷が小さくなれば，システム全体の負荷が大きく改善する可能性が高い．

検証プロトコル実施時に発生する負荷として大きなものは 2 種類ある．

- 当該証明書が式 (21) を満たさないことを確認する際の計算負荷．これは無効 $ID$リストの登録数に比例した大きさとなる．
- $E$ および $D$ から変換式を用いて $Z_i''$ を計算する際の負荷．これは $R$ に比例した大きさとなる．この計算では多倍精度整数演算を行うことが予想されるので，負荷の大きさが憂慮される．

これらはともに検証側，すなわち $RM$ の負荷である．$RM$ は，当該証明書が無効証明書リストに登録されていないことも確認しなければならないが，適当な検索アルゴリズムを用いれば，この計算負荷は十分に小さい．

上記の 2 つの負荷を削減する手法の 1 つとして，次のような役割を持つサード パーティ（以降 **TP** と呼ぶ）を導入する方法がある．

**TP の役割**

- ユーザから提示された証明書 $C$ に対して，この有効性を確かめてから，$C$ が有効であることを保証する署名 $s_{TP}$ をユーザに発行する．また，$C$ および $s_{TP}$ を保存する．この処理は，各証明書に対して 1 度だけ行われる．
- 裁判官が無効証明書リスト，無効 $ID$ リストを更新した場合，保存してあるすべての証明書に対して，新たに無効となったものがないかを調べる．もし無効となったものが発見されれば，これと対応する $s_{TP}$ を無効として発表する．この $s_{TP}$ の無効化の手段には，既存の CRL（Certificate Revocation List）[10] と同様の機構を採用できる．

$TP$ を導入した場合，検証プロトコルは以下のように変更される．

**検証プロトコル（2）**

（1）ユーザは証明書 $C$ と $s_{TP}$ を $RM$ に送る．

（2）$RM$ は $C$ および $s_{TP}$ について 2 つの検証を行う．

- $TP$ の公開鍵（と CRL）を用いて $s_{TP}$ が有効であることを確認する．
- 既存の公開鍵署名を利用した認証手順を用いて，ユーザが公開鍵 $P$ に対応した秘密鍵 $S$ を保持していることを確認する．

$s_{TP}$ は,『証明書 $C$ が有効であることを $TP$ が保証している』ことを示しているので，厳密には上記の検証プロトコルが検証している事項と，3.4 節のオリジナルのものが検証している事項とは異なる．しかし，検証を実施する $RM$ にとって，$TP$ がその仕事に対して十分に信頼のおける場合には，これら 2 つの意味の違いは問題にならない．

$TP$ は，検証プロトコル実行時に $RM$ が行うことになっていた 3 つの検証処理のうち，ユーザが当該証明書に対応した秘密鍵を保持しているかどうかを確認する処理を除いた 2 つの処理を行う役割を担う．$TP$ はこれらの処理を，検証プロトコルが実行されるそのときにではなく，前もって実行しておくことができる．$TP$ は，3.4 節の検証プロトコルの手順 (2) の第 1 項の内容については，ユーザが証明書を $TP$ に提示したときに 1 度だけ実行する．第 3 項の内容については，ユーザが証明書を $TP$ に提示したときに実行し，また後に裁判官が各種無効リストを更新した時点でその更新分について実行する．$TP$ の被る負荷は，ユーザが証明書を利用する回数とは無関係である．

$TP$ を導入すると，$RM$ の負荷が劇的に軽減することは明らかだ．また，発行プロトコルの手順 (2) では，$CI$ が証明書 $C_o$ の有効性を確認する作業があるが，この部分にも $s_{TP}$ を利用することができ，この場合 $CI$ の負荷も軽減できる．ユーザは，$TP$ により発行された署名 $s_{TP}$ を，任意の $RM$ および $CI$ に対して，繰り返し利用できる．$TP$ を導入したシステムでは，証明書自体の有効性を確認するための上記 2 つの検証処理を，証明書が利用されるときに実行される検証プロトコルから切り離し，独立させている．これによって，上記の処理が実行される回数を減らすと同時に，処理結果を複数の $RM$ および $CI$ で共有できるようにしている．このシステムアーキテクチャの違いによって，$TP$ を導入することでシステム全体としての計算負荷を改善することができる．

ところで，ユーザにとっては，$TP$ が信頼できる存在である必要はない．4.3 節で述べたように，ユーザが $TP$ に提示する証明書自体からは，ユーザの名前などの個人情報が流出することはないからである．$TP$ は，その署名を利用する $RM$ や $CI$ にとってのみ信頼できる存在であればよい．この点で，$TP$ と裁判官との立場は大きく異なる．

## 5. 関連研究

電子ネットワーク上での認証処理における匿名性の実現法は，Chaum によって初めて原始的な手法が紹介された．文献 2) において，Chaum はブラインド署名を提案し，電子現金の方向性を示している．その後，匿名性は，電子現金の分野で活発に議論され，採り入れられている [1),3),18)]．これらの分野では，電子現金（電子チケット，電子証明書）が重複して利用された場合にのみその利用者が判別可能となるような手法を提案し，これによって匿名性と不正対策を両立させているものが多い．この手法はとても有益だが，本方式のように証明書に再利用性を認める場合には利用が難しい．

意味的安全な暗号の定義と実現方法は，Goldwasser らによってに示された [9]．これらについても積極的に研究されている．本研究では特に文献 11)，15) などを参考とした．2.3 節で定義した符号系は本稿独自のものであるが，暗号における意味的安全性と対応する形で設計を行った．

ユーザの公開鍵を認証時に識別子として利用する手法については，SDSI（Simple Distributed Security Infrastructure）[13] や SPKI（Simple Public-Key Infrastructure）[7),8)] などでも検討されている．ただし，これらでは個人情報の保護をその対象とはしていない．また，文献 17) では，本稿の方式と同様に証明書

を用いて匿名性を得る1つの方式が提案されている．ただし，この方法では，証明書の発行局に対する匿名性が考慮されておらず，本方式とは大きく異なる．

1章で述べたように，著者らは，文献14)，16)などにおいて，耐タンパ性を持つ個人携帯端末を前提として，本方式と同様の機能を持つシステムの構築方法の概略を示している．こちらの方式は，各ユーザが持つ端末の機能を利用することで，本稿の方式よりも若干簡単な構造となっている．たとえば，こちらの方式では，本稿2章で述べた暗号系と符号系に対する条件のいくつかが省略できており，これにともなって証明書の発行および検証のプロトコルは大きく異なる．また本稿の方式では，ブラインド署名技術としてRSAを基盤としたものを採用しているが，上記の方式ではこれ以外のブラインド署名技術を採用できる．ただし，本稿の方式は，耐タンパ性端末の存在を必要としないという点で汎用性に優れている．

## 6. むすび

本稿では，特殊な電子証明書を利用して，ユーザが匿名のまま権利を主張しこれを行使できる新たな認証の枠組みを提案した．提案した方式では，裁判官と名付けた1つの特殊な存在を設定し，必要となった場合には，裁判官が匿名ユーザをその実体に結びつける役割を果たす．これによって，通常時のユーザの匿名性を確保しつつ，かつ匿名性を利用したユーザの違反行為に対応している．裁判官を除くすべての存在に対して，ユーザが自分についてのすべての行動情報の流れを制御できる点が大きな特徴である．今後の課題としては，具体的暗号関数の選定やプロトコルのさらなる効率化，より深い安全性の追求と検討，匿名性があることによって実現可能となる新たなサービス分野の開拓と具体的適用方法の検討などがあげられる．

**謝辞** 本研究にあたり，有益なご討論をいただいたNTT情報流通プラットフォーム研究所の齋藤孝文氏（現サービスインテグレーション基盤研究所），桑名栄二氏，曽根岡昭直氏〔現NTTデータ（株）〕に感謝いたします．また，様々な議論の場を通してご協力いただいた同研究グループの皆様に感謝いたします．

## 参考文献

1) Brands, S.: Untraceable Off-line Cash in Wallet with Observers, *Proc. CRYPTO'93*, LNCS, Vol.773, pp.302–318, Springer-Verlag (1993).
2) Chaum, D.: Security without Identification: Transaction Systems to Make Big Brother Obsolete, *Comm. ACM*, Vol.28, No.10, pp.1030–1044 (1986).
3) Chaum, D., Fiat, A. and Naor, M.: Untraceable Electronic Cash, *Proc. CRYPTO'88*, LNCS, Vol.403, pp.319–327, Springer-Verlag (1988).
4) Cranor, L.F., et al.: Internet Privacy, *Comm. ACM*, Vol.42, No.2, pp.29–67 (1999).
5) Diffie, W. and Hellman, M.: New direction in cryptography, *IEEE Trans. Information Theory*, Vol.IT-22, No.6, pp.644–654 (1976).
6) Elgamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Trans. Information Theory*, Vol.IT-31, No.4, pp.469–472 (1985).
7) Ellison, C.M.: Establishing Identity Without Certification Authorities, *6th USENIX Security Symposium* (1996).
8) Ellison, C.M., Frantz, B., et al.: SPKI Certificate Theory, RFC 2693 (1999).
9) Goldwasser, S. and Micali, S.: Probabilistic Encryption, *Journal of Computer and System Sciences*, Vol.28, pp.270–299 (1984).
10) ITU-T Recommendation X.509: *Information Technology – Open Systems Interconnection The Directory: Authentication Framework* (1997).
11) Okamoto, T. and Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring, *Proc. EUROCRYPT'98*, LNCS, Vol.1403, pp.308–318, Springer-Verlag (1998).
12) Rivest, R., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).
13) Rivest, R.L. and Lampson, B.: SDSI – A Simple Distributed Security Infrastructure. http://theory.lcs.mit.edu/~cis/sdsi.html (1996).
14) Sato, N. and Suzuki, H.: An authentication system based on certificates that protects privacy, *Proc. ICOIN-14*, pp.1B-2.1–1B-2.8 (2000).
15) Tsiounis, Y. and Yung, M.: On the Security of ElGamal Based Encryption, *PKC'98*, LNCS, Vol.1431, pp.117–134, Springer-Verlag (1998).
16) 佐藤直之，鈴木英明：プライバシ保護に注目した証明書を基盤とした認証システムの一方式，情報処理学会研究報告，Vol.99, No.45, pp.31–36 (1999).
17) 満保雅浩，岡本栄司：限定匿名証明書の使用者を指定する方法について，信学技報，ISEC96-34, pp.9–20 (1996).
18) 岡本龍明，太田和夫：理想的電子現金方式の一

方法，電子通信学会論文誌（D-I），Vol.J76-D-I, No.6, pp.315–323 (1993).

(平成 11 年 12 月 8 日受付)
(平成 12 年 6 月 1 日採録)

**佐藤　直之**

1997 年慶応大学大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話（株）入社．入社直後は情報検索技術の研究に努め，その後，暗号理論と暗号理論を応用した新サービスについての研究および開発に従事．現在，NTT 情報流通プラットフォーム研究所所属．

**鈴木　英明**（正会員）

1985 年早稲田大学理工学部数学科卒業．同年，NTT 武蔵野電気通信研究所入所．以来，知識獲得，定理証明系，リエンジニアリング，プログラムの抽象化検索，情報流通基盤技術の研究に従事．現在，NTT 情報流通プラットフォーム研究所主任研究員．ACM，IEEE-CS，電子情報通信学会各会員．