素因数分解に基づく効率的な署名方式の提案

岡本 健[†] 多田 充[†] 宮地充子[†]

1999年，Poupard と Stern は on the fly 署名と呼ばれる署名方式（PS 方式）を提案した．この方式は，on line の署名作成時において，剰余演算を不要にすることにより，高速な署名実現する．本論文では，PS 方式を改良することにより，新しい on the fly 署名を提案する．PS 方式は，秘密鍵のサイズが法のサイズと素因数の数に依存しており，素因数の数の増大に従って秘密鍵のサイズが大きくなるという短所があった．提案方式は，秘密鍵の構造を変更することによって，この問題点を解決している．これらの改善により，提案方式は計算処理量（事前計算，署名生成，検証）とデータサイズ（秘密鍵，署名）の 2 点に関して効率が良くなっている．PS 方式と提案方式の性能を比較した場合，事前計算，署名生成，検証の計算量は，それぞれ 55%，33%，47%以上削減可能である．また，秘密鍵，署名サイズは，それぞれ 33%，23%以上削減可能である．提案方式のこのような特徴は，現在においても計算処理や記憶量に制限のある IC カードの利用に適している．

Proposal of Efficient Signature Schemes Based on Factoring

Takeshi Okamoto,[†] Mitsuru Tada[†] and Atsuko Miyaji[†]

In 1999, Poupard and Stern proposed *on the fly* signature schemes (PS schemes), which aim at minimizing the on-line computational work for the signer. In this paper, we propose efficient on the fly signature schemes which are derived from three-pass identification scheme. To construct our schemes, we improve PS schemes in the computational work (pre-computation, on-line signature generation and verification) and the data size (secret key and signature) by changing the structure of the secret key and by extending two primes of RSA modulus to three or more primes. Compared with the previous scheme, the complexity of pre-computation, on-line signature generation, and verification are reduced by at least 55%, 33% and 47% respectively, and the size of secret key and signature is also reduced by at least 33% and 23% respectively. Consequently, our proposed schemes are suitable for a smart card application, whose CPU power or memory is rather limited.

1. はじめに

近年，情報化社会の進展やデバイス技術の発達により，IC カードや携帯端末を利用した情報通信が急増している．このような通信の安全性を確保するため，通信基盤の基幹技術である暗号技術，特に公開鍵基盤となる安全な署名技術が求められている．

これらの要求に対応し，かつ利用者の利便性を保つためには，署名の高速化，およびコンパクト化が要求される．例として，IC カードを用いた電子商取引を考える．利用者は安全性を確保するため，このカードに署名機能を付加することが望ましい．通常，利用者はカードリーダに IC カードを通すという操作によって，電子決済を行うが，利用者の利便性を向上させるために，署名の高速化が求められる．また，銀行のカードやプリペイドカードという一般的な IC カードの場合，現在における性能は，CPU が 16 ビット程度，メモリサイズが 10 K バイト程度なので，通常の計算機と比べて，計算処理能力や記憶容量に制約がある．このため，署名のコンパクト化というのも大切な要素である．

署名作成に必要な計算は，事前計算と，on line で必要となる署名生成の 2 つに分類できる．署名の性能評価において，我々はこの 2 つを明確に区別しなければならない．事前計算は，署名者が使用する機器の稼働時にバックグラウンドで計算が可能なため，リアルタイムの処理時間に影響を与えない．一方，署名生成はこのような処理時間を要求するため，署名生成の効率化は重要な課題である．

1992 年，Girault [7] は，Schnorr 署名[22] において公開鍵となる素数法の代わりに RSA 法（素因数が 2 つからなる合成数）を用いる署名方式（GPS 方式）を提案した．Schnorr 署名では署名生成において加算，乗算，剰余という 3 種類の演算が必要であった．これ

† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology

に対し，GPS 方式は，この中で剰余を不要にしているため，Schnorr 署名より高速な署名生成を実現できる．1998 年，Poupard と Stern [17] は GPS 方式の厳密な安全性について考察した．本論文では，GPS 方式のように署名生成において剰余を行わない署名方式を，on the fly 署名と呼ぶことにする．

さらに 1999 年，Poupard と Stern は安全性が素因素分解に帰着する on the fly 署名[18]（PS 方式）を提案した．この方式は，公開鍵のパラメータ数が GPS と比べて少ないため，公開鍵のサイズを削減できるという利点を持つ．しかしながら，PS 方式における秘密鍵のサイズは，公開鍵となる合成数法の素因数の数に依存しており，この数値を増やすに従って秘密鍵のサイズが大きくなり，結果として計算処理量とデータサイズの効率が悪くなるという問題点がある．

本論文では，素因数分解に基づく新しい on the fly 署名を提案する．提案方式は，PS 方式を改良することによって得られた方式である．提案方式の主な特徴は，秘密鍵の構造を変更している点である．これにより，提案方式は PS 方式と異なり，合成数法の素因数の数を増やすことによって，秘密鍵のサイズを削減できる．現在，RSA 暗号[21] を高速化するために，公開鍵 n の素因数の数を増やす試み[23] がなされているが，提案方式は PS 方式と異なり，このような手法を容易に適用できる．また，提案方式は認証から変換された方式であり，この認証は正直な検証者に基づくゼロ知識対話証明なので，他の既存方式[6),18),22)] と同様に，安全性の証明[16] が可能である．

さらに，提案方式は PS 方式と同程度の安全性を持ちながら，計算処理量（事前計算，署名生成，検証）とデータサイズ（秘密鍵，署名）の点で優れている．両方の方式を比較した場合，提案方式は，事前計算，署名生成，検証の計算量について，それぞれ 55%，33%，47%以上，秘密鍵，署名のサイズについて，それぞれ 33%，23%以上の効率化を実現している．

本論文の構成は以下のようになる．2 章では，本論文で使用する記号や用語の定義を与える．3 章では，PS 方式について説明し，この方式の問題点を述べる．4 章，5 章では，今回提案する認証方式，および署名方式をそれぞれ述べる．また，提案方式の特徴，および安全性についても検討する．6 章では，提案方式で用いる鍵やパラメータが満たすべき条件等について検討する．7 章では，提案方式のデータサイズを効率化する方法について述べる．8 章では，これまでに提案された主要な署名方式と比較することにより，提案方式の性能を評価する．9 章で結論を述べる．

2. 準　　備

本論文で用いる記号や用語について定義する．

$\mathbb{Z}$：すべての整数の集合
$\mathbb{Z}_p$：0 以上 p 未満の整数の集合
$\mathbb{Z}_p^*$：$\mathbb{Z}_p$ かつ p と互いに素な整数の集合
$\mathbb{N}$：すべての自然数の集合
$\mathbb{N}_{>i}$：i より大きい自然数の集合
$\mathbb{N}_{\mathrm{prime}}$：すべての素数の集合
$a|b$：a は b を割り切る
$|x|$：x を 2 進表現したときの長さ
$\gcd(a,b)$：a と b の最大公約数
$\varphi(x)$：x の Euler 関数
$\lambda(x)$：x の Carmichael 関数
$\mathrm{ord}_n(g)$：乗法群 $\mathbb{Z}_n^*$ における g の位数

定義 2.1　f，g を非負関数とする．このとき，

$(\exists c>0, \exists u, \forall x>u)\ [f(x)/g(x)<c]$,
$(\forall c>0, \exists u, \forall x>u)\ [f(x)/g(x)<c]$,

であるならば，それぞれ $f(x)=O(g(x))$，$f(x)=o(g(x))$ と表記する．■

定義 2.2　f を非負関数とする．このとき，

$(\forall c>0)\ [f(x)=o(1/x^c)]$,

が成り立つのであれば，$f(x)$ は x に関して無視できるという．逆に $f(x)$ が無視できる関数でないとき，$f(x)$ は x に関して無視できないという．■

本論文では，特に記述がない場合，「無視できる」，「無視できない」という議論はセキュリティパラメータに関して行う．このため，特別な場合を除き，「セキュリティパラメータに関して」という記述は省略する．

定義 2.3　素因数分解問題とは，$n\in\mathbb{N}_{>1}$ が与えられたとき，$a|n$ $(1<a<n)$ となるような整数 a を求める問題である．■

3. 従来方式とその特徴

本章では，PS 方式の認証（**図 1**）と署名の両機能について説明する．

3.1 認 証 方 式

鍵生成アルゴリズム

Step 1　サイズが同じ 2 つの素数 p，q $(p=2p'+1,\ q=2q'+1,\ (p',q')\in\mathbb{N}_{\mathrm{prime}})$ を選ぶ．$n=pq$ とする．

Step 2　$L\in\{p'q', 2p'q'\}$ に対して，$\mathrm{ord}_n(g)=L$ となるような $g\in\mathbb{Z}_n^*$ を選ぶ．

Step 3　$s=n-\varphi(n)=p+q-1$ を計算する．

Step 4　証明者の公開鍵を (n,g)，秘密鍵を s とする．

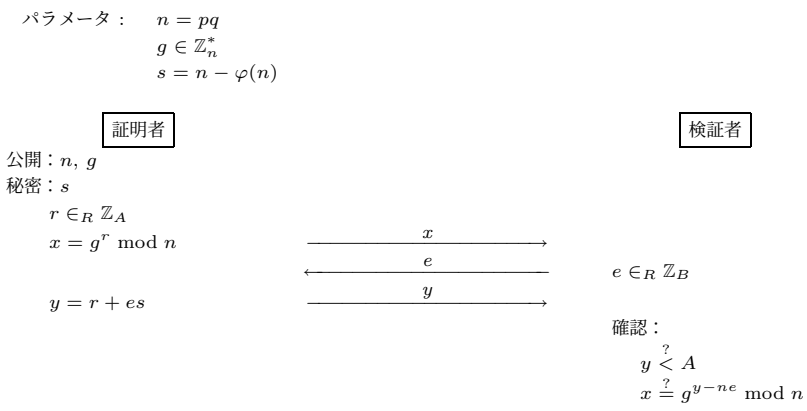

図 1 Poupard-Stern 認証方式
Fig. 1 Poupard-Stern identification scheme.

認証アルゴリズム 証明者は，以下に示す手順を ℓ 回繰り返す．すべてのラウンドで検証に合格すれば受理する．そうでなければ受理しない．

Step 1 証明者は，乱数 $r \in \mathbb{Z}_A$ を選び，$x = g^r \bmod n$ を計算し，x を検証者に送る．ここで，$A < n$, $|A| = \kappa + k + |B|$ である☆．

Step 2 検証者は，乱数 $e \in \mathbb{Z}_B$ を選び，証明者に送る．

Step 3 証明者は，$y = r + se$（$\mathbb{Z}$ 上）を計算し，検証者に送る．

Step 4 検証者は，$y < A$ および $x = g^{y-ne} \bmod n$ の両方が成り立つか確認する．

3.2 署名方式

鍵生成アルゴリズム 認証方式と同じ手順により，署名者の鍵を作成する．

署名生成アルゴリズム

入力：署名者の公開鍵 (n, g)，秘密鍵 s，メッセージ m

出力：メッセージ m に対する署名 (e, y)

Step 1 乱数 $r \in \mathbb{Z}_A$ を選ぶ．

Step 2 $x = g^r \bmod n$ を計算する．

Step 3 $e = \mathcal{H}(x, m)$ を計算する．ここで，$\mathcal{H}$ は $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|B|}$ となるハッシュ関数である☆☆．

Step 4 $y = r + se$（$\mathbb{Z}$ 上）を計算する．

Step 5 (e, y) を出力する．

署名検証アルゴリズム

入力：署名者の公開鍵 (n, g)，メッセージ m，署名 (e, y)

出力：「受理」あるいは「不可」

Step 1 もし，$y < A$ が成り立たなければ「拒否」を出力してプロトコルを停止する．

Step 2 $x' = g^{y-ne} \bmod n$ を計算する．

Step 3 $e' = \mathcal{H}(x', m)$ を計算する．

Step 4 もし，$e = e'$ が成り立つのであれば「受理」を出力，そうでないならば「不可」を出力する．

3.3 特徴と問題点

本節では，文章の簡単化のため，署名に関して記述するが，認証についても同様に考察できる．

検証者は，検証時に署名の一部となる y のサイズを明示的に確認する必要がある．このような確認は，既存方式[4),8),22)]には見られないため，PS 方式の特徴といえる．

秘密鍵 $s = n - \varphi(n)$ は，公開鍵 n のみに依存している．また，s と n は法 $\varphi(n)$ において合同であり，s のサイズは n のサイズに比べ約 1/2 である．また，s は，サイズを大きくすると，計算処理量（事前計算，署名生成，検証）とデータサイズ（署名）の両方に関して効率が悪くなる．

署名の一部である $y = r + se$ の演算は，$\mathbb{Z}$ 上で行うが，se のサイズより大きいサイズの乱数 r を加えることにより，秘密鍵の情報漏洩を防いでいる（秘密情報のマスク化）．このため，y のサイズは se のサイズに依存する．

公開鍵 g の位数 L は公開されないため，検証者は，

☆ k はセキュリティパラメータであり，$k = |n|/2$, κ は情報リークパラメータであり，$1/\kappa$ が無視できるように設定する．また，B と ℓ は $1/B^\ell$ が無視できるように設定する．

☆☆ B は $1/B$ が無視できるように設定する．

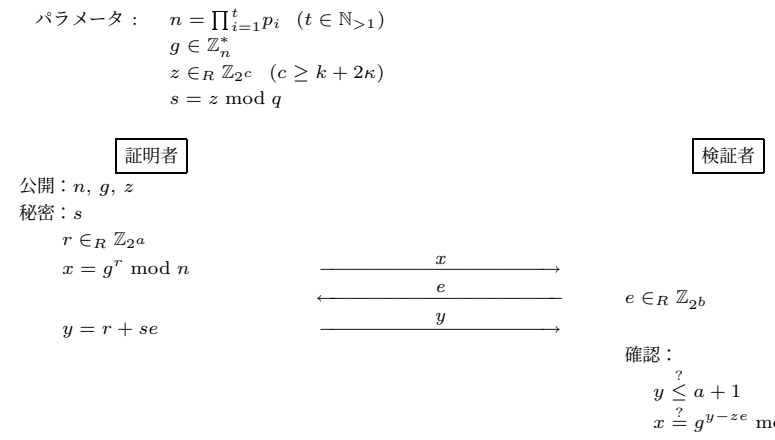

図 2 提案型認証方式
Fig. 2 Proposed identification scheme.

$x = g^{y-ne} \bmod n$ の検証時において，$|y - ne|$ ビットの指数演算を行う必要がある．

上記のような特徴から，PS 方式には以下のような問題点がある．

公開鍵 n の拘束 公開鍵 n の素因数の数を，3 つ以上に変更した場合，秘密鍵 s のサイズが増大する．たとえば，$n = pqr$ $(p, q, r \in \mathbb{N}_{\text{prime}})$ というように 3 つの素数の積からなる n を用いてシステムを構成した場合，秘密鍵は $s = n - \varphi(n) = n - (p-1)(q-1)(r-1) = pq + qr + rp - (p + q + r) + 1$ となり，秘密鍵のサイズが変更前と比べ，約 3/2 になる．このことは前述のとおり，計算処理量とデータサイズの両方が増加するため，変更前と比べて効率が悪くなる．

検証における計算量の増大 検証式 $x = g^{y-ne} \bmod n$ に関して，$y < A$ なので，$|y| < |ne|$ という条件が成り立つ．このため，検証者の計算量は $|ne|$ の増加に従って増大する．たとえば，$|n| = 1024$, $|e| = 80$ とすると約 1104 ビットの指数演算が必要である．これは現在使用されている主要な方式[13),21),22)] と比べて計算量が非常に大きい．

4. 認 証 方 式

本章では，提案する認証方式について説明する（**図 2**）．本章で記述するパラメータは，$2^{k+b} \ll 2^a$, $q < 2^k \ll 2^c$ という条件を満たす．鍵やパラメータに関する詳細については，6 章に記述する．

鍵生成アルゴリズム

Step 1 素因数の数 $t \in \mathbb{N}_{>1}$ を決定する．次に，サイズが同じ t 個の素数 p_i $(p_i = 2q_i + 1, q_i \in \mathbb{N}_{\text{prime}}, 1 \leq i \leq t)$ を選ぶ．$n = \prod_{i=1}^{t} p_i$ とする．

Step 2 $\text{ord}_n(g) = q$ $(q | \lambda(n))$ となるような $g \in \mathbb{Z}_n^*$ を選ぶ．

Step 3 乱数 $z \in \mathbb{Z}_{2^c}$ を生成する．

Step 4 $s = z \bmod q$ を計算する．

Step 5 証明者の公開鍵を (n, g, z)，秘密鍵を s とする．

認証アルゴリズム 証明者は，以下に示す手順を ℓ 回繰り返す．すべてのラウンドで検証に合格すれば受理する．そうでなければ受理しない．

Step 1 証明者は，乱数 $r \in \mathbb{Z}_{2^a}$ を選び，$x = g^r \bmod n$ を計算し，x を検証者に送る．

Step 2 検証者は，乱数 $e \in \mathbb{Z}_{2^b}$ を選び，証明者に送る．

Step 3 証明者は，$y = r + se$（$\mathbb{Z}$ 上）を計算し検証者に送る．

Step 4 検証者は，$|y| \leq a + 1$ および $x = g^{y-ze} \bmod n$ の両方が成り立つか確認する．

4.1 変更点と利点

前述のとおり，PS 方式の公開鍵 n は，効率の観点から，RSA 法を用いなければならなかった．このとき，s のサイズは n のサイズの約 1/2 になる．これに対し，提案方式の秘密鍵は，$s = z \bmod q$ となるので，s のサイズは，q のサイズと同程度になる．提案方式に対する攻撃として，6.1 節に記述しているような q に依存する攻撃を考えた場合，計算量は $O(\sqrt{q})$ となり，指数時間かかるため，実装環境では q のサイズを小さくとることができる．

また，検証式に関して，PS方式は $x = g^{y-ne} \bmod n$ であったが，提案方式は $x = g^{y-ze} \bmod n$ となり，指数部分のパラメータが n から z に変更されている．z のサイズは，n のサイズより小さくできるため，提案方式はPS方式と比べ，検証時の計算量を大幅に削減できる．

上記のような変更により，提案方式は以下のような手法を利用できる．

中国人剰余定理の利用 事前計算 $x = g^r \bmod n$ の計算を速くするため，中国人剰余定理を用いる方法がある．このとき，n の素因数の数をPS方式より増やした場合，計算量はさらに削減できる．たとえば素因数の数を2から3に変更した場合，計算量は素因数の数が2の場合と比べて約4/9になる．

4.2 安 全 性

提案する認証は，知識の対話証明であり，その中でもゼロ知識対話証明と呼ばれる方式である．本論文において，対話証明の定義は文献4)に従う．また，安全性の証明のために素因数分解問題の困難性を仮定する．

次の補題を準備する．

補題 4.1 n は任意の整数，L は $\lambda(n)$ の倍数であり，L のサイズは $|n|$ の多項式で抑えられるとする．このとき，入力 (n, L) に対し，$O(|n||L|)$ の計算量で n の素因数を出力する確率的多項式時間Turing機械を構成できる．

証明 この補題は，Miller[12]により示された．例として，n を異なる素数の積からなる合成数とするとき，以下のようなアルゴリズムを用いる．

素因数分解アルゴリズム 表記として，$f(x)$ は整数 x の多項式で表現されるような関数とする．このとき，$a \le f(|n|)$ となるようなすべての整数 a に対して以下を実行する．

Step 1 $a|n$ が成り立つか確認する．もし，成り立つのであれば a を出力する．

Step 2 $a \le b \le \max\{K : 2^K|L\}$ となるようなある整数 a，b に対して

$$\gcd((a^{L/2^b} \bmod n) - 1, n) = c$$

を計算する．もし，$c \neq 1$ ならば，c を出力する．

このアルゴリズムの計算量は，$O(|n||L|)$ である．任意の合成数を素因数分解するアルゴリズムの詳細は，文献12)に記述されている． □

提案する認証方式は，統計的ゼロ知識証明であることを証明する，完全性，健全性，統計的ゼロ知識性をそれぞれ示す．証明の手法は，いずれも文献17)，18)に従う．

定理 4.2 ［完全性］ 認証アルゴリズムは，完全である．

証明 正直な証明者 P と正直な検証者 V が，認証アルゴリズムを実行したとき，$|se| < |r| \le a$ より，$y\ (= r + se) \le a + 1$ となる．また，

$$g^{y-ze} = g^{r+(z \mod q)e-ze} = g^r = x \bmod n$$

なので，このようなアルゴリズムの実行により，つねに検証式は成り立つ． □

定理 4.3 ［健全性］ 認証アルゴリズムは，健全である．

証明 不正な証明者を P^* とする．ここで，P^* は乱数テープ ω を持つ確率的多項式時間Turing機械である．P^* を用いて，以下のような確率的多項式時間Turing機械 M を構成する．

最初に M は，2つの乱数 $\tilde{g} \in \mathbb{Z}_n^*$，$\tilde{z} \in \{2^{k+\kappa}, 2^{k+\kappa+1}, \cdots, 2^c - 1\}$ を選び，攻撃対象となる公開鍵を新たに $(n, \tilde{g}, \tilde{z})$ と設定する．

偽造アルゴリズム M は以下に示す手順を ℓ 回に達するか，あるいはアルゴリズムの停止が宣言されるまで繰り返す．

Step 1 乱数テープ ω と公開情報を P^* に与え，$x \in \mathbb{Z}_n^*$ を得る．

Step 2 $\mathbb{Z}_{2^b}$ の中から，今回のラウンドにおいて，まだ選択されていない整数をランダムに選ぶ．次に，この整数を e として P^* に与え，検証式が成り立つような y が得られたかどうか確認する．その後，P^* の状態（P^* はプログラム化されている）をクリアにして，Step1の終了時と同じにする．このような試行を 2^b 回繰り返す．

Step 3 もし P^* から検証式の成り立つ2組 $(e_1, y_1), (e_2, y_2)$ を得たのであれば，$(Y, E) = (y_1 - y_2, e_1 - e_2)$ を出力して，アルゴリズムを停止する．

V は，P^* によって，$1/2^{b\ell} + \varepsilon$ $(\varepsilon > 0)$ 以上の確率で受理すると仮定する．このとき，偽造アルゴリズムを1回試行することより，M が (Y, E) を出力する確率は，分割補題（Splitting Lemma）[16]より $\varepsilon/2$ 以上であり，計算量は $O(2^b \ell \tau)$ となる．ここで，τ は ℓ ラウンドにおける平均実行時間である．

次に，M は偽造アルゴリズムを $|n|/\varepsilon$ 回繰り返す．このとき，偽造に成功する確率は，

$$\begin{aligned} 1 - \left(1 - \tfrac{\varepsilon}{2}\right)^{\frac{|n|}{\varepsilon}} &= 1 - \left(1 - \tfrac{\varepsilon}{2}\right)^{-\frac{2}{\varepsilon}\left(-\frac{|n|}{2}\right)} \\ &> 1 - e^{-|n|} \end{aligned}$$

であり，計算量は $O(2^b|n|\ell\tau/\varepsilon)$ となる．また，$x=\tilde{g}^{y_1-\tilde{z}e_1}=\tilde{g}^{y_2-\tilde{z}e_2}$ より，$\tilde{g}^{Y-zE}=1 \bmod n$ なので，$L=Y-\tilde{z}E$ は $\tilde{g}$ の位数の倍数になる．さらに，$\tilde{g}$ の位数が $\lambda(n)$ である確率は，

$$\frac{\varphi(\lambda(n))}{\varphi(n)}=\frac{\varphi(\prod_i q_i)}{2^t\prod_i q_i}=\frac{1}{2}\prod_i\left(1-\frac{1}{q_i}\right)>\frac{1}{2^{2t}}$$

である．最後に，M は Miller のアルゴリズム[12] を用いて n の素因数を求める．このアルゴリズムの計算量は $O(|n||L|)=O(|n|^{O(1)})$ である．

上記のような操作により，M は $O(2^b|n|\ell\tau/\varepsilon+|n|^{O(1)})$ の計算量で，n の素因数を求めることができる．もし，ε が無視できない確率ならば，M は多項式時間で公開鍵 n の素因数を求められるが，このことは素因数分解問題の困難性に矛盾する． □

定理 4.4 ［統計的ゼロ知識性］ $qT/2^a$ が無視できると仮定する．ここで，T は k の関数であり，同一鍵に対する認証アルゴリズムの最大繰返し回数とする．このとき，認証アルゴリズムは，統計的ゼロ知識性を持つ．

証明

任意の検証者を V^* とする．このとき，P と V^* の対話を再構成する確率的多項式時間 Turing 機械（シミュレータ）TM_{V^*} を考え，以下のように構成する．

模倣アルゴリズム TM_{V^*} は (x',e',y') の出力が ℓ 組になるまで，以下を繰り返す．

- **Step 1** 乱数 $e'\in\mathbb{Z}_{2^b}$ と $y'\in\{\Phi,\Phi+1,\cdots,2^a-1\}$ を選ぶ．ここで，$\Phi=(2^b-1)(2^k-1)$ である．
- **Step 2** $x'=g^{y'-ze'} \bmod n$ を計算する．
- **Step 3** V^* に x' を入力し，e を得る．
- **Step 4** もし，$e=e'$ ならば (x',e',y') を出力する．そうでなければ今回のラウンドを無効にする．

整数 A，正の整数 Δ，関数 $F:\mathbb{Z}_n\to\mathbb{Z}_{2^a}$ に対して $\mathcal{N}(F,A,\Delta)$ は，$e=F(g^{y-ze})$ であり，$(e,y)\in\mathbb{Z}_{2^b}\times\{A,A+1,\cdots,A+\Delta-1\}$ となるような集合の要素数とする．このとき，$F(g^{y-ze})=e$，$z=s \bmod n$ より，任意の整数 $d\neq 0$ に対して，$g^{(y+sd)-z(e+d)}=e\neq e+d$ なので，$\mathbb{Z}_{2^b}\times\{A,A+1,\cdots,A+\Delta-1\}$ の集合を $\Delta-s(2^b-1)$ と $2s(2^b-1)$ の部分集合に分類したとき，前者には必ず 1 組の (e,y)，後者にはたかだか 1 組の (e,y) が存在する．このため，$\Phi=(2^b-1)(2^k-1)\geq s(2^b-1)$ より，$\Delta-\Phi\leq\mathcal{N}(F,A,\Delta)\leq\Delta+\Phi$ が成り立つ．

P と V^* の対話により，(x,e,y) の組を得る確率を $p(x,e,y)$ とする．このとき，

$$\begin{aligned}p(\alpha,\beta,\gamma) &= \sum_{0\leq r<2^a}\mathtt{Pr}\begin{bmatrix} g^r \bmod n=\alpha\\ F(g^r)=\beta\\ r+sF(g^r)=\gamma\end{bmatrix}\\ &=\frac{1}{2^a}\rho\begin{pmatrix}0\leq\gamma-s\beta<2^a\\ F(\alpha)=\beta\\ \alpha=g^{\gamma-z\beta} \bmod n\end{pmatrix}\end{aligned}$$

となる．ここで，ρ は特性関数であり，述語 P に関して，P が真ならば $\rho(P)=1$，偽ならば $\rho(P)=0$ となる．

同様に，TM_{V^*} が模倣アルゴリズムを実行することによって (x,e,y) の組を得る確率を $p'(x,e,y)$ とすると，

$$\begin{aligned}&p'(\alpha,\beta,\gamma)\\ &=\sum_{0\leq r<2^a}\mathtt{Pr}\left[\begin{matrix}e=\beta\\ y=\gamma\\ \alpha=g^{\gamma-z\beta} \bmod n\end{matrix}\;\middle|\; F(\alpha)=\beta\right]\\ &=\rho\begin{pmatrix}\Phi\leq\gamma<2^a\\ F(\alpha)=\beta\\ \alpha=g^{\gamma-z\beta} \bmod n\end{pmatrix}\Big/\mathcal{N}(F,\Phi,2^a-\Phi)\end{aligned}$$

となる．

1 ラウンドにおける $p(\alpha,\beta,\gamma)$ と $p'(\alpha,\beta,\gamma)$ の差異の総和を，

$$\Sigma_0=\sum_{\alpha,\beta,\gamma}\left|p(\alpha,\beta,\gamma)-p'(\alpha,\beta,\gamma)\right|$$

とする．このとき，

$$\begin{aligned}\Sigma_0&=2(1-\mathcal{N}(F,\Phi,2^a-\Phi)/2^a),\\ 2^a-\Phi&\leq\mathcal{N}(F,\Phi,2^a-\Phi),\\ \Phi&\leq(2^b-1)2q\quad(\because 2^{k-1}\leq q<2^k),\end{aligned}$$

より，

$$\Sigma_0<8q(2^b-1)/2^a$$

となる．これより，T ラウンドの繰返しによる差異は，

$$\begin{aligned}\Sigma_T&=\sum_{\substack{\alpha_i,\beta_i,\gamma_i\\ i\leq T}}\left|\begin{matrix}\mathtt{Pr}\left[(\alpha_i,\beta_i,\gamma_i)=(x_i,e_i,y_i)\right]\\ -\mathtt{Pr}\left[(\alpha_i,\beta_i,\gamma_i)=(x_i',e_i',y_i')\right]\end{matrix}\right|\\ &<8(2^b-1)\frac{qT}{2^a}\end{aligned}$$

となる．このとき，$qT/2^a$ は仮定より無視できるので，この認証アルゴリズムは，統計的ゼロ知識性を持つ．

□

5. 署名方式

表記として，$\mathcal{H}$ は $\mathcal{H}:\{0,1\}^*\to\{0,1\}^b$ となる

ハッシュ関数を表す．

鍵生成アルゴリズム 認証と同じ手順により，署名者の鍵を作成する．

署名生成アルゴリズム

入力：署名者の公開鍵 (n, g, z)，秘密鍵 s，メッセージ m

出力：メッセージ m に対する署名 (e, y)

Step 1 乱数 $r \in \mathbb{Z}_{2^a}$ を選ぶ．

Step 2 $x = g^r \bmod n$ を計算する．

Step 3 $e = \mathcal{H}(x, m)$ を計算する．

Step 4 $y = r + se$（$\mathbb{Z}$ 上）を計算する．

Step 5 (e, y) を出力する．

署名検証アルゴリズム

入力：署名者の公開鍵 (n, g, z)，メッセージ m，署名 (e, y)

出力：「受理」あるいは「不可」

Step 1 もし，$|y| \leq a+1$ が成り立たなければ「拒否」を出力してプロトコルを停止する．

Step 2 $x' = g^{y-ze} \bmod n$ を計算する．

Step 3 $e' = \mathcal{H}(x', m)$ を計算する．

Step 4 もし，$e = e'$ が成り立つのであれば「受理」を出力，そうでないならば「不可」を出力する．

5.1 安 全 性

これまで提案されてきた on the fly 署名[7),18)]は，いずれも認証を署名に変換した方式である．これらの方式の安全性は，文献 16)〜19) に記述されている．提案する署名の安全性は，これらの結果を用いることによって考察できる．

本節では，メッセージ m に対する署名を (x, e, y) とする．また，安全性の証明のために，素因数分解問題の困難性およびランダムオラクルモデルを仮定する．認証の場合，b は定数であり，ℓ は k に依存していた．この設定を変更することによって，以下のような補題を導ける．

補題 5.1 認証アルゴリズムに関して，$\ell = 1$ とし，$1/2^b$ および $2^b q/2^a$ は無視できるように設定する．このとき，認証アルゴリズムは，正直な検証者に基づく統計的ゼロ知識対話証明である．

証明 定理 4.2 と同様の考察により，この認証アルゴリズムは，完全である．

定理 4.3 および文献 22) より，もし，攻撃者 $\mathcal{A}$ が $\varepsilon > 1/2^b$ の確率で偽造に成功するならば，$O(1/\varepsilon + |n|^{O(1)})$ の計算量で，公開鍵 n の素因数を分解できる．よって，この認証アルゴリズムは，健全である．

P，V の対話に関して，以下のような確率的多項式時間 Turing 機械（シミュレータ）TM_V を考える．最初に，TM_V は定理 4.3 と同様の手順により，新しい公開鍵 $(n, \tilde{g}, \tilde{z})$ を設定する．次に，TM_V は 2 つの乱数 $e' \in Z_{2^b}$，$y' \in \{\Phi, \Phi+1, \cdots, 2^a - 1\}$ を選び，$x' = \tilde{g}^{y'-\tilde{z}} \bmod n$ を計算する．

定理 4.4 と同様に，P と V の対話により，(x, e, y) の組を得る確率を $p(x, e, y)$ とする．このとき，

$$p(\alpha, \beta, \gamma) = \frac{\rho\left(\begin{array}{c} \tilde{g}^{\gamma-\tilde{z}\beta} \bmod n = \alpha \\ \mathcal{R}(\cdot) = \beta \\ \gamma - s\beta \in \mathbb{Z}_{2^a} \end{array}\right)}{2^a}$$

となる．ここで，$\mathcal{R}$ は乱数 $\beta \in \mathbb{Z}_{2^b}$ を生成する関数である．同様に，TM_V に関する確率は，

$$p'(\alpha, \beta, \gamma) = \frac{\rho\left(\begin{array}{c} \tilde{g}^{\gamma-\tilde{z}\beta} \bmod n = \alpha \\ \mathcal{R}(\cdot) = \beta \\ \Phi \leq \gamma < 2^a \end{array}\right)}{\mathcal{N}(\mathcal{R}, \Phi, 2^a - \Phi)}$$

となる．このとき，

$$\Sigma_0 = \sum_{\alpha,\beta,\gamma} \left| p(\alpha, \beta, \gamma) - p'(\alpha, \beta, \gamma) \right|$$

とすると，$\Sigma_0 < 8 \cdot 2^b q/2^a$ であり，仮定より，$2^b q/2^a$ は無視できるので，この認証アルゴリズムは，統計的ゼロ知識性を持つ． □

定理 5.2 $1/2^b$ および $2^b q/2^a$ が無視できると仮定する．このとき，提案する署名方式は，適応的選択文章攻撃に対して，存在的偽造不可である．

証明 補題 5.1 は，署名方式に対して分岐補題（Forking Lemma）[16)]と呼ばれる補題を適用するための条件をすべて満たしている．以下では，分岐補題を用いて，素因数分解問題を解く特殊な 2 つの署名が作成できることを示す．詳細は文献 16) に記述されている．

適応的選択文章攻撃を行う攻撃者を $\mathcal{A}$ とする．ここで，$\mathcal{A}$ は確率的多項式時間 Turing 機械である．$\mathcal{A}$ を用いて以下のような確率的多項式時間 Turing 機械 M を構成する．

最初に，M は定理 4.3 と同様の手順により，新しい公開鍵 $(n, \tilde{g}, \tilde{z})$ を設定する．

署名オラクルをシミュレートするため，M は補題 5.1 と同様に，乱数 e'，y' を選び，x' を計算する．次に，M はメッセージ m に関する署名として，(x', e', y') を $\mathcal{A}$ に返す．

また，ランダムオラクルの答えとして，M は $\mathcal{A}$ の乱数テープを固定して，$\mathcal{A}$ のアルゴリズムを 2 度起動する．そして，2 度目の質問に対しては，別のランダ

ムオラクルを用いることによって，1 度目とは異なった答えを返す．このような操作を繰り返すことにより，最終的に M は，(x, e, y)，(x, e', y') というような同一メッセージに対する 2 つの異なった署名を得ることができる．

最後に，定理 4.3 と同様の操作を行うことにより，M は無視できない確率で公開鍵 n を素因数分解することができる．しかしながら，このことは素因数分解問題の困難性の仮定に矛盾する． □

6. パラメータと実装に関する記述

本章では，提案方式のパラメータが満たすべき条件や，実装における注意点について記述する．

6.1 安全性に関するパラメータ

a と b の関係 a，b は，$a \geq b + k + \kappa$ という条件を満たす．ここで，k はセキュリティパラメータであり，$q < 2^k$ である．また，κ は情報リークパラメータであり，$1/2^\kappa$ が k 無視できるように設定する．推奨値は，$k \geq 160$，$\kappa \geq 80$ である．

b の値 実装環境において，b の値は認証と署名では異なる．署名の場合，誕生日攻撃（birthday attack）によるハッシュ値の衝突を考慮に入れるため，認証における $b\ell$ より，値を大きくする必要がある．認証の場合，推奨値は数回程度の ℓ に対して $b\ell \geq 40$ であり，署名の場合，推奨値は $b \geq 80$ である．

c の条件 もし，$|y| > |ez|$ ならば，攻撃者は証明者/署名者になりすまし，通常のアルゴリズムに従って $x = g^r \bmod n$ を計算することにより，検証式 $x = g^{y-ze} \bmod n$ となる y を容易に計算できる．このような攻撃を防ぐため，$c \geq k + 2\kappa$ という条件を満たす必要がある．

b と ℓ の関係 認証と署名では，b と ℓ の関係に違いがある．認証の場合，b は定数であり，ℓ は k に依存する．このため，$1/2^{b\ell}$ は無視できるように設定する．また，認証アルゴリズムは多項式時間で実行できなければならないので，k は ℓ の多項式で抑えられる．これに対し，署名の場合 $\ell = 1$ に設定される．このため，b は k に依存しており，$1/2^b$ は無視できるように設定する．

q のサイズ 攻撃者は，証明者/署名者によって生成された x から，$\log_g(x) = r' \in \mathbb{Z}_q^*$ を計算できれば，提案方式を破ることができる．r' を計算する効率的なアルゴリズムの 1 つに，計算量が $O(\sqrt{q})$ となる baby-step giant-step 法[9] がある．q のサイズは，このような攻撃を考慮して設定する．推奨値は，$|q| \geq 160$ である．

6.2 n のサイズと素因数の数

提案方式を破るために，公開鍵 n を素因数分解する攻撃が考えられる．現在報告されている最高速の素因数分解アルゴリズムは，数体ふるい法[10] であり，計算量は，n のサイズに依存する．一方，素因数のサイズに依存する効率的なアルゴリズムとして，楕円曲線法[11] がある．合成数を素因数分解したとき，どちらが高速になるかは，合成数のサイズと素因数の数の関係によって異なる．アルゴリズムの計算量やその他の条件を，文献 23) のように設定した場合，合成数のサイズが 1024 ビット，素因数の数が 3 ならば，数体ふるい法の方が高速である．しかしながら素因数の数が 4 になると，逆転が起こる．このことから，提案方式を $|n| = 1024$，$t = 3$，PS 方式を $|n| = 1024$ と設定すると，それぞれの n を素因数分解するアルゴリズムの中で最も高速なのは，両者とも数体ふるい法となり，両者の計算量は同程度になる．

6.3 鍵とハッシュ関数の選択

g の選択 以下のアルゴリズムにより，g を選択する．最初に，$\mathbb{Z}_n^*$ から乱数 α を選択する．このとき，α の位数が，$\prod_{i=i}^{t} q_i$ の倍数となるまで繰り返す．1 回の試行でこのような条件を満たす確率は，$2^t\varphi(\prod_{i=i}^{t} q_i)/\varphi(n) \approx 1 - \frac{t}{\sqrt[t]{n}}$ となる．$\mathrm{ord}_n(\alpha) = u$ とすると，$\alpha^{u/q} \bmod n$ を計算し，その結果を g とする．

$\mathcal{H}$ の選択 $\mathcal{H}$ がランダム関数であると仮定した場合，素因数分解が困難ならば，提案する署名方式は 5.1 節に記述した安全性を満たす．しかしながら，このような関数は実際には存在しないため，実装環境においては，衝突困難性[2] を満たすことを目標に設計された MD5[20] や SHA-1[14] を利用する．

7. データサイズの効率化

本章では，提案方式の鍵生成アルゴリズムを変更することにより，公開鍵のサイズを削減する方法を述べる．ここで，$\mathcal{H}'$ は $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^c$ となるハッシュ関数である．

鍵生成アルゴリズム Step 1 と Step 2 は従来の提案方式と同じである．Step 3 から Step 5 を次のように変更する．

Step 3 $z = \mathcal{H}'(n, g)$ を計算する．

Step 4 $s = z \bmod q$ を計算する．

Step 5 証明者/署名者の公開鍵を (n, g)，秘密鍵を s とする．

表 1 署名方式に関する性能評価
Table 1 Performances on signature schemes.

方式	前提となる数学の問題	事前計算 (M)	署名生成	署名検証 (M)	公開鍵 (ビット)	秘密鍵 (ビット)	署名長 (ビット)
提案方式 $\|n\| = 1024,\ t = 3$ $\kappa = 80$	素因数分解	171	80×342	873	2048	342	582
PS 方式[18] $\|n\| = 1024,\ \|A\| = 672$	素因数分解	384	80×512	1656	2048	513	752
GPS 方式[17] $\|n\| = 1024$	離散対数法 n	384	80×1024	1796	3072	1024	1264
Feige-Fiat-Shamir [4] $\|n\| = 1024,\ k = 80$	素因数分解	1	mult 41 mod 1024	24	82944	81920	1104
RSA [21] $\|n\| = 1024,\ e = 3$	RSA	0	mult 1536 mod 1024	2	1024	1024	1024
ESIGN [5] $\|n\| = 1024$	e 乗根近似	1	mult 1 mod 342	3	1024	1024	1024
Guillou-Quisquater [8] $\|n\| = 1024,\ k = 80$	RSA	48	mult 121 mod 1024	313	2176	1024	1104
El Gamal [3] $\|p\| = 768$	離散対数法 p	648	mult 2 mod 768	1945	2304	768	1536
Schnorr [22] $\|p\| = 768,\ \|q\| = 160$	離散対数法 p	135	mult 1 mod 160	203	2464	160	240
DSA [13] $\|p\| = 768,\ \|q\| = 160$	離散対数法 p	135	mult 2 mod 160	270	2464	160	320

検証者は，$\mathcal{H}'$ と公開鍵 (n, g) から z を計算することによって，検証アルゴリズムを実行することができる．このとき，公開鍵は従来の (n, g, z) から (n, g) となり，鍵のサイズは約 2/3 になる．

8. 既存方式との比較

本章では，提案する署名と既存方式を比較することによって，提案方式の性能を評価する．

提案方式と既存方式の性能評価を，**表 1** に示す．表に記述された計算量に関しては，いずれも原始的な binary 法[9] を用いて計算しているが，これ以外の方式を用いた場合でも，計算量の評価結果は相対的に大きく変わらない．

表記として，M は法のサイズを 1024 ビットとした場合の乗算剰余の回数，$\alpha \times \beta$ は α ビットと β ビットの整数の乗算，$\binom{\texttt{mult}\ \gamma}{\texttt{mod}\ \delta}$ は，法のサイズを δ とした場合の乗算剰余の回数 γ を表す．また，事前計算において，可能であるならば中国人剰余定理を利用して計算量を削減している．ただしこの場合，署名者は n の素因数を秘密情報として持つ必要がある．

以下では，提案方式と表に記述された既存方式の特徴について考察する．最初に，これまで提案された on the fly 方式と提案方式を比較する．On the fly 方式以外については，安全性の仮定が素因数分解と離散対数に基づく方式に分類し，それぞれの方式と提案方式を比較する．

8.1 On the fly 方式

提案方式と PS 方式の公開鍵 n のサイズを同一にしたとき，提案方式は素因数の数を増やすことにより，秘密鍵のサイズをより小さくできる．これにより，提案方式は計算処理量とデータサイズの両方を削減できる．たとえば，提案方式を，$|n| = 1024,\ t = 3$，PS 方式を $|n| = 1024$ と設定した場合，事前計算，署名生成，検証の計算量は，それぞれ 55%，33%，46%削減できる．また秘密鍵および署名のサイズは，それぞれ 33%，23%削減できる．このとき，それぞれの方式を n の素因数分解を用いて破る場合の計算量は，6.2 節の記述より，現在のところ同程度である．

GPS [17] の安全性について，前提となる数学の問題は，どのような攻撃を想定するかによって異なる．想定する攻撃が提案方式と同様の場合，すなわち攻撃者が可能性のある任意の公開鍵に対して偽造文を作成できるならば，法 n（RSA 法）に対する離散対数問題との等価性が示されている[17]．また，攻撃者が 1 組の固定された公開鍵に対してのみ偽造文が作成できると仮定した場合，安全性の証明のために g の位数と秘密鍵のサイズを大きくする必要があり，結果として計算処理量とデータサイズは増大する[15]．

8.2 素因数分解に基づく方式

Fiege-Fiat-Shamir 署名[4] の安全性は，提案方式と同様，素因数分解問題の困難性を仮定している[16]．また，設定を $k = 80$, $|n| = 1024$ とすると，公開鍵のサイズは，$k \times |n| = 81920$ となり，提案方式の 2048 ビットより大きくなる．RSA 署名[21] と Guillou-Quisquater 署名[8] の安全性は，RSA 問題の困難性を仮定しているため，提案方式より仮定が強い．また，RSA 署名は事前計算処理が不要な代わりに，署名生成に多くの計算量が必要である．ESIGN[5] は，公開鍵 n の構造が p^2q $(p, q \in \mathbb{N}_{\text{prime}})$ であり，この特殊な n から素因数 p, q を計算する問題は，2 章で定義された素因数分解問題に帰着する．しかしながら，その逆は示されていない．また，安全性の仮定は e 乗根近似と呼ばれる問題の困難性であるが，この問題は RSA 問題に帰着するため，提案方式より安全性の仮定が強い．

上記の既存方式[4),5),8),21)] において，署名者あるいは検証者は，署名時/検証時に法 n の任意点を指数演算しなければならない．これに対し提案方式は g という固定点に対して指数演算を行う．固定点の指数演算アルゴリズムはテーブルを利用する等により計算の高速化が可能[1] なので，提案方式はこれらの既存方式と比べて計算量を削減できる．

8.3 離散対数に基づく方式

ElGamal 署名[3] は，公開鍵のサイズを $p = 768$ とすると，署名サイズが $2 \times |p| = 1536$ になり，提案方式より大きくなる．Schnorr 署名[22] と DSA[13] は，原始元の代わりに，位数が $q|p-1$ $(q \in \mathbb{N}_{\text{prime}})$ となる元 g を用いることにより，署名サイズおよび計算処理量を小さくしている．また Schnorr 署名はハッシュ関数の効率的な利用により，署名サイズのさらなる縮小を実現している．これらの効率的な手法は提案方式においても利用されている．

上記の既存方式[3),13),22)] において，公開鍵のパラメータ数は提案方式より多い，このため，署名者が単独で鍵を用いる場合，公開鍵のサイズは提案方式より大きくなる．

9. ま と め

今後，さらに発展が予想される情報化社会に対応するため，安全でかつ効率的な署名技術は不可欠である．これらの要求に対応するため，本論文では on the fly という機能を持った，効率的な署名方式を提案した．これは，Poupard-Stern 署名を改良することによって得られた方式であり，PS 方式と同程度の安全性を持ちながら，計算処理とデータサイズの点で，PS 方式より優れた性能を持つ．

本論文では，最初に PS 方式の特徴を解析し，この方式が持つ本質的な問題点を指摘した．次に，この問題点を解決するため，どのような特徴が望まれるかについて検討した後，これらの問題点が改善された新しい署名方式を提案した．また，この署名方式は認証を変換した方式なので，新しい認証方式についても提案している．安全性については，これまでに研究されてきた成果をふまえ，考察を行った．また，提案方式で用いられるパラメータが満たすべき条件や，生成方法についても検討した．次に，現在報告されている高速な素因数分解アルゴリズムの特徴を説明し，これらアルゴリズムの特徴を考慮して，パラメータを設定することにより，提案方式が PS 方式と同程度の安全性を持ち，かつ計算処理，伝送量の点で優れた方式を構築できることを示した．最後に，これまで提案された主要な署名方式と比較することによって，提案方式が安全性と実用性を兼ね備えていることを示した．

参 考 文 献

1) Brickell, E.F., Gordon, D.M., McCurley, K.S. and Wilson, D.B.: Fast exponentiation with precomputation, *Eurocrypt '92*, LNCS, Vol.658, pp.200–207, Springer-Verlag (1996).
2) Damgard: Collision Free Hash Functions and Public Key Signature Schemes, *Eurocrypt '87*, LNCS, Vol.304, pp.203–216, Springer-Verlag (1988).
3) ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inf. Theory*, Vol.IT-31, No.4, pp.469–472 (1985).
4) Feige, U., Fiat, A. and Shamir, A.: Zero-Knowledge Proofs of Identity, *Journal of Cryptology*, Vol.1, pp.77–95 (1988).
5) Fujioka, A., Miyaguchi, S. and Okamoto, T.: ESIGN: An Efficient Digital Signature Implementation for Smart Cards, *Eurocrypt '91*, LNCS, Vol.547, pp.446–457, Springer-Verlag (1992).
6) Giraut, M.: An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number, *Eurocrypt '90*, LNCS, Vol.473, pp.481–486, Springer-Verlag (1991).
7) Giraut, M.: Self-certified public keys, *Eurocrypt '91*, LNCS, Vol.547, pp.490–497, Springer-Verlag (1992).
8) Guillou, L.C. and Quisquater, J.J.: A "Paradoxal" Identity-Based Signature Scheme

Resulting from Zero-Knowledge, *Crypto '88*, LNCS, Vol.403, pp.216–231, Springer-Verlag (1989).
9) Knuth, D.E.: Sorting and Searching, *The Art of Computer Programming*, 2nd edition, Vol.3, Addison-Wesley (1998).
10) Lenstra, A.K., Lenstra, H.W.J., Manassse, M.S. and Pollard, J.M.: The number field sieve, *Proc. ACM Annual Symposium on Theory of Computing*, pp.564–572 (1990).
11) Lenstra, H.W.J.: Factoring Integers with elliptic Curves, *Annals of Mathematics*, Vol.126, pp.649–673 (1987).
12) Miller, G.: Riemann's hypothesis and test for primality, *Journal of Computer and System Sciences*, Vol.13, pp.300–317 (1976).
13) National Institute of Standards and Technology (NIST): Digital Signature Standard (DSS), *Federal Information Processing Standards* (1991).
14) National Institute of Standards and Technology (NIST): Secure Hash Standard (SHS), *Federal Information Processing Standards* (1995).
15) Poincheval, D.: The Composite Discrete Logarithm and Secure Authentication, *PKC '00*, LNCS, Vol.1751, pp.113–128, Springer-Verlag (2000).
16) Poincheval, D. and Stern, J.: Security Arguments for Digital Signatures and Blind Signatures, *Journal of Cryptology* (2000).
17) Poupard, G. and Stern, J.: Security Analysis of a Practical "on the fly" Authentication and Signature Generation, *Eurocrypt '98*, LNCS, Vol.1403, pp.422–436, Springer-Verlag (1998).
18) Poupard, G. and Stern, J.: On The Fly Signatures based on Factoring, *Proc.6th CCS*, pp.48–57, ACM Press (1999).
19) Poupard, G. and Stern, J.: Short Proofs of Knowledge for Factoring, *PKC '00*, LNCS, Vol.1751, pp.147–166, Springer-Verlag (2000).
20) Rivest, R.L.: The MD5 Message-Digest Algorithm, Internet Request for Comments, RFC 1321 (1992).
21) Rivest, R.L., Shamir, A. and Adleman, L.M.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).
22) Schnorr, C.P.: Efficient signature generation by smart cards, *Journal of Cryptology*, Vol.4, pp.161–174 (1991).
23) Silverman, R.D.: A Cost-Based Security Analysis of Symmetric and Asymmetric Key Length, RSA Laboratories, CryptoBytes, Bulletins, No.13 (1999).

(平成 12 年 12 月 13 日受付)
(平成 13 年 6 月 19 日採録)

岡本　健（学生会員）

平成 8 年京都工芸繊維大学工芸学部電子情報工学科卒業．平成 11 年北陸先端科学技術大学院大学情報科学研究科情報システム学専攻博士前期課程修了．現在同大学院博士後期課程在学中．平成 12 年度，山下記念研究賞受賞．現在，情報セキュリティの研究に従事．

多田　充（正会員）

平成 4 年東北大学理学部数学科卒業．平成 7 年同大学院情報科学研究科博士前期課程修了．平成 10 年同博士後期課程修了．同年北陸先端科学技術大学院大学情報科学研究科助手．現在に至る．数理論理，暗号理論に関する研究に従事．博士（情報科学）．平成 12 年電子情報通信学会「SCIS 論文賞」受賞．

宮地　充子（正会員）

昭和 63 年大阪大学理学部数学科卒業．平成元年同大学院修士課程修了．同年松下電器産業株式会社入社．平成 10 年北陸先端科学技術大学院大学・情報科学研究科助教授．現在に至る．情報セキュリティの研究に従事．博士（理学）．SCIS93 若手論文賞，科学技術庁注目発明賞各受賞．電子情報通信学会会員．