最近のデジタル署名における理論研究動向について

宇根正志（うねまさし）／岡本龍明（おかもとたつあき）

要　旨

本稿は、これまでに提案されている主要なデジタル署名方式のアルゴリズムや標準化動向を紹介したうえで、最近明らかになったRSA署名に対する攻撃法や、安全性が証明されているデジタル署名方式の理論研究の動向について説明するものである。

従来、デジタル署名方式の安全性評価は、既存の攻撃法を前提とした評価が中心であった。しかし、デジタル署名方式の実装環境が多様化する中、これまで検討されていなかった攻撃法が有効になる可能性が高まっている。こうした中、1999年8月、RSA署名を利用したデジタル署名方式の国際標準ISO/IEC 9796に対して有効な攻撃法が提案され、本国際標準の標準化を担当するISO/IEC JTC1/SC27は、同年10月にISO/IEC 9796を取り下げることを決定した。この結果、既存の攻撃法を前提とした安全性評価では不十分であり、一定の数学的な仮定のもとで効率的な攻撃法が存在しないことを証明する「安全性証明」のような理論的な安全性評価が必要との認識が強まっている。

最近では、安全性が証明されているとともに、処理速度の面で実用性の高いデジタル署名方式が相次いで提案されており、ISOやIEEE等では、安全性が証明されている署名方式の国際標準への採用が検討されている。今後、デジタル署名を利用する際には、実装技術に関する研究成果に加えて、安全性証明に関する研究等、最新の理論的な研究成果を十分考慮することが必要であろう。

キーワード：安全性証明、公開鍵暗号、デジタル署名、RSA署名

本稿は、1999年11月1日に日本銀行で開催された「第２回情報セキュリティシンポジウム」への提出論文に加筆・修正を施したものである。

宇根正志　日本銀行金融研究所研究第２課（E-mail: masashi.une@boj.or.jp）
岡本龍明　日本電信電話株式会社情報流通プラットフォーム研究所
（E-mail: okamoto@sucaba.isl.ntt.co.jp）

1．はじめに

デジタル署名は、デジタルデータを署名者固有の情報を用いて変換することによって生成されるデータであり、①データの作成者の特定（ユーザー認証）、②データにおける改ざんの検出（メッセージ認証）、③いったん生成した署名に対して、その署名を生成した事実の否認の防止（否認防止）といった機能を実現する。このため、デジタル署名は、オープンなネットワーク上で交信される情報の安全性を確保するための重要な認証技術の１つとして、幅広い分野において実用化されている。

金融分野でデジタル署名を利用したシステムを構築する際には、最新の理論研究や技術動向を踏まえたうえで安全性評価を実施し、必要とされるセキュリティ水準を確保することができるか否かを十分に検討する必要がある。従来、デジタル署名方式における安全性評価においては、これまでに効率的な攻撃法が提案されていないから安全であるとか、既存の攻撃法に対する対策が講じられているから安全であるといった、既存の分析手法や攻撃法のみを前提とした評価が中心であった。

しかし、こうした評価方法では、これまでに見つかっていない攻撃法に対する安全性を評価することは不可能である。こうした問題点の存在を強く印象付けた事例として、デジタル署名方式の国際標準ISO/IEC 9796に対する攻撃法の発表が挙げられる。ISO/IEC 9796はRSA署名をベースとした方式であり、既存の攻撃法を前提とした評価によって十分な安全性が確保されているとみられていた。しかし、攻撃法が示されたことによって、ISO/IEC 9796の署名を効率的に偽造することが可能であることが示された。このように、デジタル署名方式の実装環境が多様化する中、従来想定されていなかった攻撃法が有効となる可能性があり、既存の攻撃法のみを前提とした安全性評価には限界があることが明らかとなった。

最近のデジタル署名に関する理論研究では、デジタル署名方式の安全性を理論的に証明する「安全性証明の研究」が注目を集めている。デジタル署名方式に対する安全性の証明は、①「攻撃者が利用可能な情報」、「デジタル署名のアルゴリズムに必要とされる仮定」、「署名の偽造が可能なデータ」の間の関連性を理論的に示し、デジタル署名方式の安全性に関する性質を明確にするとともに、②証明の内容を比較することで、複数のデジタル署名方式の安全性レベルを比較可能にする、といった利点を有している。近年、処理速度の面で高い実用性を有するとともに、安全性が証明されているデジタル署名方式が相次いで提案されており、安全性証明に関する理論研究の成果を、実際にデジタル署名を利用する際に活用することが可能な状況になりつつある。

本稿では、こうしたデジタル署名方式の安全性評価研究を中心に、デジタル署名における最新の理論研究動向について説明する。まず、第２章において、デジタル署名の機能、要件、実現方法について説明したうえで、第３章では、これまでに提案されている主要なデジタル署名方式を整理し、その概要について説明する。第４章では、デジタル署名方式の安全性評価について、RSA署名に対する攻撃法を中心に説明する。最後に、第５章では、デジタル署名方式の安全性証明に

関する研究や標準化の動向について説明する。

2．デジタル署名と公開鍵暗号

（1）デジタル署名の機能と要件

小切手や契約書等の紙ベースでの署名や捺印は、その特有の形状から署名者を一意に特定し、署名が付された文書の作成者等を確定させる役割を有している。デジタル署名は、こうした紙ベースでの署名の機能をデジタルデータにおいて実現するデータである。各署名者は、自分固有の情報（署名生成鍵）を用いて署名の対象となるデータを変換してデジタル署名を生成する。デジタル署名の機能は以下の３点である（Menezes et al.［1997］）。

①デジタル署名の生成者を特定することができる（ユーザー認証機能）。
②デジタル署名の対象であるデータが署名生成者以外によって改ざんされた場合、署名を検証するための情報（署名検証鍵）によって改ざんの事実を検出することができる（メッセージ認証機能）。
③署名者は、いったんデジタル署名を生成すると、そのデジタル署名の基になっているデータを作成した事実を後で否定できない（否認防止機能）。

これらの機能を満足するためにデジタル署名に求められる要件は以下の３つである（ISO/IEC［1997］）。

①署名者が自分固有の署名生成鍵を秘密に管理する限り、任意のデータに対するデジタル署名を他人が偽造することは困難である。
②デジタル署名やその対象となるメッセージから、別のメッセージに対する署名を偽造したり、署名生成鍵を計算したりすることは困難である。
③同じ署名生成鍵によって同じデジタル署名が生成される異なる複数のメッセージを見つけることは困難である。

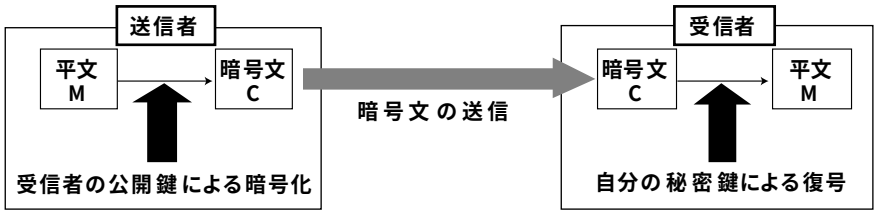
これらの要件を満たすデジタル署名は公開鍵暗号によって実現される[1]。

1 共通鍵暗号を用いたメッセージ認証方式としては、MAC（Massage Authentication Code）を利用する方法が一般に用いられる。MACを利用する方法では、メッセージの送信者と受信者の間であらかじめ秘密鍵を共有し、メッセージの送信者が共通鍵暗号方式によってメッセージから数十bit程度のメッセージダイジェストをMACとして生成する。送信者はメッセージとMACを受信者に送信し、受信者は、メッセージからMACを生成して送付されたMACと同一か否かを検証する。MACを利用する方式は、送信者と受信者がMACの鍵を共有する必要があり、特定の二者間の通信等に用途が限定される。このため、不特定多数の利用者の存在を前提とするオープンなネットワーク上での利用を想定したデジタル署名とは、その用途が異なっている。

（2）公開鍵暗号によるデジタル署名の実現方法

公開鍵暗号は、暗号化用の鍵と復号用の鍵が異なる暗号方式であり、一方の鍵から他方の鍵を算出することが計算量的に困難である[2]ため、どちらか一方の鍵を公開することができる。通常、暗号化に利用される鍵（公開鍵）が公開され[3]、復号に利用される鍵（秘密鍵）が秘密に管理される。公開鍵暗号を利用した暗号通信の手順は以下の図１のとおり。

図１　公開鍵暗号によるデータ守秘機能（暗号通信）



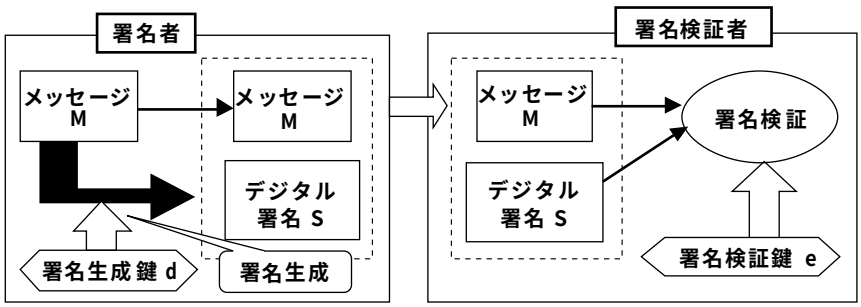
（ステップ1）送信者は、受信者の公開鍵によって平文Mを暗号化して暗号文Cを生成し、受信者に送信する。
（ステップ2）受信者は、暗号文Cを自分の秘密鍵で復号する。

公開鍵暗号をデジタル署名に利用する場合、デジタル署名の生成は、署名者が秘密鍵を用いて署名対象データを変換することによって行われる。このため、公開鍵暗号における秘密鍵が署名生成鍵に対応する。一方、デジタル署名の検証は、不特定多数の署名検証者が、署名者の公開鍵を利用して署名者固有の変換の正当性を確認することによって行われる。このため、公開鍵暗号における公開鍵が署名検証鍵に対応する。署名生成鍵を所有する者は唯一人であるため、署名者以外の個人がデジタル署名を偽造することは計算量的に困難となる。デジタル署名の生成・検証方法の概略は図２のとおり。

2 「計算量的に困難である」とは、その計算を行うことは理論的には可能であるものの、実際にその計算を実行するには計算量が非常に大量となり、膨大な費用と時間を必要とすることから、事実上不可能であることを意味する。どの程度の計算量が事実上不可能であるかは、その時々の技術条件等によって左右される。

3 公開鍵を利用するためには、その公開鍵がある特定の利用者のものであることや公開鍵が第三者によって改ざんされていないこと等を確認する仕組みが必要となる。公開鍵の所有者や有効期限等の属性情報やその真正性は、公開鍵証明書と呼ばれるデータによって確認される仕組みが利用されるケースが多い。公開鍵証明書には、証明の対象となる公開鍵やその所有者・有効期間等の情報のほか、それらの情報の真正性を確保するためのデジタル署名が含まれる。公開鍵証明書のデジタル署名は、認証機関と呼ばれる信頼できる第三者機関によって生成される。こうした公開鍵暗号を実用化するための仕組みは、公開鍵インフラ（PKI：Public Key Infrastructure）と呼ばれる。PKIについては谷口［2000］を参照。

図２　公開鍵暗号を利用したデジタル署名の生成・検証



（ステップ１）署名者は、署名生成鍵dでメッセージMを変換し、署名Sを生成。
（ステップ２）署名者は、メッセージと署名、または署名のみを署名検証者に送信。
（ステップ３）署名検証者は、署名検証鍵e、M、Sによって署名の正当性を検証。

３．デジタル署名の分類と主要な方式

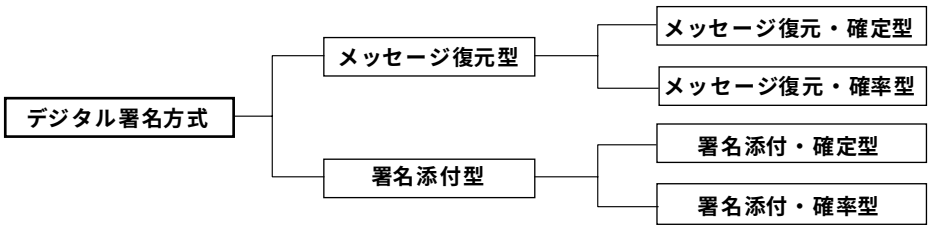
（１）デジタル署名の分類

デジタル署名方式は、①デジタル署名の形態、②安全性の根拠となる数学問題によって分類される。

イ. デジタル署名の形態による分類

デジタル署名方式は、デジタル署名の形態から分類される。第一に、署名対象のデータと署名が別々になっている（署名添付型）か、もしくは署名からデータを復元することができる（メッセージ復元型）かによって分類される。また、第二に、同一のデータに対する署名がつねに同一（確定型）か、もしくは同一のデータに対する署名が常に異なる（確率型）かによって分類される（Menezes et al.［1997］、図３参照）。

図３　デジタル署名の形態による分類



（出典）Menezes et al.［1997］

メッセージ復元型のデジタル署名方式は、署名からメッセージを復元することができるため、処理の対象となるデータ量を署名添付型に比べて少なくすることが可能であり、ICカード等比較的計算能力が制限される実装形態に適している。ただし、署名を生成可能なメッセージの大きさが制限されるほか、署名検証者が署名の対象となっているメッセージを得るためには署名を一定の手順で変換しなければならない。一方、署名添付型のデジタル署名方式は、処理の対象となるデータ量が多くなるものの、署名検証者が署名の検証を行うことなしにメッセージを得ることができる。

また、確率型のデジタル署名方式では、一定のメッセージの署名が毎回異なることから、メッセージと署名のペアを利用した攻撃法の適用が困難となり、確定型のデジタル署名に比べて安全性が高まるとみられている。ただし、署名生成の際に乱数の生成等の追加的な処理が必要となる。

各署名方式の特徴点を整理すると、表１のとおり。

表１　デジタル署名方式の各形態の特徴点

	長所	短所
メッセージ復元型	・処理対象となるデータ量を比較的少なくすることが可能。ICカード等計算能力が制限される実装環境に適している。	・署名生成可能なメッセージのサイズが制限される。 ・メッセージの入手の際には、必ず署名の検証を実行し、メッセージを復元する必要がある。
署名添付型	・署名対象となるメッセージのサイズが制約されない。 ・メッセージを入手するだけであれば署名の検証を実行する必要はない。	・処理対象となるデータ量（メッセージと署名データ）が比較的多くなる。
確定型	・乱数生成等が不要。	・署名とメッセージの対応関係が一定であり、署名とメッセージのペアを利用した攻撃が適用可能。
確率型	・署名とメッセージのペアを利用した攻撃法が適用困難。	・安全な乱数の生成等が必要。

ロ. 安全性の根拠となる数学問題による分類

デジタル署名方式は、その安全性の根拠となる数学問題によって分類される。これまでに、さまざまな数学問題を利用したデジタル署名が提案されているが、現在実用化されている主要なデジタル署名では、①素因数分解問題、②有限体上の乗法群における離散対数問題（以下、離散対数問題と呼ぶ）、③楕円曲線上の有限可換群における離散対数問題（以下、楕円離散対数問題と呼ぶ）の３種類の問題が利用されている。

①素因数分解問題

素因数分解問題は合成数を複数の素因数に分解する問題である。後述するRSA署名やESIGNでは、100桁程度の大きな素数によって構成される合成数を素因数分解

することが困難であることを利用している。RSA署名に利用される素因数分解問題は以下のとおり。

合成数 n（= p×q、ただしpとqは素数）を素因数分解せよ。

素因数分解問題の解法については、これまでにさまざまな方式が提案されている。素数pとqの形態によって、最も効率的な解法が異なっており、pとqのサイズが同一であり、pとqにそれぞれ特殊な性質が存在しない場合、現時点でAdleman-Lenstra版の数体ふるい法が最高速の解法である[4]。主要な素因数分解アルゴリズムを整理すると、表２のとおり。

表２　合成数nの主な性質と最も効率的な素因数分解アルゴリズム

合成数 n（= p×q）の性質	最も効率的な解法
p±1またはq±1が小さな素数の積となる場合	p＋1法、p－1法
|p－q|が小さな素数の積となる場合	Fermat法
上記２つの条件をいずれも満足せず、pとqのサイズが同一の場合	Adleman-Lenstra版数体ふるい法

②離散対数問題

離散対数問題は、有限体上において対数を計算する問題である。実数体上で対数を計算することは大きな数であっても容易であるが、有限体上では、有限体の要素の個数が大きくなるにつれて、必要となる計算量が指数関数的に増加することが知られている。素因数分解問題の困難性と理論的に厳密な比較は示されていないが、最高速の解法の必要計算量によってほぼ同程度困難であるとみられている。

素数pと、pを法とする乗法群の原始根aを選び、ある整数xに対して、$b = a^x \bmod p$を計算する。このとき、p、a、bを所与として、次の等式を満たすxを求めよ。

$$b = a^x \bmod p$$

離散対数問題には、素因数分解問題と同様に、これまでにさまざまな解法が提案されてきた。最も効率的な離散対数問題の解法は、法pや有限体のタイプに依存しており、（i）p–1の最大素因数がlog p以下の場合にはPohlig-Hellmanのアルゴリズム（Pohlig and Hellman［1978］）が最も効率的であり、（ii）それ以外の場合には、有限体のタイプによって、Schirokauer版の数体ふるい法（Schirokauer［1993］）、Adlemanの関数体ふるい法（Adleman［1994］）、Schirokauer-Weber-Dennyのアルゴ

4 素因数分解問題の解法の詳細については、宇根・岡本［1999］を参照。

リズム（Schirokauer et al.［1996］）が効率的な解法となる（以下の表３参照）[5]。

表３　離散対数問題のタイプと主な解法

有限体のタイプ		最も効率的な解法
p−1の最大素因数がlog p以下の場合		Pohlig-Hellmanのアルゴリズム
上記以外の場合	有限体F_p^kに対して、$k < (\log p)^{1/2-\varepsilon}$（$\varepsilon$: ある正の定数）となる場合	Schirokauer版数体ふるい法
	有限体F_p^kに対して、$k \geqq (\log p)^2$となる場合	Adlemanの関数体ふるい法
	有限体F_p^kに対して、$(\log p)^{1/2-\varepsilon} \leqq k < (\log p)^2$となる場合	Schirokauer-Weber-Dennyのアルゴリズム

③楕円離散対数問題

離散対数問題は、有限体上の乗法群において対数を計算するというものであった。これに対して、楕円離散対数問題は、有限体によって定義された楕円曲線上の点が有限可換群をなすことを利用し、この有限可換群において対数を計算するという問題である。楕円離散対数問題には、ある種の楕円曲線を除いて高速解法が適用困難である。このため、楕円離散対数問題に基づく方式は、有限体上の乗法群における離散対数問題に基づくデジタル署名方式に比べて、安全性を維持しつつ鍵長を短縮できるとされている。鍵長を短縮することが可能となれば、処理速度を向上させることが可能となる。楕円離散対数問題を公開鍵暗号やデジタル署名に利用するアイデアはKoblitz［1987］とMiller［1986］によって発表されており、比較的新しい方式である。このため、素因数分解問題や離散対数問題に比べて研究の蓄積は少ないが、デジタル署名方式の有望な分野として注目を集めている。

まず、楕円曲線および有限体上の楕円曲線は以下のように定義される。

【楕円曲線】

３次曲線（関数$F(x,y)$の次数が３となる代数方程式 $F(x,y)=0$ の解の集合）のうち、特異点（$F(x,y)$のxおよびyに関する偏微分係数が0となる(x,y)）を含まない点の集合。

【有限体上の楕円曲線】

要素の個数 p（$p>3$の素数）である有限体F_pにおいて、

$$\{(x, y) \mid y^2 = x^3 + ax + b(\mathrm{mod}\ p)\}\ (4a^3 + 27b^2 \neq 0,\ a,\ b \in F_p)$$

を満足する点 (x,y)（x,y はF_pの要素）の集合。

楕円離散対数問題は以下のとおり。

5 これらの解法の詳細については、宇根・岡本［1999］を参照。

素数pに対して有限体 F_p 上の楕円曲線に無限遠点Oを加えた集合を$E(F_p)$とし、$E(F_p)$上の２点間の加法演算を定義すると、$E(F_p)$は有限可換群となる（無限遠点Oはこの有限可換群の単位元となる）。

$E(F_p)$上の点A、B（$A \neq B$）を選び、AとBがある自然数xに対してA=xBという関係にある（定義された加法演算によって点Bをx回加えると、点Aとなる）とする。

このとき、p、$E(F_p)$、A、Bを所与として、A = xBを満たす自然数xを求めよ。

楕円離散対数問題の解法は、適用する楕円曲線の種類によって異なる。楕円曲線の中でも、(i) トレース[6]が0となる楕円曲線（超特異楕円曲線と呼ばれる）にはMOV帰着（Menezes et al.［1991］）、FR帰着（USアルゴリズム、Frey and Rück［1994］、内山・斎藤［1998］）といった手法によって、楕円離散対数問題を離散対数問題に帰着させることが可能となるほか、(ii) トレースが１となる楕円曲線（アノマラス（Anomalous）曲線と呼ばれる）にはSSSAアルゴリズム（Smart［1999］、Semaev［1998］、Satoh and Araki［1998］）と呼ばれる高速の解法が提案されている。また、(iii) トレースが２となる楕円曲線に対しても、FR帰着（USアルゴリズム）によって楕円離散対数問題を離散対数問題に帰着させることが可能となる。これら以外の楕円曲線が利用される楕円離散対数問題に対しては、現時点ではポーリック=ヘルマン（Pohlig-Hellman）のアルゴリズムやシャンクス（Shanks［1985］）によるBSGS（baby-step-giant-step）アルゴリズムが最も効率的となっている（表４参照）[7]。

表４　各楕円曲線に基づく離散対数問題のタイプと最も効率的な解法

楕円曲線のタイプ	解　法
拡大次数が小さい拡大体への埋込みが可能な楕円曲線	MOV帰着、FR帰着（USアルゴリズム）
超特異楕円曲線（トレース0）	MOV帰着、FR帰着（USアルゴリズム）
トレース2の楕円曲線	FR帰着（USアルゴリズム）
アノマラス曲線（トレース1）	SSSAアルゴリズム
上記以外の楕円曲線	ポーリック=ヘルマンのアルゴリズム、シャンクスのBSGSアルゴリズム

6　要素の個数pの有限体をF_pとする。このとき、F_p上で定義される楕円曲線上の点から構成される有限可換群の要素の個数は、$p-t+1$と表される（ハッセの定理）。ただし、tの値は$-2\sqrt{p} \leq t \leq 2\sqrt{p}$を満足し、各楕円曲線ごとに一意に定まる。このときのtの値がトレースと呼ばれている。

7　楕円離散対数問題の解法の詳細については、宇根・岡本［1999］を参照。

ハ. デジタル署名方式の分類

以上の分類によって、デジタル署名は12のカテゴリーに分類される。現在ISOの国際標準に規定されている方式を中心に、デジタル署名方式を分類すると表５のとおり。

表５　主要なデジタル署名方式の分類

分類		素因数分解問題に基づく方式	離散対数問題に基づく方式	楕円離散対数問題に基づく方式
署名添付型	確定型	•RSA署名（ナイーブな方式） •RSA署名（FDH-RSA署名） •RSA署名（PKCS #1 Ver. 2.0）		
	確率型	•フィアット=シャミア署名 •ESIGN（ISO/IEC 14888-3） •ギユー=キスケーター署名（ISO/IEC 14888-3） •RSA署名（PSS署名） •TSH-ESIGN	•エルガマル署名（ISO/IEC 14888-3） •シュノア署名（ISO/IEC 14888-3） •DSA（ISO/IEC 14888-3） •KCDSA •岡本=シュノア署名 •改良エルガマル署名	•EC-エルガマル署名（ISO/IEC 15946-2） •EC-シュノア署名 •EC-DSA（ISO/IEC 14888-3, ISO/IEC 15946-2） •EC-KCDSA（ISO/IEC 15946-2） •アグニュ=ムーリン=バンストン（ISO/IEC 14888-3） •EC-岡本=シュノア署名 •改良EC-エルガマル署名
メッセージ復元型	確定型	•RSA署名（ISO/IEC 9796） •RSA署名（ISO/IEC 9796-2）		
	確率型	•RSA署名（PSS-R署名）	•ナイバーグ=リュッペル署名 •ISO/IEC 9796-3 •阿部=岡本署名	•EC-ナイバーグ=リュッペル署名 •ISO/IEC 9796-3 •EC-阿部=岡本署名

― 汎業界向けの情報セキュリティ技術の標準化を担当するISO/IEC JTC1/SC27が策定したデジタル署名に関する国際標準（案）に規定されているデジタル署名を取り上げる。関連する国際標準（案）は以下のとおり。

①ISO/IEC 9796（Information technology－Security techniques－Digital signature scheme with giving message recovery）：メッセージ復元型のデジタル署名方式を規定。Part 2はMechanisms using a hash-function、Part 3はDiscrete logarithm based mechanismsであり、Part 2、3の標準化が開始された際に、ISO/IEC 9796本体のリバイス版としてISO/IEC 9796-1（Mechanisms using redundancy）が提案され、標準化が進められていた。なお、ISO/IEC 9796とISO/IEC 9796-1は同一内容である。

②ISO/IEC 14888（Information technology－Security techniques－Digital signature with appendix、Part 2はIdentity-based mechanisms、Part 3はCertificate-based mechanisms）：署名添付型のデジタル署名方法を規定。

③ISO/IEC WD 15946（Information technology－Security techniques－Cryptographic techniques based on elliptic curves、Part 2はDigital signatures）：楕円離散対数問題に基づくデジタル署名方式を規定。

（２）主要なデジタル署名方式のアルゴリズム

イ. 素因数分解問題に基づく方式

素因数分解問題に基づく方式の中から、主要な方式として以下の10の方式を説明する。まず、各デジタル署名方式の概要は以下の表６のとおり。

①署名添付・確定型署名方式

（A）RSA署名（ナイーブな方式）

RSA署名のナイーブな方式は、1978年にリベスト（Rivest）、シャミア（Shamir）、エーデルマン（Adleman）によって提案された（Rivest, Shamir and Adleman［1978］）。本方式は、公開鍵のみを利用する受動的攻撃（詳細は第４章（２）イ）に対して法nの素因数分解よりも効率的な攻撃法が提案されていないものの、攻撃者が任意のメッセージに対する署名を利用する適応的選択文書攻撃（詳細は第

表６　素因数分解問題に基づく主要なデジタル署名方式の概要

分類	方式名・標準規格名	提案者［提案年］	他の署名方式との関連	安全性証明の有無
署名添付・確定	RSA署名（ナイーブな方式）	リベスト、シャミア エーデルマン［1978］	RSA署名の中で最もシンプルな署名方式	なし
	RSA署名（FDH-RSA署名）	ベラーレ、ロガウェイ［1996］	RSA署名のナイーブな方式を改良	あり
	RSA署名（PKCS #1 Ver. 2.0）	RSA 社［1998］	RSA署名のナイーブな方式を改良	なし
署名添付・確率	フィアット=シャミア署名	フィアット、シャミア［1987］	フィアット=シャミア認証方式を署名方式に改良	あり
	ギユー=キスケーター署名	ギユー、キスケーター［1988］	――	あり
	ESIGN	岡本［1990］	――	なし
	RSA署名（PSS署名）	ベラーレ、ロガウェイ［1996］	RSA署名のナイーブな方式を改良	あり
	TSH-ESIGN	岡本、藤崎、森田［1998］	ESIGNの改良方式	あり
メッセージ復元・確定	RSA署名（ISO/IEC 9796）	――	RSA署名のナイーブな方式を改良	なし
	RSA署名（ISO/IEC 9796-2）	――	RSA署名のナイーブな方式を改良	なし
メッセージ復元・確率	RSA署名（PSS-R署名）	ベラーレ、ロガウェイ［1996］	RSA署名のナイーブな方式を改良	あり

４章（２）イ）に対しては効率的な攻撃法が提案されている[8]。このため、本署名方式を基にしたさまざまな改良方式が提案され、いくつかの方式が国際標準に規定されており、金融分野をはじめとする幅広い分野において実用化されている。署名生成・検証方法は以下のとおり。

〈ナイーブなRSA署名〉

【鍵生成】 ２つの大きな素数pとqを選び、これらの積 n=pq を計算。p−1とq−1の最小公倍数 L = LCM（p−1, q−1）を計算。最大公約数 GCD（e, L）= 1を満足する自然数eを選び、ed = 1（mod L）を満たすdを選択。

【署名生成鍵】 d

【署名検証鍵】（e, n）

【署名生成】 メッセージMのハッシュ値$H(M)$を生成した後（Hはハッシュ関数）、以下の計算によってMに対する署名Sを生成。

$$S = H(M)^d \bmod n$$

【署名検証】 以下の等式が成立するか否かを検証。

$$H(M) = S^e \bmod n$$

本方式では、署名生成鍵dによるべき乗剰余演算の対象となるデータ（以下、署名変換対象データと呼ぶ）の生成に、法nのサイズに比べて小さなハッシュ値を生成するハッシュ関数[9]が利用される。これに対して、ハッシュ値のサイズが法nのサイズと等しくなるフル・ドメイン・ハッシュ関数を利用する方式としてFDH-RSA署名が提案されている（詳細は次節）。

（B）RSA署名（FDH-RSA署名）

FDH-RSA（Full-Domain-Hash-RSA）署名は、RSA署名のナイーブな方式におけるハッシュ関数をフル・ドメイン・ハッシュ関数に置き換えた方式である。本署名方式は、1996年にベラーレ（Bellare）とロガウェイ（Rogaway）によって提案され、フル・ドメイン・ハッシュ関数にランダム・オラクル・モデル（詳細は第２章（２）ロ①）を仮定すると、適応的選択文書攻撃に対して存在的偽造（詳細は第４章（２）ロ）が不可能であることが証明されている（Bellare and Rogaway［1996］）。署名生成・検証方法は以下のとおり。

8 ナイーブなRSA署名に対する適応的選択文書攻撃の提案、これを受けての改良方式の提案、さらに、改良方式に対する攻撃法の提案等、RSA署名に関連する研究の経緯については第４章（２）を参照。

9 ハッシュ関数としては、通常、ハッシュ値のサイズが160 bitであるSHA-1、MD5、RIPEMD-160等の方式が利用される。

〈FDH-RSA署名〉
【鍵生成】ナイーブなデジタル署名方式と同一。
【署名生成】メッセージMのハッシュ値$H(M)$を生成する。ただし、ハッシュ関数Hは、以下の２つの条件を満足する。
① ハッシュ値のサイズが法nのサイズと同一となる。
② ランダム・オラクル・モデルの仮定に従う。
以下の計算によってMに対する署名Sを生成。

$$S = H(M)^d \bmod n$$

【署名検証】以下の等式が成立するか否かを検証。

$$H(M) = S^e \bmod n$$

ベラーレとロガウェイは、フル・ドメイン・ハッシュ関数を実現する方法として、SHA-1等による複数のハッシュ値を連結する方法等を提案している（Bellare and Rogaway［1993］）。

（C）RSA署名（PKCS #1 Ver. 2.0）

PKCS #1 Ver. 2.0[10]は、RSA方式を利用したデータ守秘方式およびデジタル署名方式の利用方法に関する技術仕様である（RSA Laboratories［1998］）。PKCS #1 Ver. 2.0は、Netscape 社が提唱する暗号通信、認証等のセキュリティ機能が付加されたHTTPプロトコルであるSSL等に採用されている。署名生成・検証方法は以下のとおり。

〈PKCS #1 Ver. 2.0〉
【鍵生成】ナイーブなデジタル署名方式と同一。
【署名生成】メッセージをM、ハッシュ関数を*Hash*（ハッシュ値のサイズをh byte）、SRを署名変換対象データとする。
＜1＞メッセージMのハッシュ値H=*Hash*(*M*)を計算する。
＜2＞ハッシュ関数のID情報とハッシュ値Hを含むデータT（t byte）を生成。
＜3＞パディング・データPSのサイズは（k − t − 3）byteであり、各bitの値はすべて “1” である。SRは以下のフォーマットとな

10 PKCS（Public-Key Cryptosystem Standard）は、RSA社が作成する公開鍵暗号の技術仕様であり、現在PKCS #1をはじめとして12の仕様が定められている。PKCSシリーズでは、RSA方式を利用する際のデータ変換方法、守秘、署名、鍵管理等についてルールを設けている。ただし、PKCSはRSA社が独自に制定する技術仕様であって、国際的な標準化検討委員会等によって定められた標準ではないことに注意が必要である。

り、サイズは公開鍵と同じk byteとなる[11]。また、“‖”はデータの結合を表す。

$SR = [\,00\ 01_{16} \| PS \| 00_{16} \| T\,] = [00\ 01_{16} \| FF...FF_{16} \| 00_{16} \| T]$

＜4＞署名生成者は、上記のSRを秘密鍵dで変換して署名Sを生成。

$$S = SR^d \bmod n$$

【署名検証】署名Sと署名検証鍵eを用いて$S^e \bmod n$を計算し、メッセージをM、ハッシュ関数*Hash*を用いてSRのフォーマット（上記＜3＞）が満足されていることを確認。

PKCS #1 Ver. 2.0の安全性については証明が示されていないものの、素因数分解よりも有効な攻撃法はこれまで提案されていない。

②署名添付・確率型署名方式

(A) フィアット=シャミア署名

フィアット=シャミア（Fiat-Shamir）署名は、認証者と被認証者との間でデータを３回交信してユーザー認証を行う（三交信認証方式と呼ばれる）フィアット=シャミア認証方式に基づく署名方式である（Fiat and Shamir［1987］）。本署名方式は、ハッシュ関数についてランダム・オラクル・モデルを仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている（Pointcheval and Stern［1996］）。署名生成・検証方法は以下のとおり。

〈フィアット=シャミア署名〉

【鍵生成】素数pとqを生成し、n = pqを計算。n−1以下の乱数kを生成し、k個のn−1以下の乱数s_i(i = 1,...,k）を生成（s_iはnを法とする乗法群の要素）。正整数s_iを用いて、$v_i = s_i^{-2} \bmod n$ (i = 1,...,k）を計算。

【署名生成鍵】p, q, s_i (i = 1,...,k)

【署名検証鍵】n, v_i (i = 1,...,k)

【署名生成】以下の計算を実行して、メッセージmに対する署名（y, e）を生成。ただし、*h*はハッシュ関数（k bit出力）。

- 乱数r（1 < r < n−1）を生成し、$u = r^2 \bmod n$を計算。
- $e = h(m \| u)$
- e_i (i = 1,...,k）：eをbit表現した時の各bitの値（各e_iは0または1の値となる）
- $y = r\prod_{j=1}^{k} s_j^{e_j} \bmod n$

【署名検証】$w = y^2\prod_{j=1}^{k} v_j^{e_j} \bmod n$、$e' = h(m \| w)$を計算し、e = e'が成立するか否かを確認

11 以下の表記では、「01_{16}」の「16」の添え字はbyte表示であることを表し、添え字がない場合はbit表示であることを表す。例えば「01_{16}」はbit表示では「0000 0001」となる。

(B) ギユー=キスケータ署名

ギユー=キスケータ（Guillou-Quisquater）署名は、三交信認証方式のギユー=キスケータ認証方式に基づく署名方式（Guillou and Quisquater［1988］）であり、ISO/IEC 14888-2に規定されている。本署名方式は、ハッシュ関数にランダム・オラクル・モデルを仮定し、RSA暗号関数の一方向性を仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている（Pointcheval and Stern［1996］）。署名生成・検証方法は以下のとおり。

〈ギユー=キスケータ署名〉

【鍵生成】 素数pとqを生成し、n = pqを計算。n−1以下の自然数で、(p−1)(q−1)と互いに素なeを選択。署名者のIDとしてnと互いに素な整数x（1<x<n）を選択。署名生成鍵としてxa^e mod n = 1を満足するaを設定。

【署名生成鍵】 a

【署名検証鍵】（n, e, x）

【署名生成】 以下の計算によって署名（s, l）を生成。ただし、メッセージをm, ハッシュ関数をh、"‖"をデータの結合を表すものとする。kは乱数。

- r = k^e mod n
- l = $h(m \| r)$
- s = ka^l mod n

【署名検証】 以下の計算を行い、l = l'が成立するか否かを確認。

- u = s^{e}x^l mod n
- l' = $h(m \| u)$

(C) ESIGN

ESIGNは、Okamoto［1990］によって提案された高速処理を特徴とするデジタル署名方式であり、素因数分解問題の困難性と合同多項不等式求解問題の困難性に基づいている。本署名方式はISO/IEC 14888-3に規定されている。署名生成・検証方法は以下のとおり。

〈ESIGN〉

【鍵生成】 大きな素数pとq（p>q）を選び、n = p^{2}qを計算。次に k>3となる自然数kを選択。

【署名生成鍵】（p, q）

【署名検証鍵】（k, n）（nはb bitとする）

【署名生成】 乱数x（ただし、0 <x < pq）を生成し、以下の計算によって署名sを生成（ただし、メッセージをM、ハッシュ関数をh）。

- Q= $(h(M) - (x^k \bmod n))/pq$
- y = w/(kx^{k-1}) mod p（wはQ以上の最小の整数）
- s = x + ypq

【署名検証】署名検証鍵kを用いて以下の不等式が成立するか否かを確認（ただし、Nは、(2b)/3以上の最小の整数）。

$$h(M) \leqq s^k \bmod n < h(M)+2^N$$

ESIGNのパラメーターについては、安全性の観点から、kとnは1,024 bit程度、pとqは340 bit程度が推奨されている。

（D）RSA署名（PSS署名）

PSS（Probabilistic Signature Scheme）署名は、RSA署名のナイーブな方式における署名変換対象データの生成方法に改良を加えた方式である。PSS署名はベラーレとロガウェイによって提案され、ランダム・オラクル・モデルとRSA暗号関数の一方向性（詳細は第５章（２）ロ②）を仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている（Bellare and Rogaway [1996]）。署名生成・検証方法は以下のとおり。

〈PSS署名〉

【鍵生成】署名生成鍵・検証鍵の生成方法はナイーブなRSA署名と同一。

【署名生成】メッセージをMとし、k_0 bitの乱数rを生成して以下の計算よってデジタル署名Sを生成。

- $w=h(M \| r)$
- $R=g_1(w) \oplus r$
- $X=(0 \| w \| R \| g_2(w))$
- $S = X^d \bmod n$

ただし、

- h：ランダム関数（出力値k_1 bit）
- g_1：ランダム関数（入力値k_1 bit、出力値k_0 bit）
- g_2：ランダム関数（入力値k_1 bit、出力値$k-k_0-k_1-1$ bit）

【署名検証】$X= S^e \bmod n$を計算し、$X = (b \| y \| z \| \alpha)$とする（b：1 bit、y：$k_1$ bit、z：k_0 bit、α：残りのbit）。$R'=g_1(y) \oplus z$ を計算し、次の等式がすべて成立するかを検証。

- $h(M \| R')= y$
- $g_2(y) = \alpha$
- $b = 0$

成立すれば署名が真正であることが確認される。そうでない場合には署名を受け付けない。

PSS署名の署名生成・検証に必要な計算量は、RSA署名のナイーブな方式に必要な計算量に１回のデータ圧縮変換と２回のデータ拡張変換が追加されるのみであり、ナイーブなRSA署名と同程度の実用性を有している。

（E）TSH-ESIGN

TSH-ESIGN（Trisection-Size-Hash ESIGN）は、ESIGNの改良方式であり（Okamoto, Fujisaki and Morita［1998］）、ランダム・オラクル・モデルとe乗根近似問題の困難性（詳細は第Ⅴ章２.（２）②を参照）を仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている。署名生成・検証手順は以下のとおり。

〈TSH-ESIGN〉

【鍵生成】大きな素数pとqを選び、$n = p^2q$を計算。ただし、pとqのサイズはいずれもkとする。次に、$e > 4$となる自然数eを選ぶ。

【署名生成鍵】（p, q）

【署名検証鍵】（e, n）（nは3k bit）

【署名生成】乱数x（$0 < x < pq$）を生成し、以下の計算で署名sを生成（Mをメッセージ、hをハッシュ関数とし、hのハッシュ値のサイズはk–1）。また、“‖”はデータの結合を表す。

- $Z = [0 \| h(M) \| 0^{2k}]$
- $\alpha = (Z - x^e) \bmod n$
- $Q = \alpha / pq$とし、Q以上の最小の整数をW_0に設定。
- $W_1 = pqW_0 - \alpha$（$W_1 \geqq 2^{2k-1}$の場合には、乱数xを選び直す）
- $t = W_0 / (ex^{e-1}) \bmod p$
- $s = (x + tpq) \bmod n$

【署名検証】署名検証鍵eを利用して以下の等式が成立するか否かを確認。ただし、$[Y]^k$はYの左k bit分のデータを表す。

$$[s^e \bmod n]^k = [0 \| h(M)]$$

TSH-ESIGNの署名生成・検証に必要なべき乗剰余演算[12]の回数を、PSS署名やEC-DSA（後述）と比較した結果は以下の表７のとおり。TSH-ESIGNは、他の主要なデジタル署名方式と比較して高い実用性を有している。

表７　TSH-ESIGNと他のデジタル署名方式との計算量の比較

デジタル署名方式	署名生成（べき乗剰余演算回数）	署名検証（べき乗剰余演算回数）
TSH-ESIGN	9	5
PSS署名	384	17
EC-DSA	41	48

（出典）岡本・藤崎［1999］

（注）TSH-ESIGN、PSS署名の署名検証鍵のサイズは1024 bit、EC-DSAの署名検証鍵のサイズを160 bitとして試算。

12 べき乗剰余演算は、ある数のべき乗を計算し、その計算結果に対して別の数で割った余りを計算するという演算（例えば、$x^e \bmod n$）。べき乗剰余演算は、他の算術演算（和、差、積、商）に比べて、一般的に処理時間が多く必要となる。

③メッセージ復元・確定型署名方式

（A）RSA署名（ISO/IEC 9796）

ISO/IEC 9796は、メッセージ復元型デジタル署名方式の国際標準であり、1991年に国際標準となっている。ISO/IEC 9796は、RSA署名のナイーブな方式における署名変換対象データの生成方法に改良を加えたものであり、署名変換対象データにメッセージを埋め込み、冗長性を持たせている点が特徴である（ISO/IEC［1991］）。署名生成・検証手順は以下のとおり。

〈ISO/IEC 9796〉

【鍵生成】RSA署名のナイーブな方式と同一。

【署名生成】メッセージMを用いて署名変換対象データ$U(M)$を生成し、署名生成鍵dを用いて署名Sを生成。

$$S = U(M)^{d} \bmod n$$

ただし、$U(M)$は、Mが8z bitの場合（zは正の偶数、8z bitでない場合はパディングを実施）、Mを4 bitのデータm_i（i=0,...,2z－1）に分割した後、３種類の換字変換s_1、s_2、sを用いて、

$$\begin{aligned} U(M) = [\, & s_1(m_{2z-1}) \| s_2(m_{2(z-1)}) \| m_{2z-1} \| m_{2(z-1)} \\ & \| s(m_{2z-3}) \| s(m_{2(z-2)}) \| m_{2z-3} \| m_{2(z-2)} \\ & \quad \cdots\cdots \\ & \| s(m_3) \quad \| s(m_2) \quad \| m_3 \quad \| m_2 \\ & \| s(m_1) \quad \| s(m_0) \quad \| m_1 \quad \| 0110 \,] \end{aligned}$$

となる（s_1はsの出力の左1 bitを1に固定する変換であり、s_2はsの出力の右1 bitを反転させる変換）。

【署名検証】 $U(M)' = S^{e} \bmod n$を計算し、$U(M)'$がISO/IEC 9796規定のフォーマットに従うことを確認した後、Mを復元。

ISO/IEC 9796は、法nのサイズが1024 bit、メッセージMのサイズが256 bitの場合、$U(M)$は法n のサイズと同じ1024 bitとなる。署名生成可能なメッセージのサイズの上限は法nによって変化し、法nが1024 bitの場合、署名生成可能なメッセージのサイズの上限は256 bitとなる。

ただし、これまでにいくつかの攻撃法（Coppersmith et al.［1999］等）が提案されており、1999年10月、SC27においてISO/IEC 9796を取り下げることが決定されている（詳細は第４章（３）ロ②）。

（B）RSA署名（ISO/IEC 9796-2）

ISO/IEC 9796-2は、ハッシュ関数を利用したメッセージ復元型のデジタル署名方式の国際標準であり、ISO/IEC 9796と同様、RSA署名のナイーブな方式における署名変換対象データを改良したものである。ISO/IEC 9796-2では、署名変換対象データ$U(M)$に署名対象のメッセージMやメッセージのハッシュ値が含まれて

おり、法nのサイズが1024 bitの場合、署名から最大848 bitのメッセージを復元できる（ISO/IEC［1997］）[13]。

〈ISO/IEC 9796-2〉

【鍵生成】RSA署名のナイーブな方式と同一。

【署名生成】メッセージMを用いて署名変換対象データ$U(M)$を生成し（Uは署名変換対象データの生成関数、図４参照）、以下の計算によってMに対する署名Sを生成。

$$S = U(M)^d \bmod n$$

【署名検証】$U(M)' = S^e \bmod n$を計算し、$U(M)'$がISO/IEC 9796-2規定のフォーマットに従うことを確認した後、メッセージMを復元。

図４　U（M）のフォーマット（例 nとM: 1024 bit, ハッシュ値: 160 bit）

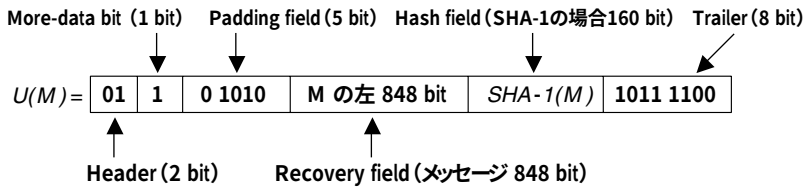


図４にあるように、法nおよびメッセージが1024 bit、160 bitのハッシュ値を持つハッシュ関数（例えば、SHA-1）を利用する場合、$U(M)$は６つの部分から構成される。各部分の内容は以下のとおり。

（i）Header：$U(M)$の左から2 bit分のデータであり、常に“01”。

（ii）More-data bit：メッセージ全体が復元できる場合は“0”、そうでない場合は“1”。

（iii）Padding field：$U(M)$を1,024 bitにするためのパディングデータ“01010”。

（iv）Recovery field：メッセージのうち復元可能なデータ（左から848 bit分）。

（v）Hash field：メッセージのハッシュ値 *SHA-1*(M)（サイズは160 bit）。

（vi）Trailer：ハッシュ関数の属性を表すデータ。*SHA-1*の場合は“1011 1100”。

ISO/IEC 9796-2の安全性については、これまでに攻撃法（Coron et al.［1999］）が提案されており（詳細は第４章（３）ロ③を参照）、現在SC27において今後の対応方針について検討が行われている。

13 これ以上のサイズのメッセージに対する署名を生成する場合には、メッセージのうち左から848 bit分のデータのみが復元可能となる。

④メッセージ復元・確率型署名方式 ― RSA署名（PSS-R署名）

PSS-R署名は、PSS署名のアルゴリズムにおける署名変換対象データを、署名からメッセージが復元可能なように改良した方式である。本署名方式は、ベラーレとロガウェイによって提案され、PSS署名と同様の条件のもとで、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている（Bellare and Rogaway［1996］）。署名生成・検証方法は以下のとおり。

〈PSS-R署名〉

【鍵生成】ナイーブなRSA署名と同一。

【署名生成】メッセージをMとし、$k_0$bitの乱数rを生成して以下の計算によってデジタル署名Sを生成。

- $w = h(M \| r)$
- $R = g_1(w) \oplus r$
- $X = (0 \| w \| R \| g_2(w) \oplus M)$
- $S = X^d \bmod n$

ただし、

- h：ランダム関数（出力値k_1 bit）
- g_1：ランダム関数（入力値k_1 bit、出力値k_0 bit）
- g_2：ランダム関数（入力値k_1 bit、出力値$k-k_0-k_1-1$ bit）

【署名検証】$X = S^e \bmod n$を計算し、$X = (b \| y \| z \| \alpha)$とする（b：1 bit、y：$k_1$ bit、z：k_0 bit、α：残りのbit）。$R' = g_1(y) \oplus z$を計算し、次の等式がすべて成立するかを検証。

- $\alpha \oplus g_2(y) = M'$
- $h(M' \| R') = y$
- $b = 0$

成立すれば署名が真正であることが確認される。そうでない場合には署名を受け付けない。M'がメッセージとして復元される。

PSS-R署名がPSS署名と異なる点は、署名変換対象データXのうち、$g_2(w)$の代わりに$g_2(w) \oplus M$が利用されている点である。また、署名生成および検証に必要となる計算量については、PSS署名に１回の排他的論理和演算が追加されるだけであるが、メッセージのサイズを大きくすると署名変換対象データXが拡大し、べき乗剰余演算の計算量が増加する。このため、実際の処理速度はメッセージのサイズにも依存する。

ロ. 離散対数問題に基づく方式

離散対数問題に基づく方式の中から、主要な方式として以下の方式を説明する。まず、各デジタル署名方式の概要は表８のとおり。

表８　離散対数問題に基づく主要なデジタル署名方式の概要

分類	方式名・標準規格名	提案者［提案年］	他の署名方式との関連	安全性証明の有無
署名添付・確率型	エルガマル署名	エルガマル［1985］	―――	なし
	シュノア署名	シュノア［1990］	エルガマル署名の改良方式	あり
	DSA	NIST［1991］	エルガマル署名および シュノア署名の改良方式	なし
	KCDSA	―――	DSAの改良方式	あり
	岡本=シュノア 署名	岡本［1993］	シュノア署名の改良方式	あり
	改良エルガマル 署名	ポワンシュバル、スターン ［1996］	エルガマル署名の改良方式	あり
メッセージ 復元・確率型	ナイバーグ=リュッペル 署名	ナイバーグ、リュッペル ［1993］	DSAの改良方式	なし
	ISO/IEC 9796-3	―――	DSAの改良方式	なし
	阿部=岡本署名	阿部、岡本［1999］	―――	あり

①署名添付・確率型署名方式

（A）エルガマル署名

エルガマル（ElGamal）署名は、ElGamal［1985］によって提案された署名方式であり、ISO/IEC 14888-3に規定されている。エルガマル署名には、（i）署名生成のつど秘密の乱数を生成する必要がある、（ii）署名が法のサイズの２倍となる、という短所が存在する。署名生成・検証手順は以下のとおり。

〈エルガマル署名〉

【鍵生成】大きな素数pと、pを法とする乗法群の要素xを選び、$y = \alpha^x \bmod p$（αはpを法とする乗法群の原始根）を計算。

【署名生成鍵】 x

【署名検証鍵】（y, p,α）

【署名生成】乱数kを生成し、以下の計算によって署名（r, t）を生成（メッセージをM、ハッシュ関数をhとする）。

・$r = \alpha^k \bmod p$

・$t = (h(M) - xr)/k \bmod (p-1)$

【署名検証】署名検証鍵yを用いて以下の等式が成立するか否かを確認。

$$\alpha^{h(M)} = y^r r^t \bmod p$$

エルガマル署名を安全に利用するためには、毎回異なる乱数kを利用する必要がある。また、公開鍵pとαの値がある一定の条件を満足する場合には、秘密鍵を知らなくても容易にデジタル署名の偽造が可能となることが示されており（Bleichenbacher［1996］）、他者が生成した公開鍵pとαを利用するべきではないといわれている。

（B）シュノア署名

シュノア（Schnorr）署名は、Schnorr［1990］によって提案された方式であり、エルガマル署名における「署名のサイズが法pの２倍になる」という短所を改善し、処理速度を向上させた方式である。本署名方式では、ハッシュ関数がランダム・オラクル・モデルに従うことを仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている（Pointcheval and Stern［1996］）。また、本署名方式はISO/IEC 14888-3に規定されている。署名生成・検証方法は以下のとおり。

〈シュノア署名〉

【鍵生成】素数pと、p−1の素因数qを選択。pを法とする乗法群の要素xを選び、$y = g^{(-x)} \bmod p$（gは、pを法とする乗法群の原始根であり、位数[14]がqとなるもの）を計算。

【署名生成鍵】 x

【署名検証鍵】（y, g, p, q）

【署名生成】乱数kを生成し、以下の計算によって署名（e, s）を生成（Mはメッセージ、*h*はハッシュ関数）。

- $r = g^k \bmod p$
- $e = h(M, r)$
- $s = (xe + k) \bmod q$

【署名検証】署名検証鍵yを用いて以下の計算を実行。

- $v = g^s y^e \bmod p$
- $e' = h(M, v)$

生成したe'を用いて、e = e'が成立するか否かを確認。

シュノア署名は、署名生成に利用するgの値として、位数がp−1の素因数qとなるような値を選択する。この結果、署名（e, s）の一部sのサイズをpのサイズからqのサイズまで短縮することが可能となり、署名サイズの縮小によって署名生成・検証による計算量も縮小する。

（C）DSA

DSAはエルガマル署名の改良方式であり、NIST[15]［1991］によって提案され、1994年に米国連邦政府のデジタル署名標準（FIPS 186）となっているほか、

14 位数：自然数に0を加えた集合 {0, 1, 2, ...}をZと定義し、各Zの要素に対して$g^x \bmod p$を計算する。計算される$g^x \bmod p$の値の集合をyとするとき、yの集合の要素の数が位数と呼ばれる。例えば、gとpがそれぞれ3、7の場合、$3^x \bmod 7$は{1, 2, 3, 4, 5, 6}のいずれかの値となることから、法7における乗法群の元3の位数は6となる。

15 NIST（National Institute for Standards and Technology）：米国商務省の下部組織で、科学技術全般に関する標準を策定する役割を担っているほか、情報通信の分野では、1987年に成立したComputer Security Act により、米国政府内部における情報通信規格であるFIPS（Federal Information Processing Standard）を制定する権限を有している。

ISO/IEC 14888-3に規定されている。DSAは、シュノア署名と同様に、「法p、p–1の素因数qの下で、位数がqとなるようなgを用いて署名を生成する」という方法により、エルガマル署名の署名長を2pから2qに短縮することを可能にした。署名生成・検証方法は以下のとおり。

〈DSA〉

【鍵生成】素数pと、p–1の素因数qを選ぶ。pを法とする乗法群の要素xを選び、$y = g^x \bmod p$（gはpを法とする乗法群の要素であり、位数がqとなるもの）を計算。

【署名生成鍵】 x

【署名検証鍵】（y, g, p, q）

【署名生成】乱数kを生成し、以下の計算によって署名（r, t）を生成（ただし、Mはメッセージ、*h*はハッシュ関数）。

- $r = (g^k \bmod p) \bmod q$
- $t = (h(M) + xr)/k \bmod q$

【署名検証】署名検証鍵yを用いて以下の等式が成立するか否かを検証する。

$$r = (g^{h(M)/t} y^{r/t} \bmod p) \bmod q$$

DSAは、署名生成のアルゴリズムを若干変更し、利用するハッシュ関数にランダム・オラクル・モデルの仮定をおくことによって、適応的選択文書攻撃に対していかなるメッセージに対する署名の偽造も不可能となることを証明可能である（Pointcheval and Vaudenay［1996］）。ポワンシュバル（Pointcheval）とバウデニー（Vaudenay）は、こうしたDSAの改良方式として２種類のアルゴリズムを提案している。第一の改良方式は、ハッシュ関数*h1*と*h2*を利用し、署名生成式として$r = h2(g^k \bmod p)$、$t = (h1(M) + xr)/k \bmod q$、署名検証式として$r = h2(g^{h1(M)/t} y^{r/t} \bmod p)$を利用するものである。この時、ハッシュ関数*h1*と*h2*がランダム・オラクル・モデルの仮定に従うならば、上記証明が可能となる。また、第二の改良方式は、署名生成式として$r = (g^k \bmod p) \bmod q$、$t = (h(M, r) + xr)/k \bmod q$、署名検証式として$r = (g^{h(M, r)/t} y^{r/t} \bmod p) \bmod q$を利用するものである。この時、ハッシュ関数*h*がランダム・オラクル・モデルに従うならば、上記証明が可能となる。

（D）KCDSA

KCDSA（Korean Certificate-based Digital Signature Algorithm）は、DSAの改良方式であり、韓国のデジタル署名標準KICS（Korean Information and Communication Standard）となっている（KCDSA Task Force Team［1998］）。本署名方式を楕円離散対数問題に基づく方式に変換したEC-KCDSAが、現在国際標準案ISO/IEC WD 15946-2に記載されている。本署名方式は、ハッシュ関数がランダム・オラクル・モデルの仮定を満足すると仮定した場合、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている。署名生成・検証方法は以下のとおり。

〈KCDSA〉

【鍵生成】素数pと、p–1の素因数qを選ぶ。pを法とする乗法群の要素xを選び、$y = g^{(-x)} \bmod p$（gはpを法とする乗法群の要素であり、位数がqとなるもの）を計算。また、署名者のIDや（y, p, q, g）等をハッシュ化したデータzを生成[16]。

【署名生成鍵】 x

【署名検証鍵】（p, q, g, y, z）

【署名生成】乱数kを生成し、以下の計算によってデジタル署名（r, s）を生成（*h*はハッシュ関数、mはメッセージ）。

・$r = h(g^k \bmod p)$

・$s = x(k - r \oplus h(z \| m)) \bmod q$

【署名検証】署名者IDや署名検証鍵等からzを生成したうえで、以下の計算を実行。

・$e' = r \oplus h(z \| m) \bmod q$

・$r' = h((y^s)(g^{e'}) \bmod p)$

最後に、r＝r'が成立しているか否かを確認。

本署名方式は、署名生成において署名者のID情報等が利用されている点が特徴である。また、署名の生成・検証に必要となる計算量はDSAと同程度となっている。

（E）岡本=シュノア署名

岡本=シュノア（Okamoto-Schnorr）署名は、Okamoto［1993］によって提案された方式であり、利用されるハッシュ関数が無相関一方向性ハッシュ関数（第５章（２）を参照）であることを仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている。署名生成・検証方法は以下のとおり。

〈岡本=シュノア署名〉

【鍵生成】整数k（サイズはt）を選択し、以下の手順によって鍵生成を実行。

・p–1がqで割り切れるような素数pとqを選択。

・pを法とする乗法群で、位数がqとなる素数g1とg2を選択。

・q未満の自然数s1とs2をランダムに選択。

・$v = (g1)^{(-s1)} (g2)^{(-s2)} \bmod p$を計算。

・出力値のサイズがtとなるハッシュ関数*H*（無相関一方向性ハッシュ関数）を選択。

【署名生成鍵】（s1, s2）

16 zの元となるデータとして、KCDSAの署名検証鍵に用いられる公開鍵認証書が挙げられている（KCDSA Task Force Team［1998］）。

【署名検証鍵】（p, q, g1, g2, t, v, H, k）

【署名生成】メッセージmに対し、以下の計算から生成される（e, y1, y2）が署名となる。

・$x = (g1)^{(r1)} (g2)^{(r2)} \bmod p$（r1とr2はq未満の乱数）

・$e = H(x, m)$、$y1 = (r1 + e \cdot s1) \bmod q$、$y2 = (r2 + e \cdot s2) \bmod q$

【署名検証】署名検証鍵を利用して、以下の手順で署名を検証。

・$x = (g1)^{(y1)} (g2)^{(y2)} v^e \bmod p$を計算。

・$e = H(x, m)$が成立するか否かを確認。

成立すれば正当な署名と判断するが、成立しなければ不当な署名として受け付けない。

岡本=シュノア署名の署名生成・検証に必要となる計算量は、シュノア署名よりもやや多くなる。公開鍵のpとqのサイズをそれぞれ1,024 bit、160 bitとして、べき乗剰余演算の回数を比較すると以下の表９のとおり。

表９　岡本=シュノア署名とシュノア署名との計算量の比較

デジタル署名方式	署名生成（べき乗剰余演算回数）	署名検証（べき乗剰余演算回数）
岡本=シュノア署名	280	300
シュノア署名	240	280

（出典）岡本・藤崎［1999］

（F）改良エルガマル署名

改良エルガマル署名は、Pointcheval and Stern［1996］によって提案された方式であり、エルガマル署名で利用されているハッシュ関数に一部変更を加えた方式である。安全性については、ランダム・オラクル・モデルを仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている。署名生成・検証方法は以下のとおり。

〈改良ElGamal署名〉

【鍵生成】署名生成鍵・検証鍵はエルガマル署名と同一。

【署名生成】ランダム・オラクル・モデルが成立するハッシュ関数hを用意し、乱数kを生成。そのうえで、署名生成鍵xを用いた以下の計算によってデジタル署名（r, t）を生成（Mはメッセージ）。

・$r = \alpha^k \bmod p$

・$t = (h(M, r) - xr)/k \bmod (p-1)$

【署名検証】署名検証鍵yを利用して、以下の等式が成立するか否かを検証。

$$\alpha^{h(M,r)} = y^r r^t \bmod p$$

改良エルガマル署名は、エルガマル署名で利用されているハッシュ関数$h(M)$を$h(M, r)$に変更するという比較的マイナーな改良によって構成されていることから、改良エルガマル署名の処理速度はエルガマル署名とほぼ同程度とみられている。

②メッセージ復元・確率型署名方式

(A) ナイバーグ=リュッペル署名

ナイバーグ=リュッペル（Nyberg-Rueppel）署名は、DSAをベースとしたメッセージ復元・確率型署名方式であり、Nyberg and Rueppel［1993］によって提案された。署名生成・検証方法は以下のとおり。

〈ナイバーグ=リュッペル署名〉

【鍵生成】署名生成・検証鍵はDSAと同一。

【署名生成】メッセージmに対してm' = 16m+ 6を計算。乱数k（$1 < k < q-1$）を生成し、$r = g^{(-k)} \bmod p$を計算。署名生成鍵xを用いた以下の計算によって署名（e, s）を生成。

- $e = m'r \bmod p$、$s = (xe+k) \bmod q$

【署名検証】署名検証鍵yを用いて、以下の手順で署名を検証。

- $0 < e < p$かつ$0 < s < q$を確認。
- $v = g^{s}y^{(-e)} \bmod p$
- $m' = ve \bmod p$

計算したm'が $m' \bmod 16 = 6$を満足しているか否かを確認し、満足する場合、$m = (m' - 6)/16$を計算してmを復元。

(B) ISO/IEC 9796-3

ISO/IEC 9796-3は、メッセージ復元型のデジタル署名方式の国際標準案であり、現在SC27において標準化が進められている。本国際標準案は、当初ISO/IEC 9796-4として標準化が検討されていたが、RSA署名をベースとするデジタル署名方式として提案されていたISO/IEC 9796-3が有効な攻撃法の発表（Misarsky［1997］）によって廃止となり、ISO/IEC 9796-4がISO/IEC 9796-3に変更となった。署名生成・検証方法は以下のとおり。

〈ISO/IEC 9796-3〉

【鍵生成】署名生成・検証鍵はDSAと同一。

【署名生成】メッセージMに対して$U(M)$を計算（$U(M)$のフォーマットについては図５参照）。乱数k（$1 < k < q-1$）を生成し、署名生成鍵xを用いた以下の計算によって署名（R, S）を生成。

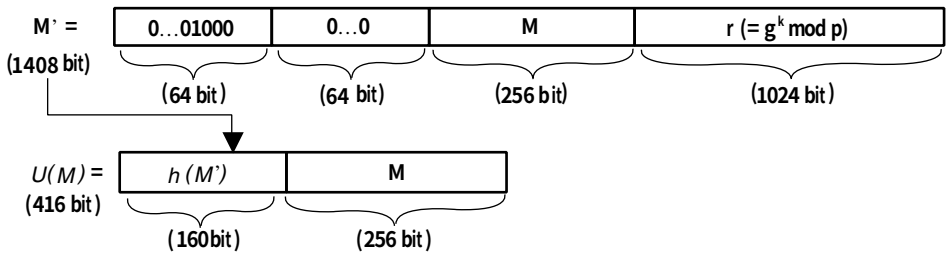
- $r = g^{(-k)} \bmod p$

・$R = (r + U(M)) \bmod q$

・$S = (k - xR) \bmod q$

【署名検証】 $r' = g^s y^r \bmod p$を生成し、$r = r'$が成立するか否かを確認。成立する場合には、$U(M) = (R - r) \bmod q$を計算し、$U(M)$のフォーマットを手掛かりにMを復元。

図５　U（M）のデータフォーマット（例 p: 1024 bit, M: 256 bit）



（C）阿部=岡本署名

阿部=岡本署名は、Abe and Okamoto［1999］によって提案された方式であり、アルゴリズムに利用されているランダム関数にランダムオラクルモデルを仮定すると、適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されている。署名生成・検証方法は以下のとおり。

〈阿部=岡本署名〉

【鍵生成】 素数p（サイズはk1 bit）と、p－1の素因数q（サイズはk2 bit）を選ぶ。qを法とする乗法群の要素xとwをランダムに選び、$y = g^{(-x)} \bmod p$（gはpを法とする乗法群の要素であり、位数がqとなる数）を計算。以下の４つのランダム関数を用意。

・F_1：入力k3 bit、出力k4 bit

・F_2：入力k4 bit、出力k3 bit

・H：入力（k3 + k4）bit、出力k1 bit

・G：入力k1 bit、出力（k3 + k4）bit

F_1, F_2, H, Gはランダム・オラクル・モデルの仮定を満たすとする。

【署名生成鍵】 x

【署名検証鍵】（y, g, p, q）

【署名生成】 メッセージm（サイズはk3 bit）に対して以下の計算から署名（r, z）を生成。

・$m' = F_1(m) \| (F_2(F_1(m)) \oplus m)$

・$r = G(g^w \bmod p) \oplus m'$

・$c = H(r)$

・$z = (w + cx) \bmod p$

【署名検証】 以下の式により、m'、mを計算する（$[m']_{k3}$ はm'の右からk3 bit分のデータ、$[m']^{k4}$ はm'の左からk4 bit分のデータを指す）。

・$m' = r \oplus G(g^z y^c \bmod p)$

・$m = [m']_{k3} \oplus F_2([m']^{k4})$

以下の等式が成立することを確認し、成立すればmを出力するが、成立しなければ不正な署名として受付けない。

$$[m']^{k4} = F_1(m)$$

阿部=岡本署名の署名生成・検証に必要な計算量はシュノア署名やナイバーグ=リュッペル署名と同程度となっているほか、楕円離散対数問題を利用したEC-阿部=岡本署名の場合もEC-ナイバーグ=リュッペル署名等と同程度となっており、実用性が高い方式である（Abe and Okamoto［1999］）。

ハ. 楕円離散対数問題に基づく方式 － EC-DSA

離散対数問題に基づく方式は、楕円離散対数問題に基づく方式で利用することができる。例えば、エルガマル署名、シュノア署名、DSAをはじめとする離散対数問題に基づく方式には、対応する楕円離散対数問題に基づく方式としてEC-エルガマル署名、EC-シュノア署名、EC-DSA[17]等が提案されている。DSAとEC-DSAを例にとり、離散対数問題に基づく方式と楕円離散対数問題に基づく方式の対応関係は図６のとおり。

17 EC-DSAは、ISO/IEC 14888-2およびISO/IEC WD 15946-2においてデジタル署名方式のひとつとして記載されているほか、現在米国の金融分野におけるデジタル署名の標準案ANSI X9.62として標準化が完了している。

図６　離散対数問題に基づく方式と楕円離散対数問題に基づく方式の対応関係

DSA（離散対数問題）		EC-DSA（楕円離散対数問題）
1. p−1**が大きな素因数qを持つ素数pを選択。** 2. p**を法とし、乗法を演算とする乗法群を定義。** 3. 定義した乗法群において**位数qとなる元**gを選択。	**パラメーターや利用する群等の選択** ⇔	1. **素数**pを選択。 2. p**を法とする有限素体**F_p上の点(x, y)を**含む楕円曲線**E(例えば、$y^2 = x^3 + cx + d$)を選択。E**上の点の集合において、加法を演算とする有限可換群を定義。** 3. 定義した有限可換群において、位数qが大きな素数となる元 $g=(x_g, y_g)$を選択。
4. 法pにおける乗法群の要素aを選び、署名生成・検証鍵を生成。 ・署名生成鍵 a ・署名検証鍵(b, g, p, q) ただし、$b = g^a \bmod p$	**鍵生成** ⇔	4. 正の整数aを選び、署名生成・検証鍵を生成。 ・署名生成鍵 a ・署名検証鍵(b, g, p, q, E) ただし、$b = a*g = (x_b, y_b)$ (*は、定義された加法演算を表し、a*gは点gをa回加える演算を表す)
5. 乱数kを生成。 6. ハッシュ関数hを利用して、メッセージMに対する署名(r, s)を生成。ただし、 ・$r = (g^k \bmod p) \bmod q$ ・$s = (h(M)+ar)/k \bmod q$	**署名生成** ⇔	5. 乱数kを生成。 6. ハッシュ関数hを利用して、メッセージMに対する署名(r, s)を生成。ただし、 ・$r = x\ (k*g) \bmod q$ (x(k*G)はk*Gのx座標の値) ・$s = (h(M) + ar)/k \bmod q$
7. 署名検証鍵を利用して、以下の検証式が成立することを確認。 $r = (g^{h(M)/s} b^{r/s} \bmod p) \bmod q$	**署名検証** ⇔	7. 署名検証鍵を利用して、以下の検証式が成立することを確認。 $r = x((h(M)/s)*g + (r/s)*b) \bmod q$

両者の方式の基本的な違いは、署名生成・検証に利用される集合および演算である。具体的には、以下の２点である。

①離散対数問題に基づく方式における「乗法群上の要素」を、楕円離散対数問題に基づく方式における「楕円曲線によって定義される有限可換群上の要素」に対応させる。

②離散対数問題に基づく方式における「乗法群上での乗法」を、楕円離散対数問題に基づく方式における「有限可換群上での加法」に対応させる。

この結果、乗法群におけるべき乗の演算（例えば、$b = g^a \bmod p$）は、有限可換群における積の演算（例えば、$b = a*g$）に対応する。

なお、乗法群における要素のr乗の計算は、楕円曲線上の有限可換群における要素のr倍の計算と同程度の計算量によって実現される（Koblitz［1997］）。

４．デジタル署名方式の安全性の評価

（１）デジタル署名方式の安全性評価に関する３種類の研究

デジタル署名方式の安全性評価に関する研究は、大きく分けると以下の３種類に分類することができる。これらの研究はデジタル署名方式の安全性を評価するうえで、相互依存関係にある。

①デジタル署名方式の安全性が依拠している数学問題の困難性に関する研究
②デジタル署名方式のアルゴリズム（鍵生成や署名生成・検証方法）の安全性に関する研究
③デジタル署名方式の実装環境が、そのデジタル署名方式の安全性に与える影響に関する研究

現在主要なデジタル署名方式とされている方式は、素因数分解問題、離散対数問題、楕円離散対数問題のいずれかの数学問題の困難性に依拠している。これら３種類の数学問題は、これまで上記①に関する研究の蓄積により、現時点でデジタル署名方式に利用可能であると多くの暗号研究者から評価されている。このため、主要な署名方式の安全性について、安全性の根拠となっている数学問題の困難性の観点からは、現時点では問題は少ないとみられている[18,19]。

上記①の研究に基づき、「数学問題の困難性の観点からは安全性上問題がない」と評価されたとしてもそれだけでは十分とはいえず、デジタル署名方式のアルゴリズムに欠陥がないか否かについて研究・評価を行う必要がある。このような研究は上記②に該当する。RSA署名等、前章において示された主要なデジタル署名方式は、これまでに多くの暗号学者によってアルゴリズムの安全性に関する研究が行われ、現時点において致命的な欠陥が示されていない[20]。あるデジタル署名方式が安全性の観点から実用的とみなされるためには、少なくとも上記①および②の研究におい

18 ただし、今後の研究の進展によっては、これらの数学問題に対して効率的な解法が提案される可能性もある。

19 数学の問題を解くために必要となる計算量は不変でも、コンピューターのコスト・パフォーマンスの向上等によって計算を実行する時間や費用が低下し、デジタル署名の安全性が低下する可能性がある。このため、現在利用されている数学問題におけるパラメーター（例えば、素因数分解問題における合成数nのサイズ）や鍵長等の設定が適切か否かについても、最新の研究動向を踏まえて評価する必要がある。こうした評価の必要性を強く印象付ける最近の事例として、512 bit 合成数の素因数分解の成功が挙げられる。RSA社は、２つの素数の積の素因数分解がどれだけ短時間で成功するかを競うコンテストを実施しており、1999年８月、512 bitの合成数が数体ふるい法によって約５か月で素因数分解されたことを発表した。この結果、鍵長が512 bitのRSA署名方式を利用しているアプリケーションにおいては鍵長の拡大が必要とされている。

20 新しいデジタル署名として提案された方式の中には、アルゴリズムにおいて安全性上の欠陥が発見された方式がいくつも存在している。そうした署名方式のひとつとして、ナップザック問題と呼ばれる数学問題を利用した公開鍵暗号方式マークル=ヘルマン（Merkle-Hellman暗号（デジタル署名としても利用可能なアルゴリズム）が挙げられる。ナップザック問題自体はその困難性に関してデジタル署名方式に利用す

て致命的な欠陥が見つかっていないことが必要である。

上記①および②に関する研究によって、「安全性が依拠している数学問題の困難性や署名方式のアルゴリズムの観点からは、安全性上問題がない」と評価された署名方式に対しては、上記③の観点からも安全性の評価を行うことが必要である。③の研究では、具体的には、「デジタル署名が実装される環境に基づいて、攻撃者が利用可能な情報や攻撃法を分析し、どのようなメッセージに対する署名の偽造が可能か」について評価が行われる。

従来の安全性評価の方法は、それまでに提案された攻撃法や分析手法のみを前提とするものが中心であった。このため、新しい攻撃法に対する安全性について十分な検討を行うことが不可能であり、既存の攻撃法や分析手法に基づいて安全であると評価されていたデジタル署名方式に対して、有効な攻撃法が提案される可能性を排除できなかった[21]。近年におけるデジタル署名の実装環境の多様化が進む中で、実装環境次第で新しい攻撃法が有効になるリスクが高まっている。こうした状況下、最近のデジタル署名方式に関する安全性評価の分野では上記③に関する研究の動向が注目されていることから、本稿では、上記③に関する安全性評価の研究に焦点を当てる。

（2）デジタル署名方式に対する攻撃のタイプと達成度

デジタル署名の実装環境がその署名方式の安全性に及ぼす影響について検討を行うためには、まず、①攻撃者が利用可能な情報（攻撃のタイプ）と②署名偽造の程度（攻撃の達成度）を分類する必要がある。

イ. 攻撃のタイプ

デジタル署名方式への攻撃は、大きく２種類（受動的攻撃と能動的攻撃）に分類することができるほか、能動的攻撃はさらに２種類（一般選択文書攻撃と適応的選択文書攻撃）に分類される（表10参照）。

表10　攻撃のタイプの分類

攻撃のタイプ		内容
受動的攻撃		署名検証鍵のみを利用して署名を偽造するという攻撃。
能動的攻撃	一般選択文書攻撃	攻撃者があらかじめ指定した文書に対して真の署名者に署名させ、入手した署名等の情報を用いて別の文書の署名を偽造する攻撃。
	適応的選択文書攻撃	攻撃者が任意に選んだ文書に対して真の署名者に署名させ、入手した署名等を用いて別の文書の署名を偽造する攻撃。

る点で問題がなかったものの、マークル=ヘルマン暗号における鍵生成のアルゴリズムに欠陥があり、公開鍵を生成するアルゴリズムの特性を利用することによってナップザック問題を効率的に解くアルゴリズムが発見されている。

21 こうした研究の代表的な例として、RSA署名に基づく国際標準ISO/IEC 9796に対する攻撃法の提案が挙げられる。詳細については、本章第３節を参照。

受動的攻撃は、攻撃者が公開情報である署名検証鍵のみを利用して行う攻撃であり、容易に実行可能である。このため、実用的なデジタル署名方式は受動的攻撃に対して十分な安全性を確保することが必要である。一方、能動的攻撃は、攻撃者が自分にとって都合の良い文書に対する署名を利用して行う攻撃であり、実行可能性は受動的攻撃よりも低い。とくに、適応的選択暗号文攻撃では、いったん入手した署名とそれに対応する文書を分析した後、その分析結果を踏まえて新たに文書を選択し、その文書に対する署名を入手することができる。このように、適応的選択文書攻撃が最も強力な攻撃であり、本攻撃に対して十分な安全性を確保することが望ましい。

ロ. 攻撃の達成度

デジタル署名方式に対する攻撃の達成度は、以下の２種類（一般的偽造と存在的偽造）に分類できる（表11参照）。

表11　攻撃の達成度の分類

達成度	内　容
一般的偽造	任意の文書に対してデジタル署名を偽造できる。
存在的偽造	ある特定の文書に対してデジタル署名を偽造できる。

一般的偽造はどのような文書に対する署名も偽造可能であるというものであり、実現可能性の高い攻撃によって一般的偽造が可能になるとすれば、そのデジタル署名方式は実用的とはいえない。また、存在的偽造は、すべての文書に対して署名の偽造が可能となるわけではないが、ある特定の文書に対しては署名の偽造が可能であるというものである。このため、存在的偽造は、一般的偽造よりも攻撃の達成度が低い。

デジタル署名方式の安全性の観点からは、「いかなる文書に対する署名も偽造が不可能である」、すなわち「存在的偽造が不可能である」署名方式が望ましい。前節で説明したように、最も強力な攻撃は適応的選択文書攻撃であることから、「適応的選択文書攻撃に対して存在的偽造が不可能」なデジタル署名方式が最も望ましい。

（３）代表的なデジタル署名方式に対する攻撃と対策－RSA署名の場合

最近では、本章第１節における③の研究、すなわち、デジタル署名方式の実装環境が署名方式の安全性に及ぼす影響に関する研究が注目を集めている。さまざまな研究成果が発表されている中で、暗号研究者だけではなく、標準化に携わっている実務家の間で注目されているのが、RSA署名をベースとするデジタル署名方式の国際標準ISO/IEC 9796とISO/IEC 9796-2に対する攻撃法の提案である。以下では、

RSA署名に関する安全性の評価結果について、これらの国際標準に対する攻撃法を中心に説明する。

イ. 受動的攻撃に対する安全性

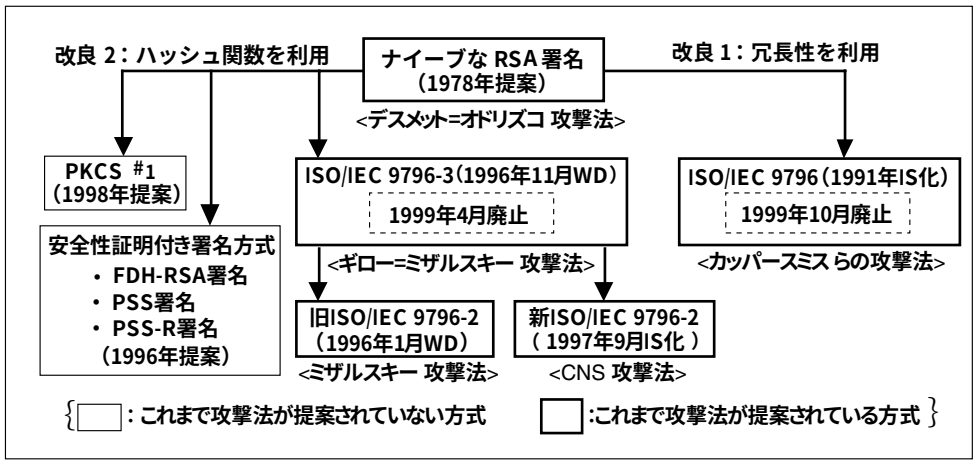
受動的攻撃が可能な環境を前提にする場合、現時点では、RSA署名に対する最も有効な攻撃法は法nを素因数分解するというものである。このため、受動的攻撃に対するRSA署名の安全性は、法nの素因数分解に必要な計算量やコンピューターのコスト・パフォーマンスに左右される。

RSA社は、２つの素因数から構成される合成数の素因数分解がどれだけ短時間で成功するかを競うコンテストを実施しており、1999年8月には、数体ふるい法を用いることにより、約５か月間を費して512 bitの合成数が素因数分解されている（クロック数300MHzのパソコンに換算して約60台を使用）[22]。こうしたことから、現時点では、ISO/IEC 14888-3や金融機関向け情報セキュリティ対策の指針ISO/TR 13569（ISO［1997］）等において、RSA暗号／署名等、素因数分解問題を利用する公開鍵暗号方式では、1024 bit以上の鍵長を利用することが推奨されている。

ロ. 能動的攻撃に対する安全性

能動的攻撃が利用可能な場合には、RSA署名のナイーブな方式をはじめとするRSA署名の主要な方式に対して、これまでにさまざまな攻撃が発表されている。各種のRSA署名と主要な攻撃を整理すると以下の図７のとおり。

図７　RSA署名に基づく方式に対する能動的攻撃と新方式の提案



22 オランダの研究機関CWI（Centrum voor Wiskunde en informatica）の調査によると、512 bitの公開鍵を利用するRSA方式は、インターネット上での電子商取引に利用されるアプリケーションの約95%において採用されている。

①ナイーブな方式に対する攻撃

RSA署名のナイーブな方式に対しては、すでにデスメット=オドリズコ（Desmedt-Odlyzko）の攻撃法によって、ある一定条件を満たすメッセージに対する署名を偽造可能であることが示されている（Desmedt and Odlyzko［1986］）。この攻撃法は、適応的選択文書攻撃の一種であり、メッセージのハッシュ値が小さな素数の積となっている場合に有効となる。攻撃に必要な計算量は、ハッシュ値のサイズに依存し、法nのサイズには依存しない。本攻撃法の手順は以下のとおり。

〈デスメット=オドリズコの攻撃法〉

（i）攻撃者は、ハッシュ値が比較的小さな素数に素因数分解できるメッセージMを選択する。例えば、Mのハッシュ値$H(M)$が$H(M)=p_1\times p_2\times\cdots\times p_{k-1}\times p_k$となる場合を考える（$p_i$は素数、kは自然数）。

（ii）攻撃者は、上記の条件を満たすMを生成し、署名者からMに対する署名Sを入手する。署名Sは以下のとおり。

$$S = H(M)^d \bmod n = p_1^d\times p_2^d\times\cdots\times p_{k-1}^d\times p_k^d \bmod n$$

（iii）攻撃者は、入手した複数の署名を使って、各素数p_iをd乗してmod nを計算した値（$p_i^d \bmod n$）を得る。

（iv）攻撃者は、入手した$p_i^d \bmod n$の値を組み合わせて、ハッシュ値が小さな素数の積となる任意のメッセージに対する署名を偽造する。

本攻撃法はRSA署名の乗法性[23]を利用したものであり、本攻撃法への対応策として、（A）署名変換対象データに冗長性を付加する方法、（B）署名変換対象データの生成にハッシュ関数を利用する方法が提案された（Misarsky［1998］）。これらの方式が、PKCS #1 Ver. 2.0やISO/IEC 9796等である。

②冗長性を利用する方式

ISO/IEC 9796は、ナイーブな方式に対して、署名変換対象データに冗長性をもたせることによってRSA署名の乗法性を回避し、能動的攻撃に対する安全性を高めるというアイデアに基づいて構成されたデジタル署名方式であり、能動的攻撃に対して十分な安全性を有していると評価されていた（Guillou et al.［1991］、Misarsky［1998］）。

しかし、1999年8月、カッパースミス（Coppersmith）、ハレビ（Halevi）、ジュルタ（Julta）は、ISO/IEC 9796に対する新たな攻撃法を発表し、ISO/IEC 9796が適応的選択文書攻撃に対して存在的偽造が可能であることを示した（Coppersmith et al.［1999］）。カッパースミスらの攻撃法の概要を説明すると以下のとおり。

23 RSA署名の乗法性とは、メッセージを変数とするRSA署名関数 $Sig(m)=m^d \bmod n$が分配法則を満たすことを意味する。すなわち、データAとBに対する署名をそれぞれ$Sig(A)$、$Sig(B)$とすると、AB（AとBの積）に対する署名は $Sig(A)\times Sig(B)$ となる。

〈カッパースミスらの攻撃法〉

【攻撃の概要】

まず、ISO/IEC 9796の署名変換対象データのフォーマットを満足し、かつ、素因数がいずれも小さな数となるデータを集める。次に、それらのデータに対する署名を正当な署名者から入手し、その署名を利用して、デスメット=オドリズコの攻撃法と同様の方法により、他のメッセージに対する署名を偽造する。

【攻撃の手順（例 法nが1024 bitの場合）】

（1）ISO/IEC 9796のフォーマットを満足するデータx・yの選択

ISO/IEC 9796の署名変換対象データ$U(M)$のbitパターンがほぼ同一のbitパターンの繰返しになっている点に着目する。$U(M)$のbitパターンは、以下のとおり、3種類のbitパターンA, B, C（それぞれ64 bit）によって表される。

$$U(M)=[\,A \| C \| \cdots (\text{Cの繰り返し}) \cdots \| C \| B] \cdots (*)$$

となる。ただし、A、B、Cのbitパターンは以下のとおり（u, vはそれぞれ任意の16 bitのデータ）。

・bit パターンA：$[s_1(u) \| s_2(v) \| u \| v]$（$U(M)$の最初の64 bitに対応）

・bitパターンB：$[s(u) \| s(v) \| u \| 0110]$（$U(M)$の最後の64 bitに対応）

・bitパターンC：$[s(u) \| s(v) \| u \| v]$（上記以外のデータに対応）

（＊）のbitパターンを考慮すると、**以下の①と②で表されるデータxとy（それぞれ64 bit、960 bit）の積x・y（1024 bit = 64 + 960）は、$U(M)$のフォーマットを満足する。このようなx・yは、2^{22}個（$=2^6\times2^8\times2^8$）存在する。**

①xは64 bitのデータであり、x=［a ‖b ‖c ‖d］（a,b,c,dはそれぞれ16 bitのデータ）となる。a,b,c,dは以下の条件を満足する。

（i）aとdはそれぞれbitパターンA、Bを満足し、a+dがbitパターンCを満足する（このようなaとdのペアは2^6個存在する）。

（ii）bとcはともにbitパターンCを満足する（このようなbおよびcはそれぞれ2^8個存在する）。

②yは960 bitのデータであり、以下のとおり。

$$y=\sum_{i=0}^{20}2^{48i}=[1_{16} \| 001_{16} \| \cdots (\text{“}001_{16}\text{”の繰り返し}) \cdots \| 001_{16}]$$

xとyの積は以下のようになり、$U(M)$のフォーマットと一致する。

x・y =［a ‖…（“‖b ‖ c ‖a+d ‖”の繰り返し）…‖d］

（2）小さな素因数から構成されるx・yの選択

2^{22}個存在するx・yの中から、すべての素因数がある一定数（例えばpとする）以下となるものを選択する。例えば、$p=2^{16}$と設定すると、すべての素因数が2^{16}以下となるx・yの数は約2^{14}個となる。

― すべての素因数があるp以下となる数の存在確率はpに依存する。例えば、$p=2^{16}$の場合の存在確率は$2^{-7.7}$となり、攻撃に利用可能なx・yの数は約2^{14}個（ $2^{22}\times2^{-7.7}$）。

（３）攻撃に利用する署名の入手

$U(M)$のフォーマットを満足し、すべての素因数が小さな素数となるx・yを入手し、x・yを正当な署名者に送付する。正当な署名者は、x・yに対する署名$S = (x \cdot y)^d \bmod n$を生成・返送する。これを繰り返して大量の署名を入手する。

（４）署名の偽造

攻撃者が入手した署名Sは小さな素数のみによって構成されており、デスメット=オドリズコの手法によって、別のデータに対応する署名を偽造できる。ただし、署名偽造が可能なデータは小さな素数の積となるものに限定される。

署名の偽造は法nの素因数分解に比べて効率よく実行できる。法nのサイズが1024 bitの場合には約3000の署名が必要であり、１台のパソコンで１日足らずで署名偽造が可能となると試算されている。

カッパースミスらは、ISO/IEC 9796の欠点として、「ISO/IEC 9796では署名変換対象データの生成に３種類の換字変換のみが利用されているため、ISO/IEC 9796のフォーマットに適合する署名変換対象データを生成するようなデータをみつけることが比較的容易となる」としている。カッパースミスらは、署名変換対象データの生成にハッシュ関数を利用する方式が望ましいとしており、とくに「フル・ドメイン・ハッシュ関数を利用し、安全性が証明されている署名方式」を挙げている。

この研究によってISO/IEC 9796の安全性に対する信頼が著しく低下し、ISO/IEC JTC1/SC27では、1999年10月にISO/IEC 9796を取り下げることを決定した。

③ハッシュ関数を利用する方式

RSA署名のナイーブな方式の安全性を高めるもうひとつの方法が、「署名変換対象データの生成にハッシュ関数を利用する」というものである。このアイデアに基づく主なデジタル署名方式として、PKCS #1 Ver. 2.0、ISO/IEC 9796-2、PSS署名、PSS-R署名、FDH-RSA署名が挙げられる。このうちISO/IEC 9796-2には、これまでにいくつかの攻撃法が発表されている。

（A）ISO/IEC 9796-2に対する攻撃と標準化の経緯

ISO/IEC 9796-2は、1996年1月にSC27において標準化が開始された（最初に提案された方式を旧ISO/IEC 9796-2と呼ぶ）。ところが、フランスのギロー（Girault）とミザルスキー（Misarsky）が、旧ISO/IEC 9796-2の署名検証に用いられている剰余演算を利用した適応的選択文書攻撃を発表した（ギロー=ミザルスキーの攻撃法、Girault and Misarsky［1997］）。このため、SC27では、上記剰余演算の代わりにハッシュ関数を採用した改良方式が提案され、新ISO/IEC 9796-2として標準化が進められた。また、同時に、別の国際標準案として、旧ISO/IEC 9796-3（メッ

セージ検証コードを利用したメッセージ復元型デジタル署名方式）が提案された[24]。

新ISO/IEC 9796-2は1997年9月に国際標準として成立したが、1999年4月に、コロン（Coron）、ナカッシェ（Naccache）、スターン（Stern）によって、法nの素因数分解よりも効率的な適応的選択文書攻撃が提案された（CNS攻撃法、Coron et al.［1999］）。CNS攻撃法の概要は以下のとおり[25]。

〈CNS攻撃法〉

【攻撃のアイデア】

デスメット=オドリズコの攻撃同様、署名変換対象データ$U(M)$が小さな素因数のみから構成されるメッセージMを見つける。ただし、$U(M)$を直接素因数分解することは困難なため、代わりに、$a \cdot n - 2^8 U(M)$**が小さな素因数のみから構成され、そのサイズが十分小さな合成数になるMとaを見つける。**

– $a \cdot n - b \cdot U(M)$が小さな素因数のみから構成される場合、$U(M)$も必ず同様に小さな素因数のみから構成される。

【攻撃の手順（例 n：1024 bit、M：1024 bit、ハッシュ値：160 bit)】

（1）$a \cdot n - 2^8 U(M)$ のフォーマット

$U(M)$は左から8 bit分が “0110 1010”となる（$hash(M)$はMのハッシュ値）。

$U(M)=$	01	1	0 1010	Mの左848bit	$hash(M)$	1011 1100

〈$U(M)$のフォーマット（1024bit）〉

法nの左8 bitが “0110 1010”となっていない場合、適当なaを選択してnをa倍し、左8 bitが “0110 1010”となる1032 bit（=1024 bit+ 8 bit）のデータa・nを生成する。データa・nの右176 bitの部分をyとし、左8 bit “0110 1010”とyの間の部分（848 bit）をxとする。

$a \cdot n=$	0110 1010	x（848bit）	y（176bit）

〈a・nのフォーマット（1032bit）〉

このとき、Mの左848 bitをxに設定すると、$a \cdot n - 2^8 U(M)$ **のサイズは、ハッシュ値が160 bitであることから、176 bit以下となる**[26]。

24 本節の旧ISO/IEC 9796-3は、第３章で説明されているISO/IEC 9796-3と異なる。旧ISO/IEC 9796-3が1999年4月に廃止されたため、当時ISO/IEC 9796-4（離散対数問題に基づくメッセージ復元型デジタル署名方式の国際標準案）として標準化が進められていた国際標準案がISO/IEC 9796-3に名称変更された。

25 CNS攻撃法の詳細については、宇根［1999］を参照。

26 法nの左7 bit分が “110 1010”の場合（確率は$1/2^7$）、$a \cdot n - 2^8 U(M)$の代わりに$n - 2U(M)$を利用することで、$n - 2U(M)$のサイズを169 bitにすることができる。

$a \cdot n - 2^8 \cdot U(M) =$	0110 1010	x（848bit）	y（176bit）	
$-$	0110 1010	x	*hash*（M）	1011 1100 0000 0000

$= x - [hash(M) \| 1011\ 1100\ 0000\ 0000] \leqq 2^{176}$

〈$a \cdot n - 2^8 \cdot U(M)$の値（176 bit）〉

（2）Mの選択

攻撃に利用するメッセージMは、左848 bitの部分がxと等しくなるように設定され、残りの176 bitの部分（zとする）を攻撃者が選択することができる。したがって、Mの取り得る値の総数は2^{176}個となる。

攻撃者は、上記$a \cdot n - 2^8 U(M)$が小さな素因数のみから構成されるようにMを選択する。選んだMに対して$a \cdot n - 2^8 U(M)$が小さな素因数のみから構成されているかを確認する必要があるが、$a \cdot n - 2^8 U(M)$が176 bit以下なので、公開鍵のサイズ1024 bitに比べて非常に少ない計算量で素因数分解が可能となる。

（3）Mに対する署名の入手と別の署名の偽造

攻撃者は、選択したMから$U(M)$を生成して署名者に送付する。署名者は、Mに対する署名$U(M)^d \bmod n$を生成・返送する。**入手した署名は小さな素因数から構成されており、デスメット=オドリズコの攻撃法と同様の方法によって、別のメッセージに対する署名を偽造できる。**

（4）署名偽造に必要な計算量

必要な計算量やメッセージ数は、$a \cdot n - 2^8 U(M)$の素因数のサイズやハッシュ値のサイズに依存する一方、法nのサイズには依存しない。たとえ法nのサイズ を2048 bitにしたとしても、ハッシュ値が160 bitの場合、$a \cdot n - 2^8 U(M)$のサイズは常に176 bit以下となる。ハッシュ値が160 bitの場合、2^{61}のオーダーの計算量と2^{40}のオーダーのメッセージが必要となると試算されている。

ISO/IEC 9796-2の署名変換対象データには、一定の規則によるパディングデータとメッセージの一部が含まれる。コロンらは、「ISO/IEC 9796-2の欠点は、攻撃者が比較的容易に推定できるbitが署名変換対象データに多く含まれている点である」と指摘しており、対応策のひとつとして、「ハッシュ値のサイズが署名変換対象データのサイズに一致する、フル・ドメイン・ハッシュ関数の利用」を提案している。本攻撃法の発表を受けて、SC27では、ISO/IEC 9796-2の取扱いに関して現在検討が行われている。

（B）旧ISO/IEC 9796-3に対する攻撃法と標準化の経緯

一方、旧ISO/IEC 9796-3は、ギロー=ミザルスキーの攻撃法に対して安全性を確保できるように設計されていた（Misarsky［1998］）。しかし、ミザルスキーが、1997年、旧ISO/IEC 9796-3に対する適応的選択文書攻撃を提案した（ミザルスキーの攻撃法、Misarsky［1997］）ことから、SC27では、1999年4月に旧ISO/IEC 9796-3の標準化作業を中止している。

ハ. 安全性が証明されているデジタル署名方式の必要性の高まり

RSA署名のナイーブな方式の安全性を高める目的で冗長性を利用する方法やハッシュ関数を利用する方法が提案されたものの、これらの対応策が施されたISO/IEC 9796とISO/IEC 9796-2には、いずれも素因数分解よりも効率的な攻撃法が存在することが示された。これらの対応策は、確実に署名方式の安全性向上に結び付くことを示す数学的な根拠に基づいて評価されたわけではなく、あくまで既存の攻撃法や分析手法を前提として評価されていた。CNS攻撃法やカッパースミスらの攻撃法は、こうした評価が不十分であったことを示唆するものであり、デジタル署名方式の評価には、安全性に関する経験的な分析だけではなく、きちんとした数学的な根拠が示されていることが必要であるとの考え方を強く印象づけるものとなった[27]。

デジタル署名方式の安全性証明に関する研究は従来から進められてきたが、最近では、既存の実用性が高いデジタル署名方式を改良して、高い実用性を維持しつつ、安全性を証明することができるデジタル署名方式がいくつか提案されている。例えば、RSA署名の利用方法としては、FDH-RSA署名、PSS署名、PSS-R署名が挙げられる。次章では、こうしたデジタル署名方式における安全性証明に関する研究について説明する。

27 公開鍵暗号の安全性評価に関する研究においても、カッパースミスらの攻撃法と同様の研究成果が示されている。1998年6月、ブライヘンバッハー（Bleichenbacher）は、SSL Ver. 3.0に実装されたPKCS#1 Ver. 1.5（RSA暗号の利用方法のひとつ）の暗号文を能動的攻撃によって効率的に解読する方法を発表した（Bleichenbacher［1998］）。この結果、RSA社は、PKCS #1 Ver. 1.5の公開鍵暗号方式のアルゴリズムを再検討し、PKCS#1 Ver. 2.0としてOAEP*を採用した。ブライヘンバッハーの攻撃法の内容やPKCS #1 Ver. 1.5の見直しの経緯については、宇根・岡本［1999］を参照。

*OAEP（Optimal Asymmetric Encryption Padding）：1995年にベラーレとロガウェイによって提案されたRSA暗号の利用方法のひとつであり、ランダム・オラクル・モデルの仮定と、RSA暗号関数の一方向性の仮定のもとで、能動的攻撃に対して安全であることが証明されている（Bellare and Rogaway［1995］）。

５．デジタル署名方式の安全性証明に関する研究

（１）これまでの研究の流れ

ゴルトバッサー（Goldwasser）、ミカリ（Micali）、リベスト（Rivest）が1984年に「適応的選択文書攻撃に対して存在的解読が不可能なデジタル署名方式」の概念を定義して以来、このような安全性が証明可能なさまざまなデジタル署名方式が提案されてきたが、いずれも実用性の面で問題が残されていた（表12参照）。例えば、ゴルトバッサー=ミカリ=リベスト署名（Goldwasser, Micali and Rivest［1988］）やナオール=ユング（Naor-Yung）署名（Naor and Yung［1989］）等は、一定の仮定のもとでいずれも適応的選択文書攻撃に対して存在的偽造が不可能であることが証明されているが、署名生成・検証に莫大な計算量が必要であった。

近年では、証明可能な安全性と処理速度の面での実用性を兼ね備えたデジタル署名方式が相次いで提案されている。例えば、ベラーレとロガウェイは1996年にFDH-RSA署名、PSS署名、PSS-R署名を提案している（Bellare and Rogaway［1996］）ほか、ポワンシュバルとスターンは改良エルガマル署名を提案している（Pointcheval and Stern［1996］）。また、太田と岡本は、フィアット=シャミア署名やギユー=キスケータ署名においてランダム・オラクル・モデルと仮定すると安全性の証明が可能になることを示している（Ohta and Okamoto［1998］）。このほか、証明可能な安全性と実用性を兼ね備えた署名方式として、岡本=シュノア署名（Okamoto［1993］）、TSH-ESIGN（Okamoto et al.［1998］）、阿部=岡本署名（Abe and Okamoto［1999］）が提案されている。

表12　デジタル署名方式における安全性証明研究の流れ

研究成果	内　容	証明されている安全性	実用性	前提となる数学の問題
Goldwasser, Micali and Rivest [1988]	ゴルトバッサー=ミカリ=リベスト署名を提案	適応的選択文書攻撃に対して存在的偽造不可能	問題あり	素因数分解問題
Naor and Yung [1989]	ナオール=ユング署名を提案			
Rompel [1990]	一方向性関数により、最も安全な署名方式が構築可能であることを証明			
Okamoto [1993]	岡本=シュノア署名を提案	適応的選択文書攻撃に対して存在的偽造不可能	シュノア署名と同程度	離散対数問題
Bellare and Rogaway [1996]	FDH-RSA署名、PSS署名、PSS-R署名を提案		RSA署名と同程度	素因数分解問題
Pointcheval and Stern [1996]	エルガマル署名、DSA、シュノア署名に改良を行うことで安全性を証明		基となる方式と同程度	離散対数問題
Okamoto, Fujisaki, Morita [1998]	TSH-ESIGNを提案		ESIGNと同程度	素因数分解問題
Abe and Okamoto [1999]	阿部=岡本署名を提案		シュノア署名等と同程度	離散対数問題

（２）証明可能な安全性と実用性を兼ね備えたデジタル署名方式

イ. 各デジタル署名方式の安全性証明に必要な仮定

デジタル署名方式の安全性を証明するためには、ランダム・オラクル・モデルをはじめとするさまざまな仮定が必要とされる。各デジタル署名方式で利用されている数学的な仮定を整理すると、以下の表13のとおり。各デジタル署名方式における安全性証明について評価を行うためには、これらの仮定がどの程度現実的なものかに留意しておくことが重要である。

表13　各デジタル署名方式の安全性証明に必要な仮定

			ランダム・オラクル・モデルの仮定	ランダム・オラクル・モデル以外の仮定
素因数分解問題に基づく方式	署名添付・確定型	FDH-RSA署名	必要	RSA暗号関数の一方向性
	署名添付・確率型	フィアット=シャミア署名	必要	素因数分解問題の困難性
		ギユー=キスケータ署名	必要	RSA暗号関数の一方向性
		PSS署名	必要	RSA暗号関数の一方向性
		TSH-ESIGN	必要	e乗根近似問題の困難性
	メッセージ復元・確率型	PSS-R署名	必要	RSA暗号関数の一方向性
離散対数問題に基づく方式	署名添付・確率型	シュノア署名	必要	離散対数問題の困難性
		KCDSA	必要	離散対数問題の困難性
		岡本=シュノア署名	不要	・無相関一方向性ハッシュ関数の仮定 ・離散対数問題の困難性
		改良エルガマル署名	必要	離散対数問題の困難性
	メッセージ復元・確率型	阿部=岡本署名	必要	離散対数問題の困難性

ロ. 安全性証明に利用されている仮定の内容

①ハッシュ関数に関する仮定

（A）ランダム・オラクル・モデルの仮定

ランダム・オラクル・モデルの仮定は多くのデジタル署名方式に利用されている。ランダム・オラクル・モデルの仮定の内容は以下のとおり。

> **〈ランダム・オラクル・モデルの仮定〉**
> ハッシュ関数が、入力値に対して乱数[28]を出力し、かつ、同じ入力値に対して同じ乱数を出力する関数（ランダム・オラクル）である。

28 ランダム・オラクル・モデルにおける乱数は、通常暗号鍵の生成等に利用される疑似乱数ではなく、真正な乱数であることが必要とされる。

デジタル署名方式に利用されるハッシュ関数がランダム・オラクルの仮定を満足する場合、ハッシュ関数の入力値と出力値の間の相関がなくなり、安全性を証明する際の手掛かりとなる（藤岡［1999］）。例えば、FDH-RSA署名の場合、フル・ドメイン・ハッシュ関数にランダム・オラクル・モデルが仮定されており、署名変換対象データはメッセージと何ら関係のない乱数となり、デジタル署名からメッセージを推測することが困難となる。

ただし、ランダム・オラクル・モデルの仮定は仮想的な仮定であり、これを満足するハッシュ関数は現時点では存在しない。現在、実用的なハッシュ関数を利用してランダム関数を構成し、適応的選択文書攻撃に対して安全なデジタル署名の構成方法に関する研究が行われている。例えば、ポワンシュバルとバウデニーは、米国政府のハッシュ関数の標準であるSHA-1を利用して実用的なハッシュ関数を構成し、信頼できる第三者機関が署名生成モジュールを耐タンパー性を有する装置（署名生成装置）に格納した上で、①署名生成装置へのアクセス回数の上限、②ハッシュ関数の条件、③署名の有効期間等、適応的選択文書攻撃に対して安全性を確保するために必要となる条件について分析を行っている（Pointcheval and Vaudenay［1996］）。

（B）無相関一方向性ハッシュ関数の仮定

無相関一方向性ハッシュ関数の仮定は、岡本=シュノア署名において用いられる仮定である。岡本=シュノア署名における無相関一方向性ハッシュ関数の仮定の内容は以下のとおり。

〈無相関一方向性ハッシュ関数の仮定（岡本=シュノア署名の場合）〉

hを岡本=シュノア署名の署名生成に利用されるハッシュ関数とし、$h(x, m)=e$とする。hが以下の２つの条件（一方向性と無相関性）を満足する場合、「hは、岡本=シュノア署名の署名検証式$x = (g1)^{(y1)} \times (g2)^{(y2)} \times (v^e) \bmod p$に対して、無相関一方向性ハッシュ関数である」という。

（a）一方向性：任意のxとmが与えられたとき、$h(x, m)=h(x, m')$を満足するm'（ただし、$m \neq m'$）を見つけることが困難である。

（b）無相関性：任意のxに対して、岡本=シュノア署名の検証式の一部$x = (g1)^{(y1(i))} \times (g2)^{(y2(i))} \times (v^e) \bmod p$（ただし$i = 1,...,t$）を満たす（$y1(i), y2(i)$）（$i = 1,...,t$）を見つけることが困難とする。

このとき、任意のメッセージmに対して、$x = (g1)^{(y1)} \times (g2)^{(y2)} \times (v^e) \bmod p$を満足し、かつ、$h(x, m)= e$を満足する$(x, y1, y2, e)$を見つけることが困難である。

本仮定における「一方向性」は、「任意の署名に対して、同じ署名が生成される複数の異なるメッセージを見つけることが困難である」ことを意味する。岡本=シュノア署名における署名は$(y1, y2, e)$であるが、この署名に対して万一

$e = h(x, m) = h(x, m')$を満足する（ただし、$m \neq m'$）が容易に見つかると、ひとつの署名が複数の異なるメッセージに対する署名となり、デジタル署名の機能のひとつである否認防止機能が満足されなくなる。

また、「無相関性」は、「任意のメッセージに対して、正当な署名者以外の人が、岡本=シュノア署名の署名検証式を満足する署名を見つけることが簡単ならば、任意のxに対して$x = (g1)^{(y1(i))} \times (g2)^{(y2(i))} \times (v^e) \bmod p$を満足する（$y1(i), y2(i)$）$(i = 1,...,t)$を見つけることも簡単である」ことを意味している。任意のメッセージmに対して、岡本=シュノア署名の署名検証式$x = (g1)^{(y1)} \times (g2)^{(y2)} \times (v^e) \bmod p$と、署名の一部の生成式である$e = h(x, m)$を同時に満足する（x, y1, y2, e）を容易に見つけることができるとすれば、mに対する署名（y1, y2, e）を容易に偽造可能となり、否認防止機能が満足されなくなる。

本仮定については厳密な証明が示されている訳ではないが、ランダム・オラクル・モデルの仮定と比較すると、より緩い仮定であるとみられている（岡本・藤崎［1999］）。

②暗号関数に関する仮定

暗号関数に関する仮定としては、RSA暗号関数の一方向性の仮定とe乗根近似問題の困難性の仮定が挙げられる。

（A）RSA暗号関数の一方向性の仮定

RSA暗号関数は、平文x、暗号文Cのもとで、$C = F(x, n, e) = x^e \bmod n$を満足する関数$F$によって表される。RSA暗号関数の一方向性の仮定の内容は以下のとおり。

> **〈RSA暗号関数の一方向性の仮定〉**
> RSA暗号の公開鍵（e, n）および n－1以下の自然数Yが与えられた場合、$Y = x^e \bmod n$を満足するxを求めることが計算量的に困難である。

本仮定は、RSA暗号関数の逆関数を計算することが困難であることを意味するものである。RSA暗号のアルゴリズムについては、これまでに法nの素因数分解よりも効率的な攻撃法が提案されていないものの、その安全性が証明されているわけではない。現時点では、RSA暗号のアルゴリズムの安全性に関する研究とともに、本仮定の現実的な妥当性に関する研究が進められている。

（B）e乗根近似問題の困難性の仮定

e乗根近似問題の困難性の仮定は、TSH-ESIGNの安全性証明に利用されており、その内容は以下のとおり。

〈e乗根近似問題の困難性の仮定〉

TSH-ESIGNの署名生成鍵を(e, n)とし、$n = p^2q$のサイズを3kとする(素数pとqのサイズはともにk)。サイズがk–1の自然数yが与えられた場合、以下の等式を満足するxを求めること(e乗根近似問題)が困難である。

$$[0 \| y] = [x^e \bmod n]^k$$

ただし、$[0 \| y]$は、yと1 bitのデータ“0”が結合したデータを表しており、$[A]^k$は、Aの左k bit分のデータを表す。

e乗根近似問題の困難性において利用されている等式は、TSH-ESIGNの署名検証式 $[s^e \bmod n]^k = [0 \| h(M)]$ である。TSH-ESIGNに利用されるハッシュ関数h、署名検証鍵(e, n)、bitサイズkは公表されているほか、署名者本人以外でもメッセージを容易に入手可能であるため、上記の署名検証式から唯一の未知数である署名sが容易に計算可能であるならば、署名の偽造が可能となる。

本仮定がどの程度現実的に妥当であるかは現在のところ未知であるが、ESIGN署名のアルゴリズムの安全性に関する研究とともに、本仮定の正当性について研究が進められているところである。

(3) 安全性が証明されているデジタル署名方式の標準化動向

安全性証明と実用性を兼ね備えたデジタル署名方式の提案を受け、デジタル署名方式に関する国際標準や業界標準において、こうした署名方式の採用に関する検討が行われている。主な検討の対象となっている方式は、PSS署名、PSS-R署名、TSH-ESIGN、阿部=岡本署名である(表14参照)。

表14　安全性証明と実用性を有するデジタル署名方式の検討動向

検討主体	検討の動向
ISO/IEC JTC1/SC27	・ISO/IEC 9796-2に規定されている既存の方式に対する代替方式としてPSS-R署名が提案されており、現在検討中。 ・ISO/IEC 9796-3に対して阿部=岡本署名が提案されているほか、ISO/IEC 15942-2に対してはEC-阿部=岡本署名が提案されており、いずれも現在検討中。
IEEE	・IEEE P1363aに規定されるデジタル署名方式として、KCDSA、PSS署名、TSH-ESIGNが提案されており、現在検討中。
PKCS #1	・RSA社がPKCS #1のデジタル署名方式としてPSS署名の採用を検討中。

イ. ISO/IEC JTC1/SC27における動向

ISO/IEC JTC1/SC27は、ISO/IEC 9796に対してカッパースミスらの攻撃法やCNS攻撃法が発表されたことを受けて、1999年10月にISO/IEC 9796を取り下げることを決定した。

また、ISO/IEC 9796-2については、CNS攻撃法が提案されたことを受けて、本国際標準の取扱いについて検討が行われているところである。こうした中、既存の攻撃法のみを前提とした従来の安全性評価のみを拠り所としていた署名方式よりも、安全性が証明されている署名方式の方が望ましいとする見方から、代替方式としてPSS-R署名が提案されている。本提案に対し、SC27では現在検討が進められている。

また、SC27においては、離散対数問題を利用したメッセージ復元型デジタル署名方式の国際標準案ISO/IEC 9796-3に対して、日本から阿部=岡本署名が提案されているほか、楕円離散対数問題に基づくデジタル署名方式の国際標準案ISO/IEC 15942-2に対しては、日本からEC-阿部=岡本署名が提案されている。SC27ではいずれの提案についても、現在検討を進めている段階である。

ロ. IEEEにおける動向

IEEE[29]では、公開鍵暗号技術を利用した鍵配送やデジタル署名等に関する標準規格P1363aの標準化が進められている。IEEE P1363aの標準化に対して、安全性が証明されているデジタル署名方式として、これまでにKCDSA（1998年8月提案）、PSS署名（1998年８月提案）、TSH-ESIGN（1998年11月提案）が提案されており、現在検討が行われている。

ハ. PKCS #1 Ver. 2.0における動向

RSA社は、1998年6月にブライヘンバッハーの攻撃法が発表されたことを受けて、PKCS #1 Ver. 1.5に定めていたパディングを利用したRSA暗号をOAEPに置き換え、PKCS #1 Ver. 2.0として1998年9月に発表した。その際、RSA社は、PKCS #1 Ver. 2.0に定められているデジタル署名方式についても見直しを行う方針を示し、「PSS署名を本標準に採用するかどうかについて現在検討中である」と発表していることから、今後PSS署名がPKCS #1 Ver. 2.0に採用される可能性もある。

29 IEEE（Institute of Electrical and Electronic Engineers）：会員数32万人以上を擁する電気・電子関連技術全般を対象分野とする学会。関連分野における新しい技術の発表の場を提供するほか、関連分野の技術を利用する際にリファレンスとなる技術標準の策定も行っている。IEEEに関する情報については、http://grouper.ieee.org/を参照。また、P1363aに関する情報については、http://grouper.ieee.org/groups/1363/addendum.htmlを参照。

6．おわりに

RSA署名をベースとしたデジタル署名方式の国際標準ISO/IEC 9796は、既存の攻撃法や分析手法を前提とした評価によって安全であるとみられてきたが、CNS攻撃法やカッパースミスらの攻撃法により、有効な攻撃法が存在することが示された。これを受けて、ISO/IEC JTC1/SC27は、本国際標準を取り下げることを1999年10月に決定した。この結果、「既存の攻撃法や分析方法に基づいた安全性評価方法では十分とはいえない」との認識が、暗号学者だけではなく、標準化に携わる実務家の間でも広がっている。

近年、処理速度の面での実用性と安全性証明を兼ね備えたデジタル署名方式が相次いで発表されている。こうしたデジタル署名方式の中には、安全性を証明するために、現実的な妥当性が必ずしも自明ではない仮定をおくことが必要とされる方式が少なくない。しかし、安全性証明による評価は、既存の攻撃法のみを前提とした場当たり的な評価とは異なり、デジタル署名方式に対する攻撃法や署名偽造の達成度等を分類したうえで、数学的な仮定と安全性との間の関連性を評価するというものである。このため、安全性証明による評価結果を利用することによって、従来の評価方法と比較して理論的により精緻な評価が可能となる。

デジタル署名方式の実装環境が多様化する中、デジタル署名を安全に利用するためには、既存の攻撃法を前提とする従来の評価だけでは不十分であることが明らかとなった。今後、金融機関がデジタル署名を利用したシステムを構築しようとする際には、実装環境を利用した攻撃法に対する安全性を考慮することに加えて、安全性証明の研究に代表される理論研究にも十分に留意していく必要があろう。

参考文献

内山成憲・斎藤泰一、「トレース２の楕円曲線上の離散対数問題について」、電子情報通信学会技術報告 ISEC98-27、pp.51-57、1998年

宇根正志、「RSA署名に対する新しい攻撃法の提案について －Coron-Naccache-Sternの攻撃法－」、『金融研究』第18巻別冊第１号、日本銀行金融研究所、1999年9月

――――・岡本龍明、「公開鍵暗号の理論研究における最近の動向」、『金融研究』第18巻第２号、日本銀行金融研究所、1999年4月

岡本龍明、「暗号の研究動向」、『NTT R&D』10月号、1999年

――――・藤崎英一郎、「安全性の証明のついたディジタル署名：TSH-ESIGNおよび（楕円）Okamoto-Schnorr」、『NTT R&D』10月号、1999年

――――・山本博資、『現代暗号』、産業図書、1998年

谷口文一、「金融業界におけるPKI・電子認証について ― 技術面、標準化に関する最近の動向を中心に」、『金融研究』第19巻別冊１号、日本銀行金融研究所、2000年4月

藤岡　淳、「ディジタル署名の実用性と安全性」、電子情報通信学会誌、Vol. 82、No. 6、pp. 580-586、1999年6月

Abe, M., and T. Okamoto, “A Signature Scheme with Message Recovery as Secure as Discrete Logarithm,” mimeo., 1999.

Adleman, L. M., “The function field sieve,” Algorithmic Number Theory, Lecture Notes in Computer Science, Vol. 887, pp. 108-121, Springer-Verlag, 1994.

Bellare, M., and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” Proceedings of the First Annual Conference on Computer and Communications Security, ACM, 1993.

――――― and ―――――, “Optimal asymmetric encryption,” Proceedings of EUROCRYPT ’94, LNCS, Vol. 950, pp. 92-111, Springer-Verlag, 1995.

――――― and ―――――, “The Exact Security of Digital Signatures －How to Sign with RSA and Rabin,” Proceedings of EUROCRYPT ’96, LNCS 1070, pp. 399-416, Springer-Verlag, 1996.

Bleichenbacher, D., “Generating ElGamal signatures without knowing the secret key,” Proceedings of EUROCRYPT ’96, LNCS 1070, pp.10-18, Springer-Verlag, 1996.

―――――, “Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1,” Proceedings of CRYPTO ’98, LNCS 1462, pp.1-12, Springer-Verlag, 1998.

Coppersmith, D., S. Halevi, and C. Jutla, “ISO 9796-1 and the new forgery strategy,” submission to IEEE P1363a, August 23, 1999.(http://grouper.ieee.org/groups/1363 /contrib.html)

Coron, J.-S., D. Naccache, and J. P. Stern, “On the Security of RSA Padding,” Proceedings of CRYPTO ’99, LNCS 1666, pp.1-18, Springer-Verlag, 1999.

Desmedt, Y. G., and A. M. Odlyzko, “A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes,” Proceedings of CRYPTO ’85, LNCS 218, pp.516-522, Springer-Verlag, 1986.

ElGamal, T. E., “A public key cryptosystems and a signature scheme based on discrete logarithm,” Proceedings of CRYPTO ’84, LNCS 197, pp. 10-18, Springer-Verlag, 1985.

Fiat, A., and A. Shamir, “How to prove yourself: Practival solutions to identification and signature problems,” Proceedings of CRYPTO ’86, LNCS 263, pp. 186-197, Springer-Verlag, 1987.

Frey, C., and H. G. Rück, “A Remark Concerning m-divisibility and The Discrete Logarithm in The Divisor Class Group of Curve,” Math. Comp., Vol. 62, No. 206, pp. 865-874, 1994.

Girault, M., and J. F. Misarsky, “Selective Forgery of RSA Signatures Using Redundancy,” Proceedings of EUROCRYPT ’97, LNCS 1233, pp. 495-507, Springer-Verlag, 1997.

Guillou, L. C., and J.-J. Quisquater, “A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory,” Proceedings of EUROCRYPT ’88, LNCS 330, pp. 123-128, Springer-Verlag, 1988.

———, ———, M. Walker, P. Landrock, and C. Shaer, “Precautions taken against various potential attack in ISO/IEC DIS 9796 “Digital signature scheme giving message recovery”,” Proceedings of EUROCRYPT ’90, LNCS 473, pp. 465-473, Springer-Verlag, 1991.

Goldwasser, S., S. Micali, and R. Rivest, “A digital signature scheme against adaptive chosen message attack,” SIAM J. Comp., Vol. 17, No. 2, pp. 281-308, 1988.

International Organization for Standardization, “ISO/TR 13569 Banking and related financial services ― Information security guidelines,” 1997.

——— and International Electrotechnical Commission, “ISO/IEC 9796 Information technology ― Security techniques ―Digital signature scheme giving message,” 1991.

——— and ———, “ISO/IEC 9796-2 Information technology ― Security techniques ― Digital signature scheme giving message recovery ―Part 2: Mechanisms using a hash-function,” 1997.

——— and ———, “ISO/IEC 14888-3 Information technology ― Security techniques ― Digital signature with appendix ―Part 3: Certificate-based mechanism,” 1998.

KCDSA Task Force Team, “The Korean Certificate-based Digital Signature Algorithm,” submission to IEEE P1363a, August 10, 1998.

Koblitz, N., “Elliptic curve cryptosystems,” Mathematics of Computation, Vol.48, pp.203-209, 1987.

———, *A Course in Number Theory and Cryptography*, Springer-Verlag, 1997. (ニール・コブリッツ著、櫻井幸一訳、『数論アルゴリズムと楕円暗号理論入門』、シュプリンガー・フェアラーク、1997年.)

Menezes, A., T. Okamoto, and S. A. Vanstone, “Reducing elliptic curve logarithms to logarithms in a finite field,” Proceedings of STOC, pp. 80-89, 1991.

———, P. C. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

Miller, V. S., “Use of elliptic curves in cryptography,” Proceedings of CRYPTO ’85, LNCS 218, pp. 417-426, Springer-Verlag, 1986.

Misarsky, J. F., “A Multiplicative Attack Using LLL Algorithm on RSA Signatures with Redundancy,” Proceedings of CRYPTO ’97, LNCS 1294, pp. 221-234, Springer-Verlag, 1997.

———, “How (not) to design RSA signature schemes,” Proceedings of Public Key Cryptography ’98, LNCS 1431, pp.14-28, Springer-Verlag, 1998.

Naor M., and M. Yung, “Universal one-way hash functions and their chosen ciphertext attacks,” Proceedings of STOC, pp. 33-43, 1989.

——— and ———, “Public-key cryptosystems provably secure against chosen ciphertext attacks,” Proceedings of STOC, pp.427-437, 1990.

National Institute for Standards and Technology, “Specifications for a digital signature standard,” Federal Information Processing Standard Publication 186, 1991.

Nyberg, K., and R. Rueppel, “A new signature scheme based on the DSA giving message recovery,” 1st ACM Conference on Computer and Communication Security, pp.58-61, ACM Press, 1993.

Ohta, K., and T. Okamoto, “On Concrete Security Treatment of Signatures Derived from Identification,” Proceedings of CRYPTO ’98, LNCS 1462, 1998.

Okamoto, T., “A fast signature scheme based on congruential polynomial operations,” IEEE Transactions on Information Theory, Vol. 36, No. 1, pp. 47-53, 1990.

——— , “Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes,” Proceedings of CRYPTO’92, pp.31-53, Springer-Verlag, 1993.

——— , E. Fujisaki, and H. Morita, “TSH-ESIGN: Efficient Digital Signature Scheme Using Trisection Size Hash,” Submission to IEEE P1363a, November 1998.

Pohlig, S., and M. Hellman, “An improved algorithm for computing logarithms over GF (p) and its cryptographic significance,” IEEE Transactions on Information Theory, Vol. 24, pp. 106-110, 1978.

Pointcheval, D., and J. Stern, “Security proofs for signature schemes,” Proceedings of EUROCRYPT ’96, LNCS 1070, pp. 387-398, Springer-Verlag, 1996.

———and S. Vaudenay, “On Provable Security for Digital Signature Algorithm,” a manuscript, 1996.(http://www.dmi.ens.fr/~pointche/)

Rompel, J., “One-way functions are sufficient for secure signatures,” Proceedings of STOC, pp.387-394, 1990.

RSA Laboratories, “PKCS #1: RSA Cryptography Specifications Version 2.0,” 1998. (ftp://ftp.rsa.com/pub/pkcs/ascii/pkcs-1v2.asc)

Rivest, R. L., A. Shamir, and L. Adleman, “A method of obtaining digital signatures and public key cryptosystems,” Communications of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

Satoh, T., and K. Araki, “Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves,” Commentarii Math, Univ. Sancti Pauli, 1998.

Schirokauer, O., “Discrete Logarithms and Local Units," Phil. Trans. R. Soc. Lond., A 345, pp.409-423, 1993.

———, D. Weber, and T. Denny, “Discrete Logarithms: The Effectiveness of the Index Calculus Method,” Algorithmic Number Theory, LNCS 1122, Springer-Verlag, pp. 335-361, 1996.

Schnorr, C. P., “Efficient signature generation for smart cards,” Proceedings of CRYPTO ’89, LNCS 435, pp.239-252, Springer-Verlag, 1990.

Shanks, D., “Class Number, a Theory of Factorization, and genera,” Proceedings of Symposium Pure Mathematics, AMS, 1985.

Semaev, I. A., “Evaluation of Discrete Logarithms in a Group of p-torsion Points of an Elliptic Curve in Characteristic p,” Math. Comp., Vol. 67, No. 221, pp. 353-356, 1998.

Smart, N. P., “The discrete logarithm problem on elliptic curves of trace one,” Journal of Cryptology, Vol. 12, No. 3, pp. 193-196, 1999.