IMES DISCUSSION PAPER SERIES

最近のデジタル署名における理論研究動向について

うね まさし　おかもと たつあき
宇根正志・岡本龍明

Discussion Paper No. 99-J-42

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES
BANK OF JAPAN

日本銀行金融研究所

〒103-8660 日本橋郵便局私書箱 30 号

IMES Discussion Paper Series 99-J-42
1999 年 11 月

最近のデジタル署名における理論研究動向について

うね まさし　　おかもとたつあき
宇根正志[*1]・**岡本龍明**[*2]

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上で、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下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
JEL Classification：L86、L96、Z00

*1 **日本銀行金融研究所研究第 2 課**（E-mail: masashi.une@boj.or.jp）
*2 **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）

本論文は、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。

目　次

頁

Ⅰ．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)、③一旦生成した署名に対して、その署名を生成した事実の否認防止（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上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、本攻撃法の発表によって、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Ⅱ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上で、第Ⅲ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Ⅳ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Ⅴ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。

Ⅱ．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、一旦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みつけることは困難である。

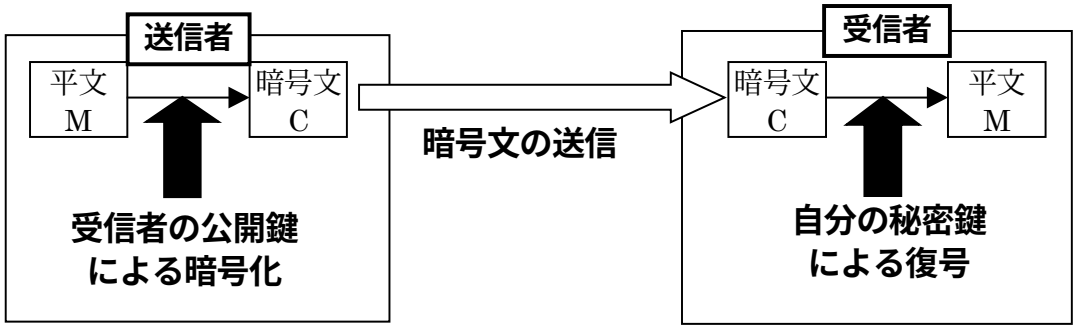
これらの要件を満たすデジタル署名は公開鍵暗号によって実現される[1]。

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

[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予め秘密鍵を共有し、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。

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の通り。



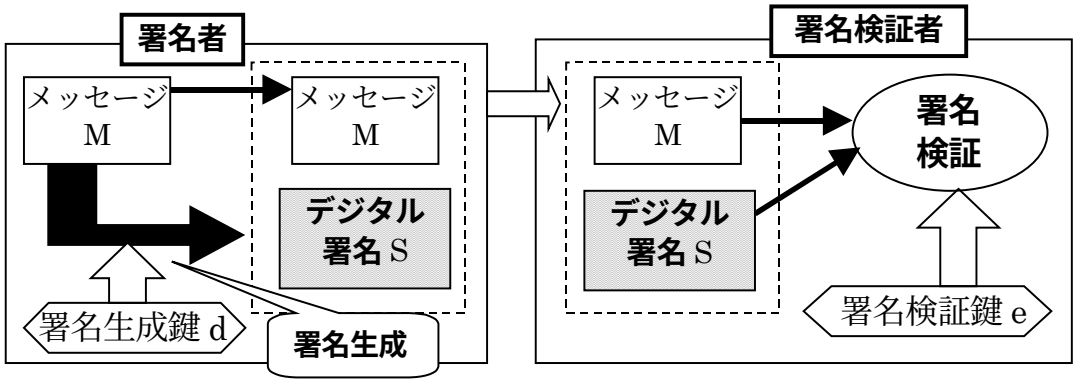
（ステップ 1）送信者は、受信者の公開鍵によって平文 M を暗号化して暗号文 C を生成し、受信者に送信する。
（ステップ 2）受信者は、暗号文 C を自分の秘密鍵で復号する。

図 1 公開鍵暗号によるデータ守秘機能（暗号通信）

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・検証方法の概略は次頁の図 2 の通り。

[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。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[1999]を参照。



（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。

図 2 公開鍵暗号を利用したデジタル署名の生成・検証

Ⅲ．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。

(1)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、①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、②同一のデータに対する署名が常に同一（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、表 1 参照）。

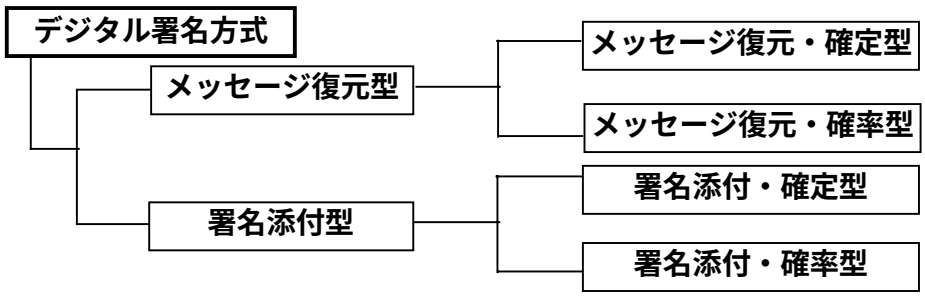


図 3　デジタル署名の形態による分類

（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大きくなるものの、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、以下の表 1 の通り。

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。 ・メッセージを入手するためには、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。

(2)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、様々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 種類に分類される。

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通り。

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、これまでに様々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

[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。

な素因数分解アルゴリズムを整理すると、以下の表 2 の通り。

表 2　合成数 n の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、以下の等式を満足する 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、これまでに様々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 のアルゴリズム（Schirokauer et al. 1996]）が効率的な解法となる（次頁の表 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、$k < (\log p)^{1/2-\varepsilon}$ (ε: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、

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

を満足する点(x,y)（x,y は F_p の要素）の集合。

---

[5] これらの解法の詳細については、宇根・岡本[1999]を参照。

楕円離散対数問題は以下の通り。

> **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、$A = xB$ を満足する自然数 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（Anomalous 曲線と呼ばれる）には SSSA アルゴリズム（Smart[1997]、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 Pohlig-Hellman のアルゴリズムや Shanks による BSGS（baby-step-giant-step）アルゴリズム（Shanks [1985]）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め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）
Anomalous 曲線（トレース 1）	SSSA アルゴリズム
上記以外の楕円曲線	Pohlig-Hellman のアルゴリズム、Shanks 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（Hasse の定理）。ただし、t の値は $-2\sqrt{p} \le t \le 2\sqrt{p}$ 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。

(3)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以下の表 5 の通り。

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	・Fiat-Shamir 署名 ・ESIGN （ISO/IEC 14888-3） ・Guillou-Quisquater 署名 （ISO/IEC 14888-3） ・RSA 署名（PSS 署名） ・TSH-ESIGN	・ElGamal 署名 （ISO/IEC 14888-3） ・Schnorr 署名 （ISO/IEC 14888-3） ・DSA （ISO/IEC 14888-3） ・KCDSA ・Okamoto-Schnorr 署名 ・改良 ElGamal 署名	・EC-ElGamal 署名 （ISO/IEC 15946-2） ・EC-Schnorr 署名 ・EC-DSA （ISO/IEC 14888-3, ISO/IEC 15946-2） ・EC-KCDSA （ISO/IEC 15946-2） ・Agnew-Mullin-Vanstone （ISO/IEC 14888-3） ・EC-Okamoto-Schnorr 署名 ・改良 EC-ElGamal 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）	・Nyberg-Rueppel 署名 ・ISO/IEC 9796-3 ・Abe-Okamoto 署名	・EC-Nyberg-Rueppel 署名 ・ISO/IEC 9796-3 ・EC-Abe-Okamoto 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）は以下の通り。

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

(1)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 の通り。

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）	Rivest, Shamir, Adleman [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）	Bellare, Rogaway [1996]	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における署名変換対象データの生成方法を改良	なし
署名添付・確率	Fiat-Shamir 署名	Fiat, Shamir [1987]	Fiat-Shamir 認証方式を署名方式に改良	あり
	Guillou-Quisquater 署名	Guillou, Quisquater [1988]	―――	あり
	ESIGN	Okamoto [1990]	―――	なし
	RSA 署名 （PSS 署名）	Bellare, Rogaway [1996]	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を改良	あり
	TSH-ESIGN	Okamoto, Fujisaki, Morita [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における署名変換対象データの生成方法を改良	なし
	RSA 署名 （ISO/IEC 9796-2）	―――	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を改良	なし
メッセージ復元・確率	RSA 署名 （PSS-R 署名）	Bellare, Rogaway [1996]	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を改良	あり

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 Rivest、Shamir、Adleman によって提案された署名方式である（Rivest, Shamir and Adleman [1978]）。ナイーブなデジタル署名方式は、公開鍵のみを利用する受動的攻撃（詳細については第Ⅳ章 2.(1)を参照）に対して法 n の素因数分解よりも効率的な攻撃法がみつかっていないものの、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（詳細については第Ⅳ章 2.(1)を参照）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、ナイーブな RSA 署名方式を基にした様々な改良方式が提案され、いくつかの改良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。ナイーブな RSA 署名における鍵生成、署名生成・検証方法は以下の通り。

＜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。

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

ナイーブな方式では、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 RSA 署名として FDH-RSA 署名が提案されている（詳細は次節）。

---

[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Ⅳ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。

(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ものであり、RSA 署名の利用方法の 1 つである。FDH-RSA 署名は、1996 年に Bellare と Rogaway によって提案された。FDH-RSA 署名では、利用されるフル・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Ⅴ章 2.(2)①を参照）を仮定することによって、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も偽造不可能であることが証明されている（Bellare and Rogaway[1996]）。FDH-RSA 署名における署名生成・検証方法は以下の通り。

＜FDH-RSA **署名**＞
【鍵生成】署名生成鍵と署名検証鍵の生成方法は、ナイーブなデジタル署名方式と同一。
【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。
$H(M) = S^e \bmod n$

Bellare と Rogaway 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。PKCS #1 Version 2.0 におけるデジタル署名方式の署名生成・検証方法は以下の通り。

＜PKCS #1 Ver. 2.0＞

【署名生成鍵】d

【署名検証鍵】(e,n)（n は k byte）

【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>パディングデータを PS とすると、PS のサイズは（k－t－3）byte となり、各 bit の値をすべて"1"とする。この結果、SR は以下のフォーマットとなり、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。

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

<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 のデジタル署名方式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。

[10] PKCS（Public-Key Cryptosystem Standard）は、RSA 社が策定する公開鍵暗号に関する技術仕様であり、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の方法から、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。

[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。

②署名添付・確率型署名方式

(A)Fiat-Shamir **署名**

Fiat-Shamir 署名は、認証者と被認証者との間でデータを 3 回やり取りしてユーザー認証を行う（三交信認証方式と呼ばれる）Fiat-Shamir 認証方式に基づいて構成された署名方式である（Fiat and Shamir[1987]）。Fiat-Shamir 署名は、素因数分解問題の困難性に加えて、利用する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いかなるメッセージに対する署名も偽造不可能であることが証明されている（Pointcheval and Stern[1996]）。署名生成・検証方法は以下の通り。

＜Fiat-Shamir **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 を生成した上で、k 個の n−1 以下の乱数 si (i = 1,…,k)を生成（ただし si は n を法とする乗法群の要素）。正整数 si を用いて、$v_i = s_i^{-2} \bmod n$ (i = 1,…,k)を計算。

【署名生成鍵】p, q, si (i = 1,…,k)

【署名検証鍵】n, vi (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)
- ei(i = 1,…,k)：e を bit 表現した時の各 bit の値（各 ei 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

(B)Guillou-Quisquater **署名**

Guillou-Quisquater 署名は、三交信認証方式の 1 つである Guillou-Quisquater 認証方式に基づいて構成された署名方式であり（Guillou and Quisquater[1988]）、ISO/IEC 14888-2 に規定されている。Guillou-Quisquater 署名は、利用する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ともに、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いかなるメッセージに対する署名も偽造不可能であることが証明されている（Pointcheval and Stern[1996]）。署名生成・検証方法は以下の通り。

＜Guillou-Quisquater **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、$xa^e \bmod n = 1$ を満足する a をみつける。

【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。

・乱数 k を生成し、$r = k^e \bmod n$ を計算。

・$l = h(m \| r)$

・$s = ka^l \bmod n$

【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。

・$u = s^e x^l \bmod n$

・$l' = h(m \| u)$

(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。ESIGN は、付録型のデジタル署名方式として、ISO/IEC 14888-3 に規定されている。ESIGN の署名生成・検証方法は以下の通り。

＜ESIGN＞

【鍵生成】大きな素数 p と q（p＞q）を選び、$n = p^2q$ 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）。

- $Q = (h(M) - (x^k \bmod n))/pq$
- $y = w/(kx^{k-1}) \bmod p$（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 の検証式に利用されるべき乗 k について、k = 2,3 に対しては、Brickell and deLaurentis[1986]によって署名の偽造方法が発表されているものの、k≧4 の場合に対しては、現時点では効率的な署名の偽造方法は示されていない。現在では、安全性の観点から、ESIGN のパラメーターとして、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ものであり、RSA 署名の利用方法の 1 つである。PSS 署名は、Bellare と Rogaway によって提案された（Bellare and Rogaway[1996]）。PSS 署名は、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いかなるメッセージに対する署名も偽造することができないことが証明されている。PSS 署名の署名生成・検証方法は以下の通り。

＜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 署名では、メッセージや乱数を理想的なランダム関数によってランダム化した上で結合し、さらに検証値として"0"を 1 bit だけ左から結合して署名変換対象データを生成する仕組みとなっている。また、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とされる計算量に 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 を改良したデジタル署名方式である（Okamoto, Fujisaki and Morita[1998]）。TSH-ESIGN では、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いかなるメッセージに対する署名の偽造も不可能であることが証明されている。TSH-ESIGN の署名生成・検証手順は以下の通り。

＜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 3 の通り。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た。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、RSA 署名の利用方法の 1 つである。ISO/IEC 9796 では、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（ISO/IEC [1991]）[13]。ISO/IEC 9796 の署名生成・検証手順は以下の通り。

＜ISO/IEC 9796＞

【鍵生成、署名生成・検証鍵】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）、以下の計算によって M に対する署名 S を生成。

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

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 mi(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、

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

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 を反転させる変換である）。

【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。

安全性については、これまでにいくつかの攻撃法（Coppersmith et al. [1999]等）が提案されたことから、1999 年 10 月、SC27 において ISO/IEC 9796 を取り下げることが決定した（詳細は第Ⅳ章 3.(2)②を参照）。

[13] ISO/IEC 9796 の Editor であるフランスの Guillou らは、公開鍵 n よりも非常に小さな署名変換対象データを利用することによっていくつかの攻撃法が適用可能となることを指摘するとともに、ISO/IEC 9796 が、それらの攻撃法に対して十分な安全性を確保できるように設計されていることを示した（Guillou et al. [1991]）。

(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 は RSA 署名の利用方法の１つである。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することができる（ISO/IEC[1997]）[14]。

＜ISO/IEC 9796-2＞

【鍵生成、署名生成・検証鍵】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 3 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。

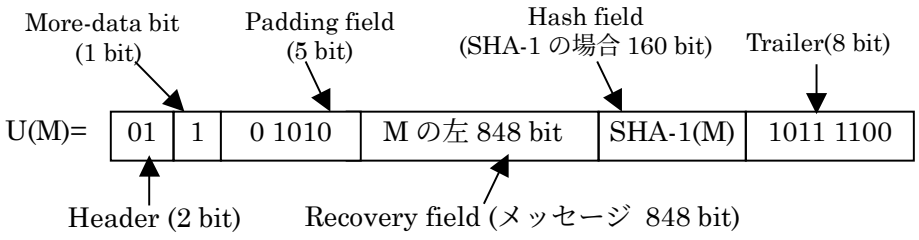


図 3　U(M)**のフォーマット（例** n **と** M: 1024 bit, **ハッシュ値**: 160 bit**）**

図 3 にあるように、法 n およびメッセージが 1024 bit、160 bit のハッシュ値をもつハッシュ関数（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通り。

(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 1024 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Ⅳ章 3.(2)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。

[14] 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ものであり、RSA 署名の利用方法の 1 つである。PSS-R 署名は、Bellare と Rogaway によって提案された（Bellare and Rogaway [1996]）。PSS-R 署名は、PSS 署名と同様の条件の下で、適応的選択文書攻撃に対して安全であることが証明されている。PSS-R 署名における生成・検証方法は以下の通り。

＜PSS-R 署名＞

【鍵生成、署名生成・検証鍵】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。

⑵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 8 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 8 の通り。

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	ElGamal 署名	ElGamal [1985]	―――	なし
	Schnorr 署名	Schnorr [1990]	ElGamal 署名の改良方式	あり
	DSA	NIST [1991]	ElGamal 署名および Schnorr 署名の改良方式	なし
	KCDSA	―――	DSA の改良方式	あり
	Okamoto-Schnorr 署名	Okamoto [1993]	Schnorr 署名の改良方式	あり
	改良 ElGamal 署名	Pointcheval, Stern [1996]	ElGamal 署名の改良方式	あり
メッセージ復元・確率型	Nyberg-Rueppel 署名	Nyberg, Rueppel [1993]	DSA の改良方式	なし
	ISO/IEC 9796-3	―――	DSA の改良方式	なし
	Abe-Okamoto 署名	Abe, Okamoto [1999]	―――	あり

①署名添付・確率型署名方式

(A)ElGamal **署名**

ElGamal 署名は、ElGamal[1985a]によって提案された署名方式であり、離散対数問題に基づく初めての方式である。ElGamal 署名には、(i)署名生成の都度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。ElGamal 署名は、ISO/IEC 14888-3 に規定されている。署名生成・検証手順は以下の通り。

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

ElGamal 署名の署名生成には乱数 k が利用されており、同一のメッセージに対する署名が毎回異なる確率型の署名方式となっている。ElGamal 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。

これまでに ElGamal 署名を改良した様々な方式が提案されており、主な方式として、Schnorr 署名、DSA、改良 ElGamal 署名が挙げられる。

(B)Schnorr **署名**

Schnorr 署名は、Schnorr[1990]によって提案された方式であり、ElGamal 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。Schnorr 署名は、利用される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いかなる署名も偽造不可能であることが証明されている（Pointcheval and Stern [1996]）。また、Schnorr 署名は、国際標準案 ISO/IEC 14888-3 に記載されている。Schnorr 署名の署名生成・検証方法は以下の通り。

＜Schnorr **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[15]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。

Schnorr 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。後述する DSA においても、上記の手法が利用されている。

[15] 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。

(C)DSA

DSA は ElGamal 署名の改良方式であり、NIST[16][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、ISO/IEC 14888-3 に規定されている。DSA は、Schnorr 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、ElGamal 署名の署名長を 2 p から 2 q に短縮することを可能にした。署名生成・検証方法は以下の通り。

＜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]）。Pointcheval と 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 1 の改良方式では、署名生成式として $r = h2(g^k \bmod p)$、$t = (h1(M)+xr)/k \bmod q$ を定義し、署名検証式として $r = h2(g^{h1(M)/t} y^{r/t} \bmod p)$ を定義した上で、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というものである。また、第 2 の改良方式では、署名生成式として $r =(g^k \bmod p) \bmod q$、$t = (h(M, r) + xr)/k \bmod q$ を定義し、署名検証式として $r = (g^{h(M, r)/t} y^{r/t} \bmod p) \bmod q$ を定義した上で、ハッシュ関数 h がランダムオラクルモデルに従うというものである。

[16] NIST（National Institute of 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。

(D)KCDSA

KCDSA（Korean Certificate-based Digital Signature Algorithm）は、DSA をベースに改良を加えた方式であり、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]）。KCDSA そのものではないが、KCDSA 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。KCDSA は、利用されているハッシュ関数 h 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いかなるメッセージに対する署名も偽造不可能であることが証明されている。KCDSA の署名生成・検証方法は以下の通り。

＜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[17]。

【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】署名者の ID や署名検証鍵等から z を生成した上で、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。

KCDSA 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。

[17] 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]）。

(E)Okamoto-Schnorr **署名**

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Ⅴ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いかなるメッセージに対する署名も偽造することができないことが証明されている。Okamoto-Schnorr 署名の署名生成・検証方法は以下の通り（岡本・藤崎[1999]）。

＜Okamoto-Schnorr **署名**＞

【鍵生成】まず整数 k（サイズを t とする）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 を法とする乗法群において、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)

【署名検証鍵】(p, q, g1, g2, t, v, H, k)

【署名生成】メッセージ m に対して、以下の計算によって生成される（e, y1, y2）が署名となる。

- $x = (g1)^{(r1)}(g2)^{(r2)} \bmod p$（q 未満の自然数 r1 と r2 をランダムに選択）
- e = H(x, m)
- y1 = (r1 + e･s1) mod q
- y2 = (r2 + e･s2) mod 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。

Okamoto-Schnorr 署名の署名生成・検証に必要となる計算量は、Schnorr 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 1024 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 の通り。

表 9　Okamoto-Schnorr 署名と Schnorr 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）
Okamoto-Schnorr **署名**	280	300
Schnorr **署名**	240	280

（出典）岡本・藤崎[1999]

(F)**改良** ElGamal **署名**

改良 ElGamal 署名は、Pointcheval and Stern[1996]によって提案された方式であり、ElGamal 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ことで、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が証明されている。改良 ElGamal 署名の署名生成・検証方法は以下の通り。

＜改良 ElGamal **署名＞**
【鍵生成】署名生成鍵・検証鍵は ElGamal 署名と同一。
【署名生成鍵】 x
【署名検証鍵】(y, p, α)
【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。その上で、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 ElGamal 署名は、ElGamal 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 ElGamal 署名の処理速度は ElGamal 署名とほぼ同程度とみられている。しかし、安全性を証明するためには、ハッシュ関数 h がランダムオラクルモデルの仮定に従うことが必要となる。

②メッセージ復元・確率型署名方式

(A)Nyberg-Rueppel **署名**

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。Nyberg-Rueppel 署名の署名生成・検証方法は以下の通り。

＜Nyberg-Rueppel **署名**＞
【鍵生成】署名生成・検証鍵は DSA と同一。
【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 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。以下の計算によって署名（e, s）を生成。
・e = m'r mod p
・s= (xe+k) mod q
【署名検証】 以下の手順で署名を検証。
・0 < e < p かつ 0 ≤ s < q を確認。
・$v = g^{s}y^{(-e)} \bmod p$ を計算。
・m' = ve mod p を計算。
計算した m'が m' mod 16 = 6 を満足しているか否かを確認。
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]）によって廃止となったことを受け、ISO/IEC 9796-4 が ISO/IEC 9796-3 に変更となった（詳細は第Ⅳ章 3.(2)③参照）。本署名方式は、現在 SC27 において標準化の審議が行われている。ISO/IEC 9796-3 の署名生成・検証方法は以下の通り。

＜ISO/IEC 9796-3＞

【鍵生成】署名生成・検証鍵は DSA と同一。

【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 に一定の変換を行って U(M)を計算（U(M)のフォーマットについては図 4 参照）。乱数 k（$1 < k < q-1$）を生成し、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。

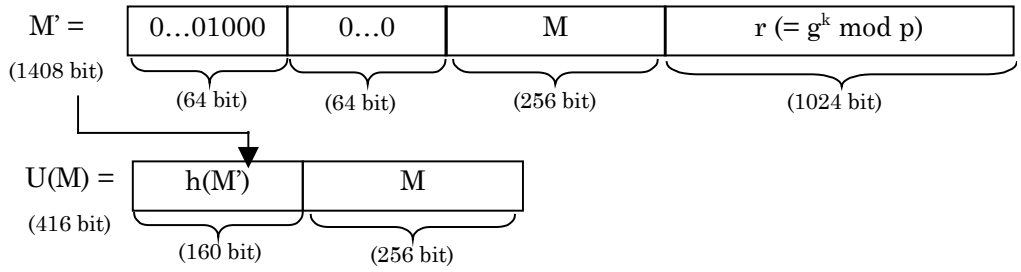


図 4　U(M)**のデータフォーマット（例 法** p: 1024 bit, M: 256 bit**）**

図 4 に示されているように、法 p およびメッセージ M のサイズがそれぞれ 1024 bit、256 bit の場合、M に一定のパディングを施したデータ M'が生成された後、ハッシュ関数 h によるハッシュ値 h(M')と M を連結したデータが U(M)となる。デジタル署名の生成には、DSA の署名生成鍵が利用される。

(C)Abe-Okamoto **署名**

Abe-Okamoto 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いかなるメッセージに対する署名も偽造不可能であることが証明されている。署名生成・検証方法は以下の通り。

＜Abe-Okamoto **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。

- F1：入力 k3 bit、出力 k4 bit
- F2：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

ランダム関数 F1, F2, H, G はいずれもランダムオラクルモデルの仮定を満足しているとする。

【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を行い、署名（r, z）を生成。

- $m' = F1(m) \| (F2(F1(m)) \oplus m)$
- $r = G(g^w \bmod p) \oplus m'$
- $c = H(r)$
- $z = (w + cx) \bmod p$

【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）。

- $m' = r \oplus G(g^z y^c \bmod p)$
- $m = [m']_{k3} \oplus F2([m']^{k4})$

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。

- $[m']^{k4} = F1(m)$

Abe-Okamoto 署名の署名生成・検証に必要な計算量は Schnorr 署名や Nyberg-Rueppel 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-Abe-Okamoto 署名の場合も EC-Nyberg-Rueppel 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-Okamoto [1999]）

(3)**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、ElGamal 署名、Schnorr 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-ElGamal 署名、EC-Schnorr 署名、EC-DSA[18]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以下の図 5 の通り。

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 **をもつ素数** 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。 $r = x((h(M)/s)*g + (r/s)*b) \bmod q$

図 5　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

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。

---

[18] 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 1 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。

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]）。

Ⅳ．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上で、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[19,20]。

上記①の研究の結果、「数学問題の困難性の観点からは安全性上問題がない」と評価された場合、次に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・評価が行われる。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デジタル署名方式である[21]。このように、実用

---

[19] 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非常に効率的な解法が提案される可能性もある。

[20] 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 ヵ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。

[21] 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 1 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 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

的なデジタル署名方式において最低限必要となる条件は、上記①および②の研究において致命的な欠陥が存在することが示されていないことである。

上記①および②に関する研究より、「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、次に上記③に関する研究・評価が行われる。③の研究では、「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[22]。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。

(1)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（次頁の表 10 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

難性に関してデジタル署名方式に利用する点で問題がなかったものの、Merkle-Hellman 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。

[22] 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。

攻撃であり、実行可能性は受動的攻撃よりも低い。特に、適応的選択暗号文攻撃では、一旦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。

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の偽造するという攻撃。
能動的攻撃	**一般選択文書攻撃**	攻撃者が予め指定した文書に対して真の署名者に署名させた後、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た後、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。

(2)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言えない。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その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。様々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、ISO/IEC JTC1/SC27 や ISO/TC68 等の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、メッセージ復元型のデジタル署名方式の国際標準 ISO/IEC 9796 および 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、ISO/IEC 9796 および ISO/IEC 9796-2 に対する攻撃法を中心に説明する。

(1)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 を素因数分解して秘密鍵 d を計算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となる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 署名の法 n の素因数 p と q が同一のサイズであり、p と q にそれぞれ特殊な性質[23]が存在しない場合、現時点では Adleman-Lenstra 版の数体ふるい法が最高速の解法となっている。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（300MHz のパソコンに換算して約 60 台を使用）[24]。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ガイドライン ISO/TR 13569（ISO[1997], [1998]）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。

---

[23] 例えば、p－1 や q－1 が小さな素数の積となるといった場合には、p－1 法と呼ばれる素因数分解の高速解法が適用可能となる。

[24] RSA 暗号／署名は、Netscape Navigator や Internet Explorer 等に組込まれた暗号通信プロトコル SSL（Secure Sockets Layer）に採用されるなど、金融分野をはじめとする幅広い分野において利用されている。特に、CWI*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。

*CWI（Centrum voor Wiskunde en informatica）：数学やコンピューターサイエンスを主な対象分野とするオランダの研究機関。CWI は、計算量理論とデータセキュリティと題するプロジェクトを 1997 年 1 月より開始しており、素因数分解や離散対数問題等、暗号に利用されている数学問題に関する研究を進めている。

⑵能動的攻撃に対する安全性

能動的攻撃が利用可能な環境を前提とする場合には、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、これまでに様々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次頁の図 6 の通り。

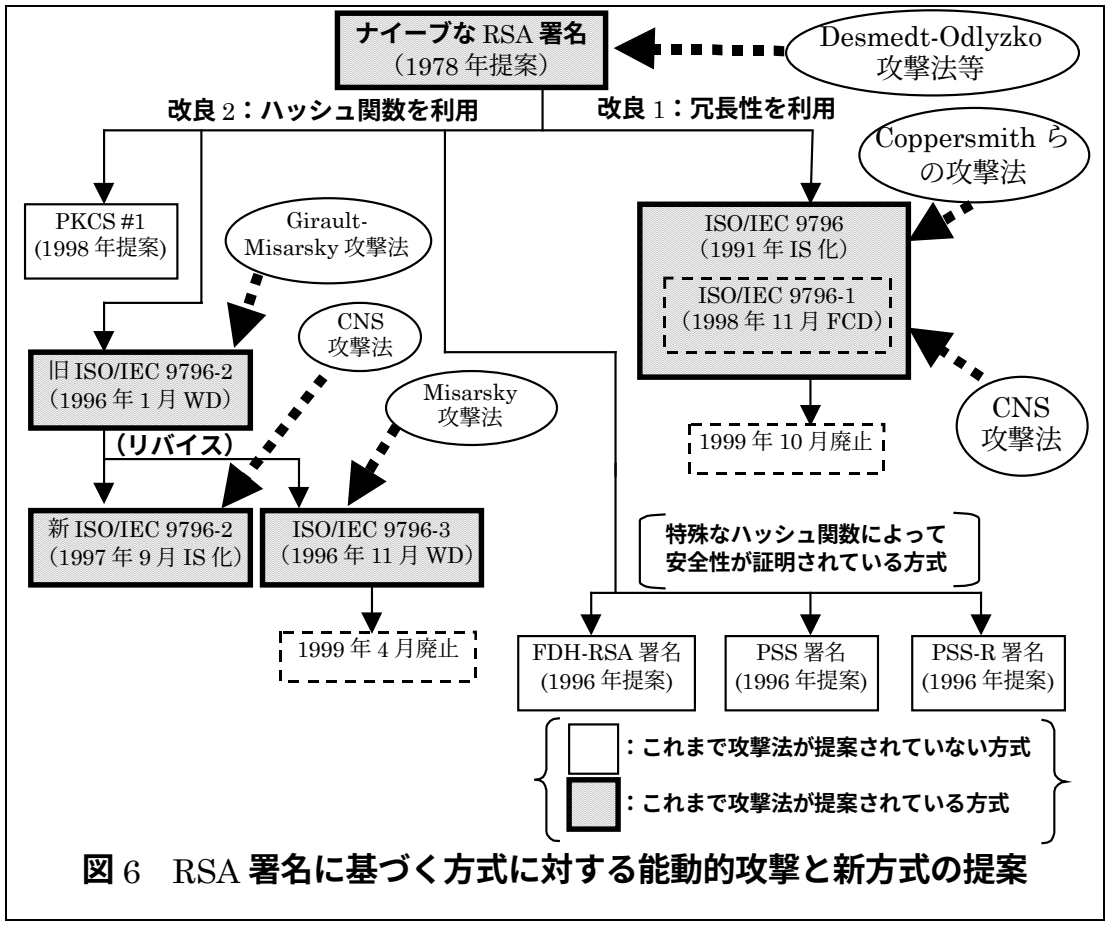


図 6　RSA 署名に基づく方式に対する能動的攻撃と新方式の提案

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、既に 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（存在的偽造が可能である）ことが示されている（Desmedt and Odlyzko [1986]）。Desmedt-Odlyzko 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。また、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通り。

＜Desmedt-Odlyzko **の攻撃法**＞

(i)攻撃者は、メッセージの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。ハッシュ関数を H とすると、例えば、M のハッシュ値 H(M)が $H(M)=p_1 \cdot p_2 \cdots p_{k-1} \cdot p_k$ 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 は以下の通り。

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

(iii)攻撃者は、各 H(M)が $p_1 \cdot p_2 \cdots p_{k-1} \cdot p_k$ という小さな素数の積となっていることを知っており、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[25]を利用した能動的攻撃であり、ナイーブな方式が能動的攻撃に対して安全性を確保できないことを示した。このため、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もたせる方法、(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]）。これらの対応方法を取り入れて設計されたデジタル署名方式が、PKCS #1 Ver. 2.0 や ISO/IEC 9796 等の方式である。

②冗長性を利用する方式

1991 年に国際標準となった ISO/IEC 9796（メッセージ復元型デジタル署名方式）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る（Misarsky[1998]）。本国際標準の Editor であったフランスの Guillou らは、署名変換対象データに冗長性をもたせることで、RSA 署名の乗法性を利用した攻撃法に対して十分な安全性を確保できるように設計したと発表している（Guillou et al.[1991]）。こうした研究成果により、ISO/IEC 9796 は、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。

しかし、1999 年 4 月、Coron、Naccache、Stern が、Desmedt-Odlyzko の攻撃法をベースとして、ISO/IEC 9796 のアルゴリズムの一部に変更を加えたデジタル署名方式に対する攻撃法（CNS 攻撃法と呼ぶ）を発表した（Coron et al.[1999]）[26]。さらに、Coppersmith、Halevi、Julta は、CNS

[25] 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)×Sig(B)となる。

[26] CNS 攻撃法は、後述する ISO/IEC 9796-2 に対しても適用可能である。CNS 攻撃法の詳細については宇根[1999]を参照。

攻撃法を改良した新たな攻撃法を1999年8月に発表した(Coppersmith et al. [1999])。CNS 攻撃法と Coppersmith らの攻撃法は適応的選択文書攻撃であり、特に Coppersmith らの攻撃法は、「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」ことを示した点で注目されている。Coppersmith らの攻撃法の概要を説明すると以下の通り。

＜Coppersmith 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、入手した署名を利用して、**Desmedt-Odlyzko **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**

まず、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 パターンの繰り返しになっている点に着目する。すなわち、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。

$$U(M)= [\ A \| C \| \cdots（C の繰り返し）\cdots \| C \| B]\quad \cdots（＊）$$

となる。ただし、A、B、C の bit パターンは以下の通り（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）

（＊）で表される 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、**ISO/IEC 9796 **に規定されている** 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。**

① x 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）によって構成される。ただし、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 を満足するとともに、a+d が bit パターン C を満足する（このような条件を満足する 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)。

② y は 960 bit のデータであり、以下の通り。

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

このとき、x と y の積は以下のようになり、上記(i)、(ii)の条件によって x･y は U(M)のフォーマットと一致する。

$$x\cdot y = [a \| \cdots（\text{”} \| b \| c \| a+d \| \text{”の繰り返し）} \cdots \| d]$$

(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（ $2^{16} \times 2^{-7.7}$）となる。

(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**

x･y **が小さな素数のみによって構成されていることから、攻撃者が入手した署名** S **も小さな素数のみによって構成されている。このため、**Desmedt-Odlyzko **の攻撃で利用された手法を用いることによって、別のデータに対応する署名を偽造することができる。**ただし、署名の偽造が可能なデータは、同様に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。Coppersmith らの試算によると、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されている。

Coppersmith らは、ISO/IEC 9796 の欠点として「署名変換対象データの生成にハッシュ関数を利用しない（hash-free encoding）」点を指摘している。具体的には、「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。Coppersmith 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、特に「ランダムオラクルモデルの仮定において安全性が証明されている、フル・ドメイン・ハッシュ関数を利用した方式」を挙げている。

こうした Coppersmith らの研究成果によって、ISO/IEC 9796 の安全性に対する信頼が著しく低下した。この結果、ISO/IEC JTC1/SC27 は、1999 年 10 月、現在 ISO/IEC 9796 として採用されている署名方式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 1 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。これらの

方式の中で、ISO/IEC 9796-2 に対しては、標準案として提案された当初から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（ハッシュ関数を利用したメッセージ復元型デジタル署名方式）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、その後フランスの Girault と Misarsky が、旧 ISO/IEC 9796-2 の署名検証に利用されている剰余演算の性質を利用した適応的選択文書攻撃を発表した（Girault-Misarsky の攻撃法、Girault and Misarsky [1997]）。このため、SC27 では、旧 ISO/IEC 9796-2 において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、別のデジタル署名方式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[27]。

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、Coron、Naccache、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]）。この結果、新 ISO/IEC 9796-2 は、「適応的選択文書攻撃によって存在的偽造が可能となる」ことが示されると同時に、実際に国際標準として幅広く利用されているデジタル署名方式[28]が攻撃された例として注目されている。CNS 攻撃法の概要は以下の通り[29]。

＜CNS 攻撃法＞
【攻撃のアイデア】
Desmedt-Odlyzko 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によって構成されるような M をみつける（このような M が見つかれば、後は Desmedt-Odlyzko の手法によって署名偽造が可能となる）。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･n－2^8U(M)**が小さな素因数のみ**

---

[27] 本節における旧 ISO/IEC 9796-3 は、第Ⅲ章で説明されている ISO/IEC 9796-3（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）とは異なる。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。

[28] 例えば、ISO/IEC 9796-2 は、Europay、MasterCard、Visa が共同で作成した IC カードに関する標準規格 EMV'96 に採用されている（Europay et al.[1996]）。具体的には、EMV'96 の ANNEX E において、ISO/IEC 9796-2 における署名生成・検証方法がそのまま記載されている。

[29] CNS 攻撃法の詳細については、宇根[1999]を参照。

から構成され、そのサイズが十分小さな合成数になるような M と a をみつける。

－　a･n－b･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･n－2^8U(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 848 bit	hash(M)	1011 1100

＜U(M)**のフォーマット**（1024 bit）＞

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･n=	0110 1010	x (848 bit)	y (176 bit)

＜a･n **のフォーマット**（1032 bit）＞

このとき、M の左 848 bit を x に設定すると、a･n－2^8U(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**[30]。

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

= x－[hash(M)‖1011 1100 0000 0000] ≦ 2^{176}

＜a･n－2^8･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･n－2^8U(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。選んだ各** M **に対して** a･n－2^8U(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ことが必要となるが、**a･n－2^8U(M)**のサイズが** 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を行うことができる。**

(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 $U(M)^d$ mod n を生成・返送する。これを繰り返し行い、大量の署名を入手する。**入手した署名は小さな素因数から構成されており、**Desmedt-

[30] 法 n の左 8 bit 分が"110 1010"の場合（確率は $1/2^7$)、a･n－2^8U(M) の代わりに n－2U(M) を利用することで、素因数分解が必要なデータ n－2U(M)のサイズを 169 bit にすることができる。

Odlyzko **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することができる。**

(4)署名偽造に必要な計算量

本攻撃法において必要な計算量やメッセージ数は、a･n－2^8U(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という特徴がある。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 の場合には、a･n－2^8U(M)のサイズは常に 176 bit 以下となる。

Coron らの試算では、ハッシュ値のサイズが 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ことが示されている。

ISO/IEC 9796-2 の署名変換対象データには、一定の規則によってパディングされる bit と、メッセージの一部が利用される bit が含まれる。例えば、法 n のサイズが 1024 bit の場合、メッセージのハッシュ値が 160 bit とすると、パディングおよびメッセージ自体が利用される bit は 864 bit（=1024－160）である。Coron らは、「ISO/IEC 9796-2 の欠点は、署名変換対象データに、攻撃者が比較的容易に推定することができる固定的な bit が多く含まれている点である」と指摘しており、ハッシュ関数を利用した対応策として、「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 の取り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、Girault-Misarsky 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、Misarsky 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（Misarsky 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た。

(3)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 および 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 2 種類の安全性を高める方法は、それらが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訳ではなく、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 Coppersmith 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[31]。

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。次章においては、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。

---

[31] 公開鍵暗号の安全性評価に関する研究においても、Coppersmith らの攻撃法と同様の研究成果が示されている。1998 年 6 月、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 1 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。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。Bleichenbacher 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 Bellare と Rogaway によって提案された RSA 暗号の利用方法の 1 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下で、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]）。

Ⅴ．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

「最も安全なデジタル署名方式」すなわち「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」の概念は、1984 年に Goldwasser, Micali and Rivest によって定義され、その後、最も安全なデジタル署名方式として、Goldwasser-Micali-Rivest 署名が提案された（Goldwasser, Micali and Rivest[1988]）。Goldwasser-Micali-Rivest 署名は、素因数分解問題が困難であるとの仮定の下で、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。しかし、署名生成・検証に必要な計算量が多くなるため、実用的な方式とはいえない。その後、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を証明するために必要となる仮定を一般化する試みがなされ、最終的に Naor と Yung さらに Rompel により、最も緩い仮定である「一方向性関数の利用可能性」だけを仮定すれば、能動的攻撃における存在的偽造が不可能となる署名方式を構築できることを示した（Naor and Yung [1989]、Rompel[1990]）。しかし、Naor and Yung および Rompel が提案した署名方式は、署名生成・検証に莫大な計算量が必要となるため、実用性は低い。このように、従来の研究においては、安全性が証明されるデジタル署名方式がいくつか提案されてきたものの、実用性の点で問題が残されていた。

最近では、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盛んに行われている。1996 年、Bellare と Rogaway は、ランダム・オラクルモデルと RSA 暗号関数の一方向性を仮定すれば、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可能な方式として、RSA 署名の改良方式「FDH-RSA 署名」、「PSS 署名」、「PSS-R 署名」を提案した（Bellare and Rogaway [1996]）。これらの署名方式に必要な計算量は RSA 署名の計算量とほぼ等しくなっており、実用性が高い。

また、Pointcheval と Stern は、ランダム・オラクルモデルと離散対数問題の困難性を仮定すれば、適応的選択文書攻撃に対する存在的偽造が不可能な方式として、ElGamal 署名に改良を加えた「改良 ElGamal 署名」を提案しているほか、Schnorr 署名についても、ランダムオラクルモデルの仮定の下で改良 ElGamal 署名と同様の安全性を証明可能であることを示している（Pointcheval and Stern[1996]）。Ohta and Okamoto は、Fiat-Shamir 署名や Guillou-Quisquater 署名においてランダムオラクルモデルと仮定すると、「最も安全なデジタル署名方式」を実現できることを示している（Ohta and Okamoto [1998]）。これらの方式は、いずれも高い実用性を有している。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、Okamoto-Schnorr 署名（Okamoto [1993]）、TSH-ESIGN（Okamoto et al. [1998]）、Abe-Okamoto

署名（Abe and Okamoto [1999]）が提案されている。

こうした安全性証明に関する研究の流れを整理すると、以下の表 12 の通り。

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]	Goldwasser-Micali-Rivest 署名を提案	適応的選択文書攻撃に対して存在的偽造が不可能	低い	素因数分解問題
Naor and Yung [1989]	Naor-Yung 署名を提案			
Rompel[1990]	一方向性関数を利用することによって、最も安全なデジタル署名方式が構築可能であることを証明			
Okamoto [1993]	Okamoto-Schnorr 署名を提案	適応的選択文書攻撃に対して存在的偽造が不可能	Schnorr 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]	ElGamal 署名、Schnorr 署名、DSA に若干の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-Okamto [1999]	Abe-Okamoto 署名を提案		Schnorr 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

(1)各デジタル署名方式の安全性証明に必要な仮定

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、かつ、高い実用性を有するデジタル署名方式において、どのような仮定が利用されているかを整理すると次頁の表 13 の通り。

Okamoto-Schnorr 署名を除き、いずれのデジタル署名方式においても、ランダムオラクルモデルの仮定が必要となっている。

また、素因数分解問題に基づく方式では、ランダムオラクルモデルに加え、素因数分解問題の困難性を仮定することによって安全性が証明されている方式は、Fiat-Shamir 署名である。また、RSA 暗号関数の一方向性を仮定することによって安全性が証明されている方式は、FDH-RSA 署名、Guillou-Quisquater 署名、PSS 署名、PSS-R 署名の 4 つである。このほか、TSH-ESIGN は、e 乗根近似問題の困難性の仮定によって、その安全性が証明されている。

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	Fiat-Shamir 署名	必要	素因数分解問題の困難性
		Guillou-Quisquater 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	Schnorr 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
		Okamoto-Schnorr 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 ElGamal 署名	必要	離散対数問題の困難性
	メッセージ復元・確率型	Abe-Okamoto 署名	必要	離散対数問題の困難性

一方、離散対数問題に基づく方式では、Schnorr 署名、KCDSA、改良 ElGamal 署名、Abe-Okamoto 署名の 4 つが、ランダムオラクルモデルと離散対数問題の困難性を仮定することによって、その安全性が証明されている。また、Okamoto-Schnorr 署名では、無相関一方向性ハッシュ関数の仮定と離散対数問題の困難性の仮定によって、安全性が証明されている。

このように、安全性が証明されているデジタル署名方式は、安全性に関する証明を成立させるために各々独自の仮定が必要となる。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。

(2)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

表 13 に整理されているように、現時点では、実用的なデジタル署名方式の安全性を証明する場合には、署名生成・検証アルゴリズムに利用されるハッシュ関数に何らかの仮定をおくことが必要となっている。特に、大部分のデジタル署名方式において、そこで利用されるハッシュ関数がランダムオラクルモデルの仮定を満たすことが求められている。また、Okamoto-Schnorr

署名において利用されるハッシュ関数には、無相関一方向性ハッシュ関数の仮定が必要とされている。

(A)ランダムオラクルモデルの仮定

ランダムオラクルは、入力値に対して乱数を出力し、かつ、同じ入力値に対しては同じ乱数を出力する関数のことである。ランダムオラクルモデルとは、このような性質を有する関数についての仮定であり、以下の通り。

＜ランダムオラクルモデルの仮定＞
デジタル署名に利用されているハッシュ関数が、入力値に対して乱数[32]を出力し、かつ、同じ入力値に対して同じ乱数を出力する「ランダムオラクル」である。

デジタル署名方式に利用されてい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るため、安全性を証明する際の手がかりとなる（藤岡[1999]）。例えば、FDH-RSA 署名の場合、メッセージはフル・ドメイン・ハッシュ関数によって法 n と同一サイズの署名変換対象データが生成されるが、フル・ドメイン・ハッシュ関数にはランダムオラクルモデルが仮定されているため、署名変換対象データは乱数となる。この結果、FDH-RSA 署名によって生成されるデジタル署名は、入力値であるメッセージとは何ら関係のない乱数に署名生成アルゴリズムを施したデータとなる。

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。しかし、実用的なハッシュ関数を利用してランダム関数を生成し、適応的選択文書攻撃に対して安全性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。Pointcheval and Vaudenay 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]）。

[32] 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。

(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、Okamoto-Schnorr 署名において用いられる仮定である。Okamoto-Schnorr 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通り。

＜無相関一方向性ハッシュ関数の仮定（Okamoto-Schnorr **署名の場合）＞**

まず、h を Okamoto-Schnorr 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。このハッシュ関数 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、「ハッシュ関数 h は、Okamoto-Schnorr 署名の署名検証式 x = ((g1)^(y1))((g2)^(y2))(v^e) mod 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 ≠ m'）をみつけることが困難である。

(b)**無相関性**：　まず、任意の x に対して、Okamoto-Schnorr 署名の署名検証式 x = ((g1)^(y1(i)))×((g2)^(y2(i)))×(v^e) mod p（ただし i = 1,…,t）を満足する（y1(i), y2(i)）（i = 1,…,t）をみつけることが困難であるとする。

　このとき、任意のメッセージ m に対して、x = $(g1)^{(y1)}$×$(g2)^{(y2)}$×(v^e) mod p を満足し、かつ、h(x, m)=e を満足する(x, y1, y2, e)をみ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みつけることが困難である」ことを意味するものである。Okamoto-Schnorr 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 ≠ m'）が容易にみつかると、1 つの署名が複数の異なるメッセージに対する署名となってしまい、デジタル署名の機能の 1 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、Okamoto-Schnorr 署名の署名検証式を満足する署名をみつけることが簡単ならば、任意の x に対して署名検証式を満足する（y1(i), y2(i)）（i = 1,…,t）をみ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、Okamoto-Schnorr 署名の署名検証式 x = $(g1)^{(y1)}(g2)^{(y2)}$ v^e mod 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。

なお、本仮定の正当性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、平文を x、暗号文を C とする場合、$C = F(x, n, e) = x^e \bmod n$ を満足する関数 F で表される。RSA 暗号関数の一方向性の仮定は、以下の通り。

> **＜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。

このように、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り、「RSA 暗号によって暗号化されたデータを解読することが計算量的に困難である」と言い換えることができる。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訳ではない。このため、RSA 暗号のアルゴリズムの安全性に関する研究とともに、本仮定の正当性について研究が進められている段階である。

(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いる仮定であり、その内容は以下の通り。

> **＜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 のサイズは共に k）。このとき、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

安全性が証明されており、かつ、処理速度の面から実用性が高い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、Abe-Okamoto 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 Abe-Okamoto 署名が提案されているほか、ISO/IEC 15942-2 に対しては EC-Abe-Okamoto 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採用を検討中。

(1)ISO/IEC JTC1/SC27 における動向

ISO/IEC JTC1/SC27 は、ISO/IEC 9796 に対して Coppersmith 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、本国際標準の取り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 Abe-Okamoto 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-Abe-Okamoto 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。

(2)IEEE における動向

IEEE[33]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。

(3)PKCS #1 Ver. 2.0 における動向

RSA 社は、1998 年 6 月に Bleichenbacher 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」と発表した。このため、今後 PSS 署名が PKCS #1 Ver. 2.0 に採用される可能性もある。

---

[33] 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。

Ⅵ．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 Coppersmith 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言えない」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上で、数学的な仮定と安全性と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。

以　上

参考文献

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、pp.51-83、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、pp.195-251、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てディジタル署名：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・電子認証技術について ― 技術面、標準化に関する最近の動向を中心に」、IMES Discussion Paper Series No. 99-J-30、日本銀行金融研究所、1999 年 8 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.

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.

Dolev, D., C. Dwork, and M. Naor, “Non-malleable cryptography,” Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, pp. 542-552, 1991.

Dwork, C., and M. Naor, “An efficient existentially unforgeable signature scheme and its applications,” Proceedings of CRYPTO ’94, LNCS 839, pp. 234-246, Springer-Verlag, 1994.

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, 1985a.

Europay International S.A., MasterCard International Incorporated, and Visa

International Service Association, “EMV ’96: Integrated Circuit Card Specification for Payment Systems,” July 1996.（http://www.visa.com/nt/chip/download.html）
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.（N. Koblitz 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.
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,” preprint, 1997.