

IMES DISCUSSION PAPER SERIES

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

宇根正志

Discussion Paper No. 99-J-22

IMES

**INSTITUTE FOR MONETARY AND ECONOMIC STUDIES
BANK OF JAPAN**

日本銀行金融研究所

〒100-8630 東京中央郵便局私書箱 203 号

備考：日本銀行金融研究所ディスカッション・ペーパー・シリーズは、金融研究所スタッフおよび外部研究者による研究成果をとりまとめたもので、学界、研究機関等、関連する方々から幅広くコメントを頂戴することを意図している。ただし、論文の内容や意見は、執筆者個人に属し、日本銀行あるいは金融研究所の公式見解を示すものではない。

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

宇根正志*

要 旨

RSA 方式は、素因数分解問題の困難性に依拠した公開鍵暗号方式であり、データ守秘 (RSA 暗号) とデジタル署名 (RSA 署名) の両方の機能を有している。RSA 方式のアルゴリズム自体に対しては、これまで現実的な脅威となる攻撃法が提案されていないことから、安全性の高い公開鍵暗号方式として、金融分野をはじめとする幅広い分野で利用されている。

Gemplus 社 (フランス) の Coron、Naccache、ルーベン・カトリック大学 (ベルギー) の Stern は、1999年4月、RSA 署名に対する新しい攻撃法を発表した。本攻撃法は、一定条件を満足するメッセージの署名を利用して別のメッセージの署名を偽造するというものであり、公開鍵を直接素因数分解するよりも少ない計算量で署名偽造が可能となる。

Coron-Naccache-Stern の攻撃法で注目されるのは、本攻撃法が RSA 署名を利用したデジタル署名方式の国際標準 ISO/IEC 9796-2 に適用できるという点である。本国際標準は、主に IC カード上での実装を想定して策定されており、署名からメッセージを復元できる仕組みとなっている。本攻撃法は、こうしたデジタル署名方式の特徴点を巧みに利用したものであり、RSA 署名のアルゴリズムを直接攻撃するものではない。Coron らの研究成果は、暗号アルゴリズム自体の安全性だけでなく、その利用方法を含めた総合的な評価の必要性を強く示唆するものである。

本稿では、まず、これまでの RSA 署名の安全性に関する主な研究成果を整理し、ISO/IEC 9796 の概要を説明する。その上で、Coron-Naccache-Stern の攻撃法について説明し、情報セキュリティ技術の標準化を担当する ISO/IEC JTC1/SC27 における対応状況や IC カード等他の標準規格への影響について説明する。

キーワード：公開鍵暗号、RSA 署名、デジタル署名、ISO、国際標準

JEL Classification：L86、L96、Z00

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

本論文を作成するに当たっては、神戸大学の田中初一教授、九州大学の櫻井幸一助教授、通信・放送機構の清水秀夫研究員から有益なコメントを頂戴した。

目次

	頁
1. はじめに	1
2. RSA 方式とそのアルゴリズム	2
3. これまでの RSA 署名の安全性に関する研究	3
(1)RSA 署名の分類	3
<1>付録型	3
<2>メッセージ復元型	3
(2)RSA 署名に対する安全性評価結果	4
<1>受動的攻撃に対する安全性	5
<2>能動的攻撃に対する安全性	6
ナイーブな RSA 署名	6
PKCS #1	7
ISO/IEC 9796-1,2,3	8
4. CNS 攻撃法の内容と特徴	12
(1)攻撃法の基本的な手順	12
(2)攻撃法の特徴点	13
(3)攻撃法の適用例	14
5. ISO/IEC 9796-2 に対する攻撃法	15
(1)攻撃の手順	15
<1>公開鍵 n の形態	15
<2>署名者に送付するメッセージの生成	15
<3>署名の偽造	17
(2)攻撃に必要となる計算量とメッセージ数	17
6. ISO/IEC 9796-1 に変更を加えたデジタル署名に対する攻撃法	18
(1)ISO/IEC 9796-1 と変更版 ISO/IEC 9796-1 との差異	18
(2)攻撃の手順	18
<1>攻撃法のアイデア	19
<2>署名者に送付するメッセージの生成	19
<3>署名の偽造	20
(3)攻撃に必要となる計算量とメッセージ数	20
7. RSA 署名を利用した安全性証明付きのデジタル署名方式	22
(1)PSS 署名	22
(2)PSS-R 署名	23
8. CNS 攻撃法による標準化活動等への影響	24
9. おわりに	25
参考文献	26
補論 1 : ISO/IEC 9796-1,2,3 の署名変換対象データのフォーマット	29
1. ISO/IEC 9796-1 の署名変換対象データ	29
2. ISO/IEC 9796-2 の署名変換対象データ	30
3. ISO/IEC 9796-3 の署名変換対象データ	31
補論 2 : CNS 攻撃法の概要	33
1. 攻撃に利用されるメッセージ M の選択	33
2. 署名の偽造方法	35

1. はじめに

RSA 方式は、データ守秘 (RSA 暗号) とデジタル署名 (RSA 署名) の両方の機能を有する公開鍵暗号方式である。RSA 方式の安全性は素因数分解問題の困難性に依拠しており、これまで現実的な脅威となる攻撃法は提案されていない。

こうした中、1999 年 4 月、フランスの IC カード製造業者である Gemplus 社の Coron、Naccache、ベルギーのルーベン・カトリック大学の Stern は RSA 署名に対する新しい攻撃法を発表した。Coron-Naccache-Stern の攻撃法 (以下、CNS 攻撃法) は、署名変換の対象となるデータが小さな素数の積となるメッセージを生成し、そのメッセージの署名を利用して、他のメッセージの署名を偽造するというものである。本攻撃法では、公開鍵を直接素因数分解する攻撃法に比べ、素因数分解の対象となる合成数を小さくすることができるため、攻撃に必要な計算量を大幅に削減できる。本攻撃法は、直ちに現実的な脅威となる訳ではないが、「攻撃者が自分にとって都合のよいメッセージの署名を入手できる」という条件が成立する実装環境においては、その安全性に問題点を生じさせる可能性もある。

CNS 攻撃法において留意すべき点は、RSA 署名を利用したデジタル署名方式の国際標準 ISO/IEC 9796-2 に対して本攻撃法が有効である点である。ISO/IEC 9796-2 は、計算能力が比較的限定されている IC カード上での実装を想定して策定されており、処理対象となるデータ量が少なく、署名からメッセージを復元可能なデジタル署名方式が採用されている。本攻撃法は、こうしたデジタル署名方式の特徴を巧みに利用したものであり、RSA 署名のアルゴリズムを直接攻撃するものではない。情報セキュリティ技術の標準化を担当している ISO/IEC JTC1/SC27 は、CNS 攻撃法への対応方法について現在検討を進めている。ISO/IEC 9796-2 は、EMV'96 をはじめとする IC カードの業界標準に採用されていることから、SC27 の対応次第では、これらの業界標準へも影響が及ぶ可能性がある。

CNS 攻撃法は、暗号方式の評価を行う際には、その利用方法や実装環境を十分考慮した上で総合的な判断を下すことが重要であることを強く示唆するものである。金融分野を含めて、暗号技術を利用した製品やシステムのセキュリティを検討する際には、こうした研究成果を考慮していく必要があるものと思われる。

本稿では、まず、これまでの RSA 署名の安全性に関する研究結果を整理し、ISO/IEC 9796 に規定されているデジタル署名方式について説明する。その上で、CNS 攻撃法の概要や ISO/IEC 9796-2 への適用方法について説明し、これまでの ISO/IEC JTC1/SC27 における CNS 攻撃法への対応状況や CNS 攻撃法が IC カードの業界標準に及ぼす影響について説明する。

2. RSA 方式とそのアルゴリズム

RSA 方式は、1978 年に、Rivest、Shamir、Adleman によって考案された最初の本格的な公開鍵暗号であり、データ守秘 (RSA 暗号) およびデジタル署名 (RSA 署名) の両方に利用可能である。RSA 方式は、大きな 2 つの素数からなる合成数を素因数分解する問題を解くことが計算量的に困難である¹ことに依拠している (Rivest, Shamir and Adleman[1978])。RSA 方式の安全性に関する証明 (例えば、RSA 方式の安全性と素因数分解問題の困難性との等価性) はこれまで示されていないものの、現在まで素因数分解よりも現実的に有効な攻撃法は報告されていない。

このため、RSA 方式は、最も信頼性の高い公開鍵暗号方式として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おり、ISO 等の数多くの国際標準に採用されている²。

RSA 方式における鍵生成、暗号化・復号手順は以下の通り。

< RSA 方式のアルゴリズム >

【鍵生成】2 つの大きな素数 p と q を選び、これらの積 $n=pq$ を計算する。
 $p-1$ と $q-1$ の最小公倍数 $L = \text{LCM}(p-1, q-1)$ を計算する。
最大公約数 $\text{GCD}(e, L) = 1$ を満足する自然数 e を選び、 $ed = 1 \pmod{L}$ を満たす d を求める。

【秘密鍵】 d

【公開鍵】 (e, n)

【暗号化】 $C = M^e \pmod{n}$ (ただし、 M は平文、 C は暗号文)

【復号】 $M = C^d \pmod{n}$

¹ 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、その時々技術条件等によって左右される。

² RSA 方式を利用している標準規格としては、ISO/IEC 9796 のほか、署名対象メッセージに署名を添付するタイプのデジタル署名方式を規定する国際標準 ISO/IEC 14888 (Information technology Security techniques Digital signature schemes with appendix, ISO/IEC[1998]) や、金融業務における公開鍵暗号を利用した鍵管理方法の国際標準 ISO 11166-2 (Banking Key management by means of asymmetric algorithms Approved algorithms using the RSA cryptosystem, ISO[1994]) が挙げられる。また、米国政府内で利用されるデジタル署名方式を規定する FIPS 186-1 (NIST[1999]) や、米国銀行業界における公開鍵暗号方式に関する標準規格 ANSI X9.31 (ANSI[1998]) においても RSA 署名が採用されている。

3. これまでの RSA 署名の安全性に関する研究

(1)RSA 署名の分類

RSA 署名を利用したデジタル署名方式は、「付録型」と「メッセージ復元型」の2つに分類することができる³。各方式の内容は以下の通り。

<1>付録型

付録型のデジタル署名方式は、署名生成者が署名をメッセージに添付して送信し、署名検証者は送付されたメッセージを利用して署名の正当性を検証する方式である。

付録型におけるナイーブな方式は、メッセージのハッシュ値を生成し、そのハッシュ値を RSA 署名の秘密鍵 d で変換することで署名を生成するというものである。ナイーブな方式における署名生成、検証手順は以下の通り。

【署名生成】メッセージ M のハッシュ値 $m = H(M)$ 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 を生成する。

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

S は M に添付されて受信者に送付される。

【署名検証】受信者は、以下の等式が成立するか否かを検証する。

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

また、RSA 署名による署名変換の対象となるデータ (署名変換対象データ) として、単にメッセージのハッシュ値を利用するのではなく、メッセージになんらかの変換を加えたデータ、乱数、予め決められたパディングデータ等を結合させて署名変換対象データを生成する方式も提案されている。こうした方式としては、PKCS #1 や PSS 署名等が挙げられる。

<2>メッセージ復元型

メッセージ復元型は、署名からメッセージ全体または一部を復元可能とする方式であり、署名検証の際にはメッセージを必要とせず、署名変換対象データのフォーマットの正当性によって検証を行うものである。

RSA 署名を利用したメッセージ復元型の署名方式としては、ISO/IEC 9796-1,2,3 や PSS-R 署名等が挙げられる。

³ 本分類方法については、Menezes, Oorschot and Vanstone[1997] p. 427 を参照。

(2)RSA 署名に対する安全性評価結果

デジタル署名方式の安全性を評価する場合には、どのような攻撃に対する安全性を考えるかを明確にする必要がある。攻撃法を大きく 2 つに分類すると、以下の表 1 の通り。

表 1 デジタル署名方式に対する攻撃法の分類

	攻撃法の内容
受動的攻撃	公開鍵と、攻撃者が選択できないメッセージに対する署名を利用した攻撃法
能動的攻撃	公開鍵と、攻撃者が選択したメッセージに対する署名を利用した攻撃法

これらの攻撃法は、攻撃者が入手可能な署名に対するメッセージを、攻撃者自ら選択することが可能か否かによって分類される。受動的攻撃は、実現可能性が高いものの、攻撃者が利用できる情報は制限されている。一方、能動的攻撃は、攻撃者が自分の選んだメッセージに対する署名を入手できるものの、受動的攻撃と比べて実現可能性は低い。

デジタル署名方式のタイプと攻撃のタイプに沿って、RSA 署名を利用した主なデジタル署名方式を分類し、これまでの安全性評価を整理すると表 2 の通り⁴。現時点で安全性が証明されている方式は、PSS 署名と PSS-R 署名だけである。以下では、安全性が証明されていない方式について説明する（PSS 署名や PSS-R 署名については 6.において説明）。

表 2 RSA 署名を利用した主なデジタル署名方式の安全性評価結果

デジタル署名方式		受動的攻撃に対する安全性	能動的攻撃に対する安全性
付録型	ナイーブな RSA 署名		Desmedt-Odlyzko の攻撃法
	PKCS #1		
	PSS 署名		
メッセージ復元型	ISO/IEC 9796-1		⁵
	ISO/IEC 9796-2		CNS 攻撃法
	ISO/IEC 9796-3		Misarsky の攻撃法
	PSS-R 署名		

(注) : 安全性が証明されている。

: 安全性は証明されていないものの、有効な攻撃法が提案されていない。

⁴ RSA 署名の安全性に関するこれまでの研究に関しては、Kaliski and Robshaw[1995]や Misarsky[1998]を参照。

⁵ Coron らは、ISO/IEC 9796-1 を一部変更したデジタル署名方式に対して有効な攻撃法を提案している（Coron et al.[1999]）。

<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、いずれのタイプのデジタル署名方式に対しても、現在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。RSA 署名の安全性は、 n の素因数分解に必要となる計算量によって左右される。これまでに提案されている主要な素因数分解アルゴリズムを整理すると、以下の表 3 の通り⁶。

表 3 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 $n(=p \cdot q)$ の性質	最も効率的な解法
$p \pm 1$ または $q \pm 1$ 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

表 3 は、合成数 n が 2 つの素数 p と q の積になっている場合、 p と q がどのような条件の場合にどの解法が最も高速かを整理したものである。 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。

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 2 月には、数体ふるい法を用いることにより、約 2 か月間を費して 465 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、計算量は約 2000 MIPS 年⁷）。こうしたことから、現時点では、付録型のデジタル署名方式に関する国際標準 ISO/IEC 14888-3 (ISO/IEC[1998]) や金融機関において利用される情報セキュリティ技術に関するガイドライン ISO/TR 13569 (ISO[1997]) 等において、RSA 方式等、素因数分解問題を利用する公開鍵暗号方式においては、1024 bit 以上の鍵長を利用することが推奨されている。

⁶ 素因数分解アルゴリズムについては、宇根・岡本[1999b]を参照。

⁷ MIPS 年：1 MIPS は 1 秒間に 100 万回の命令を実行できる計算能力を表しており、2000 MIPS 年は、1 MIPS の計算能力を有する計算機によって 2000 年間で実行される計算量を表す。

<2>能動的攻撃に対する安全性

ナイーブな RSA 署名

ナイーブな RSA 署名の場合、Desmedt-Odlyzko の攻撃法によって、ある一定の条件を満足するメッセージの署名を偽造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、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。本攻撃法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。

(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可能となるメッセージ 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 を大量に生成し、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いるため、入手した複数の署名によって、各素数 p_i を d 乗して $\bmod n$ を計算した値 ($p_i^d \bmod n$) を得る。

(iv) 攻撃者は、入手した $p_i^d \bmod n$ の値を蓄積し、それらを組み合わせることによって、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ことができる。

本攻撃法によって、ナイーブな RSA 署名は能動的攻撃に対して安全性を確保することができないことが示された。このため、能動的攻撃への対応方法として、署名変換対象データの生成にメッセージのハッシュ値を利用するという方法と、署名変換対象データに冗長性をもたせるという方法が提案された (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、後述する PKCS #1 や ISO/IEC 9796-1,2,3 等の方式である。

PKCS #1

PKCS #1⁸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。PKCS #1 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 の署名生成方法 >

【公開鍵】(e,n) (n は k byte)

【秘密鍵】d

【署名生成手順】

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 $H = \text{Hash}(M)$ 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 (サイズは t byte) を生成する。

<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。

<4>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⁹。

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

<5>署名生成者は、上記の SR を秘密鍵 d で変換して署名 S を生成する。

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

現在利用されている PKCS #1 Version 2.0 (RSA Laboratories[1998]) では、一定条件の下で安全性が証明されている OAEP (Optimal Asymmetric Encryption Padding) が採用されている (Bellare and Rogaway[1995])¹⁰。一方、デジタル署名方式の安全性は証明されていない。

⁸ 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。

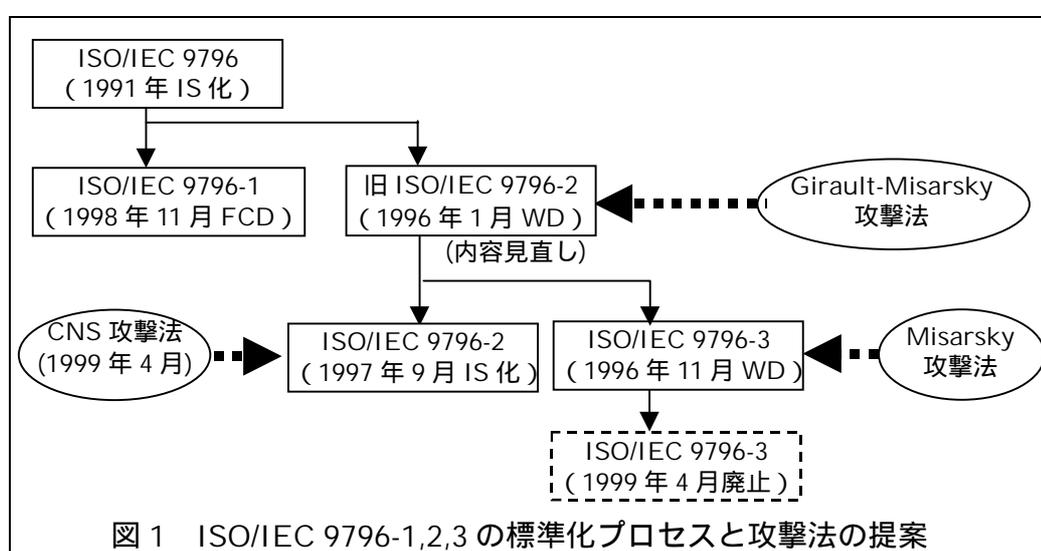
⁹ 以下の表記では、「 01_{16} 」の「16」の添え字は byte 表示であることを表し、添え字がない場合は bit 表示であることを表す。すなわち、「 01_{16} 」は bit 表示では「0000 0001」となる。

¹⁰ PKCS #1 Version 2.0 の前バージョン Version 1.5 (RSA Laboratories[1993]) では、データ守秘方式として OAEP とは別の方式が採用されていた。しかし、1998 年 6 月、本方式に対して能動的攻撃が適用可能であることが Bleichenbacher によって示されたほか、PKCS #1 を実装している SSL Version 3 において実際に能動的攻撃が可能になることが実証された (Bleichenbacher[1998])。このため、RSA 社は、1998 年 9 月に OAEP をデータ守秘方式とする PKCS #1 Version 2.0 を発表した。

PKCS #1 Version 2.0 に規定されているデジタル署名方式に対して、これまでに素因数分解よりも有効な攻撃法は提案されていない。

ISO/IEC 9796-1,2,3

ISO/IEC 9796 (Information technology Security techniques Digital signature schemes giving message recovery) は、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、メッセージを署名に埋め込むことによって送信されるデータ量を削減することが可能となっており、計算能力が比較的限定される IC カード上での実装に適した方式である。



ISO/IEC 9796 は、以下の 4 つのパートから構成されている。

- Part 1 : 冗長性を利用した方式
- Part 2 : ハッシュ関数を利用した方式
- Part 3 : 検証関数を利用した方式
- Part 4 : 離散対数問題を利用した方式¹¹

このうち、Part 1~3 のデジタル署名方式に RSA 署名が利用されており、これらのデジタル署名方式の標準化の流れは図 1 の通り(Girault and Misarsky[1997], Misarsky[1997] and [1998])。

¹¹ Part 4 は、Part 1,2,3 とは異なる経緯の中で標準化が進められているものであり、本稿では取り上げない。なお、ISO/IEC JTC1/SC27/WG2 は、Part 3 として標準化が進められていた国際標準案の廃止(後述)を決定しており、Part 4 を新たに Part 3 として標準化する方向で検討を進めている。

(a) ISO/IEC 9796-1 のデジタル署名方式

ISO/IEC 9796-1 (Information technology Security techniques Digital signature schemes giving message recovery Part 1: Mechanisms using redundancy) は、1991年に国際標準となった ISO/IEC 9796 (Information technology Security techniques Digital signature schemes giving message recovery) のデジタル署名方式のリバイス版として提案されている国際標準案であり、署名生成・検証方法は同一のものである (ISO/IEC [1991]、Coron et al.[1999])¹²。

本国際標準案に規定されているデジタル署名方式は、ハッシュ関数を利用しない方式であり、署名変換対象データにメッセージ全体を埋め込み、署名からメッセージ全体を復元可能となるように設計されている (署名変換対象データのフォーマットについては補論 1 参照)。公開鍵のサイズが 1024 bit の場合、256 bit 以下の比較的短いメッセージに対してのみ利用可能となっている。

(b) ISO/IEC 9796-2 のデジタル署名方式

ISO/IEC 9796-2 (Information technology Security techniques Digital signature schemes giving message recovery Part 2: Mechanisms using hash-function) は、署名からメッセージ全体または一部を復元可能なデジタル署名方式の国際標準であり、1996年1月に SC27 において標準化が開始され、1997年9月に完了している (ISO/IEC[1997])。

標準化が開始された当初、WD¹³として提案された ISO/IEC 9796-2(以下、旧 ISO/IEC 9796-2 と呼ぶ) には、署名検証のために剰余演算を利用した関数 ($H(M)=2(M \bmod (2^{79}+1))$) が用いられていたが、この関数の性質を利用した攻撃法がフランステレコム の Girault と Misarsky によって発表された (Girault and Misarsky [1997])。このため、旧 ISO/IEC 9796-2 で利用されていた剰余演算の関数の代わりにハッシュ関数が採用され、新しい ISO/IEC 9796-2 の標準化が進められた。

¹²ISO/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])。

¹³WD (Working Draft): 新しい国際標準の提案 (New work item Proposal) が ISO の専門委員会または分科委員会において行われ、それが投票によって承認された後、作業グループが作成した国際標準規格の作業原案。ISO における国際標準化プロセスの詳細については、岩下・谷田部[1999]を参照。

ISO/IEC 9796-2 では、ハッシュ関数を利用することにより、ISO/IEC 9796-1 と比べて利用可能なメッセージサイズを拡大することができる。公開鍵のサイズが 1024 bit の場合、署名から復元可能なメッセージは最大 848 bit となる（署名変換対象データのフォーマットについては補論 1 参照）。

(c) ISO/IEC 9796-3 のデジタル署名方式

ISO/IEC 9796-3 (Information technology Security techniques Digital signature schemes giving message recovery Part 3: Mechanisms using a check-function) は、旧 ISO/IEC 9796-2 に対する攻撃法として発表された 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たデジタル署名方式の国際標準案である（Misarsky [1998]）。本国際標準案は、1996 年 11 月に WD として SC27 に提案された。

本デジタル署名方式における署名変換対象データの生成方法の特徴は、 $C = M \bmod (2^{79} + 1)$ という検証式によって生成される検証コード C を複数の比較的小さなデータに分割し、署名変換対象データの特定の位置に埋め込む点である（署名変換対象データのフォーマットについては補論 1 参照）。検証コードの埋め込みによって、署名変換対象データが複数の部分に分割され、Girault-Misarsky の攻撃法に対する安全性が高まるとされている。

しかし、Misarsky は、本デジタル署名方式に対する攻撃法を発表し、一定の条件¹⁴の下で、公開鍵の素因数分解よりも効率的に署名を偽造可能であることを示した（Misarsky [1997]）。本攻撃法のアイデアは以下の通り。

- (i) メッセージ M に対する署名変換対象データを $U(M)$ とする。攻撃者は、署名偽造の対象であるメッセージ M に対して、 $U(M) \cdot U(X) = U(Y) \pmod{n}$ を満足するデータ X と Y を見つける。
- (ii) 攻撃者は、 X と Y を正当な署名者に送付し、署名者から X と Y に対する署名 $U(X)^d \bmod n$ と $U(Y)^d \bmod n$ を入手する。
- (iii) $U(M) \cdot U(X) = U(Y) \pmod{n}$ を変形すると、 $U(M)^d \cdot U(X)^d = U(Y)^d \pmod{n}$ となり、 $U(M)^d = (U(Y)^d / U(X)^d) \pmod{n}$ が成立する。このため、攻撃者は入手した 2 つの署名を上記等式の右辺に代入し、メッセージ M に対する署名 $U(M)^d \bmod n$ を偽造する。

¹⁴ Misarsky の攻撃法を ISO/IEC 9796-3 に適用するための条件は、公開鍵のサイズを N 、メッセージのサイズを m とする場合、 $(N/2) - 80 < m$ が成立することである。例えば、公開鍵のサイズが 1024 bit の場合には、メッセージのサイズが 433 bit 以上のときに本攻撃法が適用可能となる。

Misarsky は、本攻撃法を ISO/IEC 9796-3 に適用した場合（公開鍵のサイズを 640 bit、メッセージのサイズを 512 bit に設定）、Pentium 166MHz のパソコンを利用して約 30 分で署名の偽造に成功したとの実験結果を発表している（Misarsky[1997]）。

本攻撃法が発表されたことを受けて、ISO/IEC JTC1/SC27 は、1999 年 4 月に ISO/IEC 9796-3 の標準化作業を中止し、ISO/IEC 9796-3 を廃止することを決定している。

4. CNS 攻撃法の内容と特徴

(1) 攻撃法の基本的な手順

CNS 攻撃法は、署名変換対象データが小さな素数の積となるメッセージを大量に集め、それらのメッセージを正当な署名者に送信して署名を生成してもらい、入手した署名から別のメッセージの署名を偽造するというものである (Coron et al.[1999])¹⁵。

Coron らは、本攻撃法を ISO/IEC 9796-2 に適用可能であり、公開鍵が 1024 bit の場合には、必要となる署名を入手した上で、176 bit の合成数の素因数分解を実行することによって、別のメッセージの署名を偽造できることを示している。

本攻撃法のアイデアを説明すると以下の通り (詳しい説明は補論 2 を参照)。

< 以下の説明に利用される記号の意味 >

- U : メッセージ M から署名変換対象データ $U(M)$ を生成する関数。 $U(M)$ のサイズは公開鍵のサイズと同一となる。
- L : 攻撃の際に素因数分解が必要となる数値のサイズ (bit 数)。 ISO/IEC 9796-2 への攻撃の場合には 176 bit となる。
- p_k : 最小の素数 2 から順番に数えて k 番目に大きな素数
- (n, e) : RSA 署名の公開鍵 (ただし $n = p \cdot q$ 、 p と q は素数)
- d : RSA 署名の秘密鍵

攻撃者は p_k を設定した上で、署名変換対象データ $U(M)$ が p_k -smooth¹⁶ となるメッセージ M を大量に集める。

攻撃者は選択したメッセージ M を正当な署名者に送信し、各 M に対する署名 $U(M)^d \bmod n$ を返信してもらう。

攻撃者は入手した $U(M)^d \bmod n$ を利用して、メッセージ M' に対する署名 $U(M)^d \bmod n$ を生成する。ただし、 M' は、 $U(M')$ が p_k -smooth となるようなメッセージである。

¹⁵ ベキ乗剰余演算によって変換されたデータを解読ないしは偽造する際に、そのデータが比較的小さな素数の積となっていることを利用するアイデアとしては、1988 年に神戸大学の田中教授が発表した ID 情報に基づく鍵配送方式に対する攻撃法が挙げられる (田中 [1988])。田中教授は、同論文の中で、本攻撃法が RSA 公開鍵暗号系にも適用可能であることを指摘しているが、このアイデアに基づいた具体的な攻撃法はこれまで提案されていなかった。

¹⁶ p_k -smooth: 全ての素因数が p_k 以下となる合成数を「 p_k -smooth」という。 p_k -smooth な合成数 u は、

$$u = \prod_{j=1}^k p_j^{v_j}$$

と表される。ただし、 $p_j \leq p_k$ ($j=1, \dots, k$) である。

(2)攻撃法の特徴点

CNS 攻撃法の主な特徴点をまとめると、以下の4点。

CNS 攻撃法は能動的攻撃である。

本攻撃法は、「攻撃者が自分の都合のよいメッセージに対する署名を入手できる」という実装環境において有効であり、本攻撃法が現実的な脅威となり得るかは攻撃対象となるデジタル署名方式の実装環境に依存する。

署名を偽造することができるメッセージには制約が存在し、攻撃者が署名を偽造するメッセージを自由に選択することはできない。

本攻撃法によって署名を偽造可能なメッセージ M は、 M の署名変換対象データ $U(M)$ が p_k -smooth となる必要があるため、偽造対象のメッセージを自由に選択することはできない。ただし、本攻撃法によって意味のあるメッセージに対する署名を偽造できる可能性は残されていることから、本攻撃法が現実的な脅威となる可能性もある。

攻撃に必要な計算量のオーダーは、

$$O\left(\frac{Lk(\log e)\sqrt{k \ln k}}{\rho(L/\log_2(k \ln k))}\right)$$

となる。また、必要なメッセージ数のオーダーは $O(k(\log e))$ となる。

攻撃に必要な計算量は、 L 、 k 、公開鍵 e に依存する一方、公開鍵 n のサイズには依存しない。このため、法 n のサイズを大きくしても、本攻撃法に対する安全性は向上しない。ISO/IEC 9796-2 に本攻撃法を適用する場合、SHA-1 の利用を前提としたときには176 bitの合成数の素因数分解によって攻撃可能となり、1024 bitの公開鍵を素因数分解する攻撃法に比べて大幅に計算量を削減することができる。

攻撃に必要なメッセージ数も k と e のみに依存することから、公開鍵のサイズに依存しない。なお、最も単純なケースとして e が素数の場合、攻撃に必要なメッセージ数は $k+1$ 個となる。

予め設定された k と e に対して、攻撃に必要な数の署名を入手することができれば、 $U(M)$ が p_k -smooth となるメッセージ M に対する署名を、複数偽造することができる。

ただし、ISO/IEC 9796-2 に本攻撃法を適用する場合には、攻撃に利用する M に公開鍵 n の一部が含まれているため、 n が変更されると、それまでに入手していた M に対する署名を使って、追加的に別の新たな署名を偽造することはできなくなる。

(3) 攻撃法の適用例

Coron らは、本攻撃法が ISO/IEC 9796-2 に適用可能であることを示したほか、署名変換対象データの生成方法に一部修正を加えた ISO/IEC 9796-1 (詳細な説明については 6. を参照) や、PKCS #1 Version 2.0、ANSI X9.31 に対して、本攻撃法を適用した場合の必要計算量やメッセージ数を試算している。

ただし、署名変換対象データの生成方法に何ら変更を加えない ISO/IEC 9796-1 に対しては、本攻撃法を適用することができない (Coron et al.[1999])。

また、PKCS #1 Version 2.0 や ANSI X9.31 に対して本攻撃法を適用できるのは公開鍵 n が特殊な形態をしている場合のみであり、たとえ適用できたとしても、攻撃に必要となる計算量は素因数分解よりも多くなる。このため、これらのデジタル署名方式は、本攻撃法に対して十分な安全性を有している (Silverman and Naccache[1999])。

Coron らは、論文 (Coron et al.[1999]) の中で、「PKCS #1 Version 2.0 や ANSI X 9.31 のデジタル署名方式に対しては、本攻撃法は、公開鍵 n が $n=2^k \pm c$ となる特殊なケースにおいてのみ適用できる。したがって、これらのデジタル署名方式を実装する際に、 $n=2^k \pm c$ となる公開鍵を利用しないようにすることで本攻撃法に対して安全性を確保できる。なお、ANSI X9.31 には、既に $n=2^k \pm c$ となる特殊な公開鍵の利用を避けるように明記されている」と指摘している。

5. ISO/IEC 9796-2 に対する攻撃法

以下では、公開鍵およびメッセージのサイズが 1024 bit、ハッシュ関数として SHA-1 (ハッシュ値が 160 bit) を利用するケースを例に、ISO/IEC 9796-2 に対する攻撃法を説明する (Coron et al.[1999])。

(1) 攻撃の手順

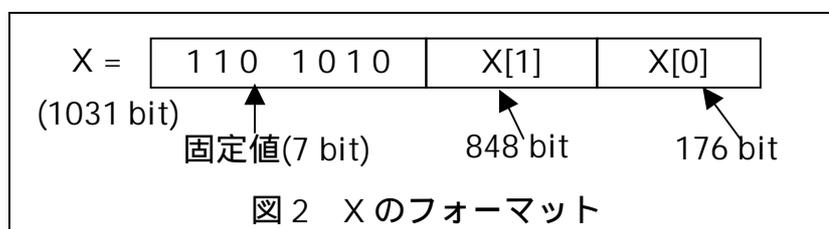
<1> 公開鍵 n の形態

本攻撃法のアイデアは、「公開鍵 n と署名変換対象データ $U(M)$ の関数 $F(n, U(M))$ を用いて、 $U(M)$ から素因数分解が比較的容易な小さな合成数を生成し、その合成数が p_k -smooth となるメッセージ M を利用する」というものである。

ISO/IEC 9796-2 の署名変換対象データ $U(M)$ は、左から 7 bit 分が "110 1010" となる。仮に公開鍵 n の左から 7 bit 分が "110 1010" となる場合 (確率は $1/2^7$)、 M の左から 848 bit 分を一定の値に定めた上で、関数 $F(n, U(M)) = n - 2U(M)$ を利用することによって、素因数分解の対象となるデータ $n - 2U(M)$ のサイズをハッシュ値のサイズと同程度の 169 bit にすることができる。

また、 n の左 7 bit 分が "110 1010" となっていない場合には、適当な数値 a を選択して n を a 倍することにより、左から 7 bit 分が "110 1010" となる 1031 bit (= 1024 bit + 7 bit) のデータ $X (= a \cdot n)$ を生成することができる。その上で、 M の左から 848 bit 分を一定の値に定め、関数 $F(n, U(M)) = a \cdot n - 2^8 \cdot U(M) = X - 2^8 \cdot U(M)$ を利用することによって、素因数分解の対象となるデータ $X - 2^8 \cdot U(M)$ のサイズを 176 bit にすることができる。

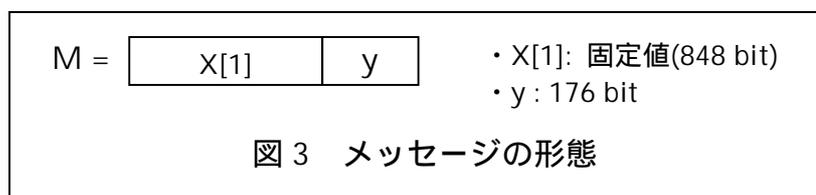
X のサイズは 1031 bit であり、そのフォーマットは以下の図 2 の通り。 X の右から 176 bit の部分を $X[0]$ とし、左 7 bit "110 1010" と $X[0]$ の間の部分 (848 bit) を $X[1]$ とする。



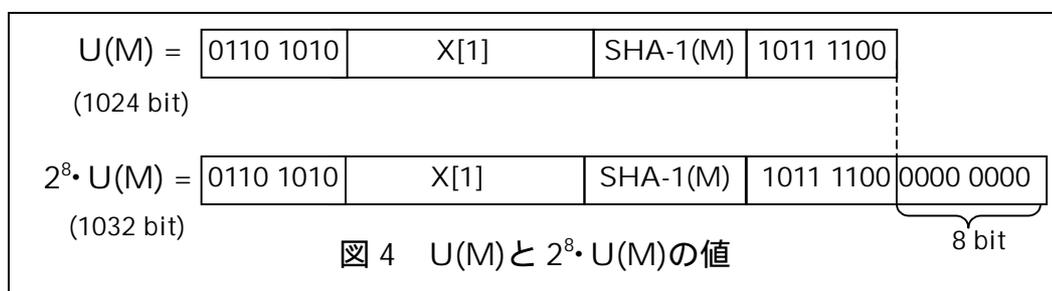
<2> 署名者に送付するメッセージの生成

攻撃に利用するメッセージ M は、左 848 bit の部分が $X[1]$ と等しくなるように設定される。残りの 176 bit を y とする。 M の取り得る値の総数は 2^{176}

個となる (図 3 参照)。



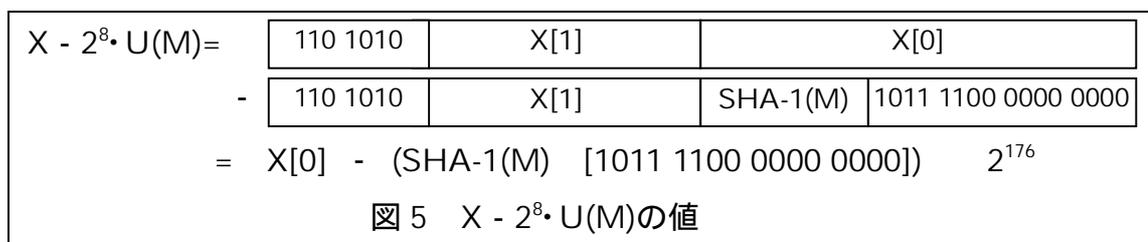
このような M の署名変換対象データ U(M)を計算し、さらに $2^8 \cdot U(M)$ を計算すると図 4 の通り (詳細については補論 1 参照)。 $2^8 \cdot U(M)$ は U(M)の各 bit が 8 つ左にシフトし、右から 8 bit だけ"0"が付け加えられる。 $2^8 \cdot U(M)$ のサイズは 1032 bit となる。



< $2^8 \cdot U(M)$ の構成>

- [Header, More-data bit, Padding field]=[0110 1010] (8 bit)
- [Data field] = X[1] (848 bit)
- [Hash field] = SHA-1(M) (160 bit)
- [Trailer] = [1011 1100] (8 bit)
- 追加されるデータ : [0000 0000] (8 bit)

次に $(X - 2^8 \cdot U(M))$ を計算する。 $2^8 \cdot U(M)$ の左 1 bit は"0"であることから、最初の 855 bit のデータ([110 1010] X[1])がキャンセルアウトされるため、計算結果である $(X[0] - (SHA-1(M) [1011 1100 0000 0000]))$ は 2^{176} 以下となる (図 5 参照)。



こうして得た $(X - 2^8 \cdot U(M))$ が p_k -smooth となるように y を選ぶ。

$$X - 2^8 \cdot U(M) = \left(\prod_{j=1}^k p_j^{v_j} \right)$$

選択した y に対して $(X - 2^8 \cdot U(M))$ が p_k -smooth となっていることを確認するためには、 $(X - 2^8 \cdot U(M))$ を実際に素因数分解する必要があるが、 $(X - 2^8 \cdot U(M))$ のサイズ(L の値)は 176 bit 以下であるため、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。

こうして入手した M を署名者に送付し、 M に対応する署名 $U(M)^d \bmod n$ を入手する。

<3>署名の偽造

入手した署名 $U(M)^d \bmod n$ を利用して、別のメッセージ M' に対応する署名 $U(M')^d \bmod n$ を偽造する (詳細は補論 2 参照)。

(2)攻撃に必要となる計算量とメッセージ数

本攻撃法において必要な計算量やメッセージ数は、 k および e に依存する¹⁷。公開鍵 n のサイズを変化させたとしても、 $(X - 2^8 \cdot U(M))$ を計算することによって、素因数分解が必要となる合成数 $(X - 2^8 \cdot U(M))$ のサイズは常に 176 bit 以下となることから、本攻撃法に必要な計算量は公開鍵のサイズに依存しない。

ISO/IEC 9796-2 への攻撃は、 M の中に公開鍵の一部が含まれていることから、公開鍵が変更されると、それ以前に生成していた M を別の署名の偽造に利用することができなくなる。このため、本攻撃法に対する運用面での対応策として、公開鍵を頻繁に変更するという方法が有効であると考えられる。

Coron らは、本攻撃法を ISO/IEC 9796-2 のデジタル署名方式に適用する場合に必要な計算量とメッセージ数を試算している。その結果、SHA-1 を利用した場合、 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 (表 4 参照)。

表 4 CNS 攻撃法を ISO/IEC 9796-2 に適用した場合の必要計算量の試算

ハッシュ長 (bit)	計算量が最小となる k のサイズ (bit)	必要計算量 のオーダー	必要メッセージ数 のオーダー
128	18	2^{54}	2^{36}
160	20	2^{61}	2^{40}

(出典)Coron et al.[1999]

¹⁷ L の値は、ハッシュ値が 160 bit であることから、176 bit で固定されている。

6. ISO/IEC 9796-1 に変更を加えたデジタル署名に対する攻撃法

Coron らは、CNS 攻撃法が、ISO/IEC 9796-1 に変更を加えたデジタル署名方式（以下、変更版 ISO/IEC 9796-1 と呼ぶ）に対しても適用可能であることを示している（Coron et al.[1999]）。変更版 ISO/IEC 9796-1 に対する攻撃法は、ISO/IEC 9796-1 の安全性を評価する上で有用と考えられるため、以下にその概要を説明する。

(1)ISO/IEC 9796-1 と変更版 ISO/IEC 9796-1 との差異

ISO/IEC 9796-1 では、署名変換対象データを生成する際に、一定のルールに基づいて特定の bit の値を反転させる（1 との排他的論理和を計算する）演算が利用されている（詳細については補論 1 を参照）。Coron らの攻撃は、この bit 反転を行わないようにした変更版 ISO/IEC 9796-1 に対してのものである。

例えば、公開鍵のサイズが 1024 bit、メッセージのサイズが 256 bit の場合、この変更により、署名変換対象データの 2 bit 分の値に差異が発生することになる（図 6 参照）。

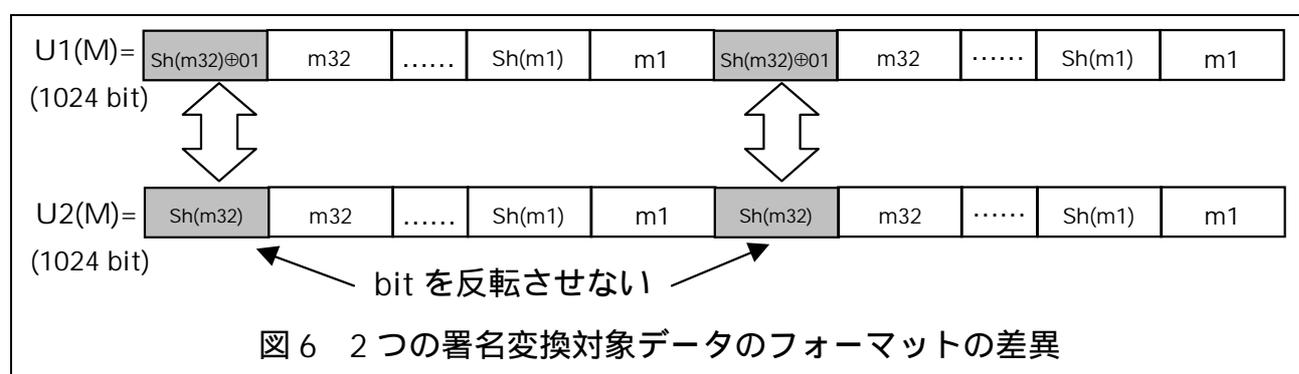


図 6 の例では、メッセージ M は $M=[m_{32}, m_{31}, \dots, m_2, m_1]$ となっており、各 m_i ($i=1, \dots, 32$) は 8 bit のデータである。また、U1(M)が ISO/IEC 9796-1 における署名変換対象データ、U2(M)が変更版 ISO/IEC 9796-1 の署名変換対象データである。なお、図 6 において明示していないが、U1(M)および U2(M)ともに、左から 1 bit が“1”に固定されるほか、右から 4 bit が“0110”に固定される。Sh は 8 bit 入出力の換字変換である。

(2)攻撃の手順

以下では、公開鍵のサイズが 1024 bit、メッセージのサイズが 256 bit の場合について説明する。

<1>攻撃法のアイデア

変更版 ISO/IEC 9796-1CNS 攻撃法を適用する場合には、署名変換対象データの bit パターンが、ほぼ同一の bit パターンの繰り返しとなる点を利用する。具体的には以下の手順で攻撃を実行する。

(a) 以下の 2 つの条件を満足するようにメッセージ M のフォーマットを決める。

(i)署名変換対象データ U(M)の bit パターンが、あるデータ m の bit パターンの繰り返しとなる（すなわち、 $U(M)=[m,m,\dots,m]$ となる）

(ii)繰り返されるデータ m のサイズは、m の素因数分解が容易に実行できるように小さくなる。

(b)m が p_k -smooth となる M の値を見つける（m のサイズは素因数分解可能なほど小さいため、m が p_k -smooth となっていることを確認することは容易）。M を署名者に送付してその署名を入手し、別のデータの署名偽造に利用する。

<2>署名者に送付するメッセージの生成

攻撃に利用するメッセージ M のフォーマット（図 7 参照）は、上記(a)の 2 つの条件を満足するように、32 bit データ MS (= [A, B, C, 0110 0110]) を 8 個結合したものとする（A、B、C は各々 8 bit のデータであり、攻撃者が選択可能）。ただし、A を選ぶ際には、「Sh(A)の左から 1 bit が常に"1"となる」という条件が加わる。



この結果、署名変換対象データ U(M)は、データ m が 16 個結合する構造となる（図 8 参照、詳細については補論 1 参照）。m は 64 bit であり、U(M)は 1024 bit である。



この結果、 $U(M)$ は m によって、 $U(M) = m \times \Gamma$ と表される。ただし、

$$\Gamma = 2^{15 \times 64} + 2^{14 \times 64} + \dots + 2^{2 \times 64} + 2^{64} + 1 = \sum_{i=0}^{15} 2^{64i}$$

次に、 m が p_k -smooth となるように A 、 B 、 C の値を定める。 m が p_k -smooth となっていることを確認するためには m の素因数分解を行う必要があるが、 m のサイズ (L の値) は 64 bit であり、公開鍵のサイズ 1024 bit に比べて非常に小さいため、 m の素因数分解は比較的容易に実行できる。

m が p_k -smooth となるような M をみつけると、 $U(M)$ は以下の等式を満足する。

$$U(M) = \left(\prod_{j=1}^k p_j^{v_j} \right) \times \Gamma$$

攻撃者は、こうして入手したメッセージ M を署名者に送付し、署名 $U(M)^d \pmod n$ を入手する。

<3>署名の偽造

署名 $U(M)^d \pmod n$ を利用して、別のメッセージ M' に対応する署名を偽造する (補論 2 参照)。ただし、 M' は $U(M')$ が以下の形式を満足するものに限定される。

$$U(M') = \left(\prod_{j=1}^k p_j^{v_j} \right) \times \Gamma$$

(3)攻撃に必要となる計算量とメッセージ数

本攻撃法では、素因数分解が必要なデータのサイズ (L の値) が 64 bit となり、必要となる計算量やメッセージ数は k や e によって変化する。公開鍵 n のサイズが変化したとしても、 $U(M)$ に含まれる m の数が変化するだけであり、 m のサイズは 64 bit で変化しない。このため、公開鍵のサイズは本攻撃法に必要となる計算量に影響を与えない。

Coron らは、本攻撃法による署名を実際に偽造する実験を行っている。その結果、Pentium 200MHz を搭載したパソコンを約一日費して、 $U(M)$ が p_k -smooth となるメッセージ M を 15830 個のみつけ、12829 個のメッセージに対する署名の偽造に成功している（表 5 参照）¹⁸。

表 5 CNS 攻撃法による署名偽造実験の結果

k の値	攻撃に利用した メッセージ数	偽造した 署名数
345	346	1
500	799	298
1000	3203	2202
1500	6198	4697
2000	9344	7343
2500	12555	10054
3000	15830	12829

(出典) Coron et al.[1999]

¹⁸ なお、Coron らの論文 (Coron et al.[1999]) には、本実験において利用された公開鍵の一部 e について詳細な情報が記載されていないものの、 $k=345$ の場合、 $346(=k+1)$ 個のメッセージによって 1 つの署名を偽造することが可能となっていることから (表 5 参照)、必要なメッセージ数は $k+1$ となっていることがわかる。

7. RSA 署名を利用した安全性証明付きのデジタル署名方式

ISO/IEC 9796-2 に規定されているデジタル署名方式は安全性が証明されておらず、実際に CNS 攻撃法が適用可能となった。しかし、近年、RSA 署名を利用したデジタル署名方式の中でも、安全性が証明されている方式がいくつか提案されている¹⁹。付録型のデジタル署名方式として PSS 署名が提案されているほか、メッセージ復元型として PSS-R 署名が提案されている。

(1) PSS 署名

PSS (Probabilistic Signature Scheme) 署名は、Bellare と Rogaway によって提案された方式である (Bellare and Rogaway[1996])。PSS 署名は、RSA 署名を利用した付録型の方式であり、以下の 2 つの仮定の下で、「能動的攻撃が可能な環境であったとしても、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。

仮定 1 : 入力値に対してランダムな数値を出力する理想的なランダム関数が利用できる (ランダムオラクルモデル)²⁰

仮定 2 : RSA 暗号関数が一方向性を有している²¹

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 (図 9 参照)。このように、署名変換対象データを左 1 bit を残して完全にランダム化することによって、能動的攻撃に対して攻撃の手掛かりを与えないように設計されている。

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、RSA 署名と同程度の実用性を有している。

¹⁹ 安全性が証明されている公開鍵暗号方式については、宇根・岡本[1999a]を参照。

²⁰ ランダム関数は仮想的な仮定であり、ランダム関数をハッシュ関数で代用した場合には証明が成立しなくなるという問題が残されている。

²¹ 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とき、「RSA 暗号関数は一方向性を有する」と呼ばれる。本命題が正しいか否かについては、現時点では証明されていない。

【鍵生成】秘密鍵と公開鍵の生成方法は RSA 方式と同じ。

- ・ 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)

【署名生成】 平文を 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。

$$w = h(M || r), \quad R = g_1(w) \oplus r$$

$$S = X^d \bmod n \quad (\text{ただし、} X = (0 || w || R || g_2(w)))$$

【署名検証】 $X = S^e \bmod n$ を計算し、 $X = (b || y || z || \dots)$ とする (b : 1 bit、 y : k_1 bit、 z : k_0 bit、 \dots : 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。

$$h(M || R') = y, \quad g_2(y) = z, \quad b = 0$$

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。

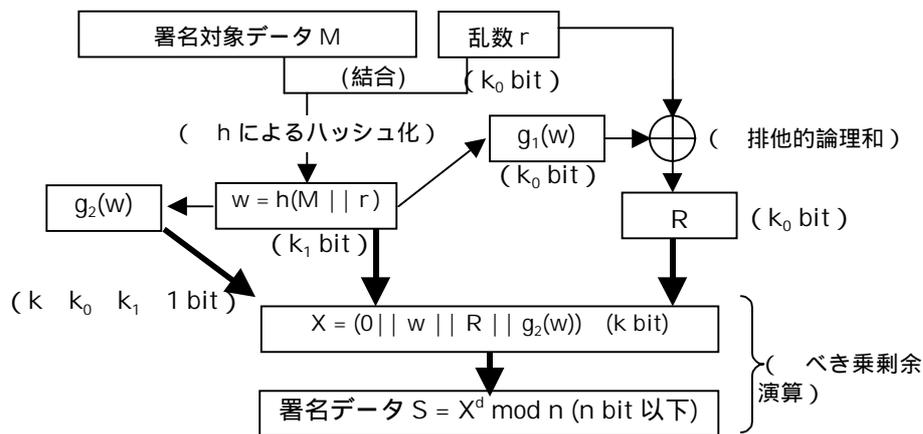


図9 PSS署名の署名生成手順の概念図

(2) PSS-R 署名

PSS-R 署名は、PSS 署名のアルゴリズムを、署名からメッセージを復元可能なように改良した方式である (Bellare and Rogaway[1996])。PSS-R 署名の安全性は PSS 署名と同様であり、前述の 2 つの条件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。

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。署名検証の際に、 $g_2(w) \oplus M$ に $g_2(w)$ の排他的論理和を計算することで、 M を復元することができる。

署名生成および検証に必要な計算量については、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。

8. CNS 攻撃法による標準化活動等への影響

ISO/IEC 9796-1,2 の標準化を担当している ISO/IEC JTC1/SC27/WG2 では、現在、CNS 攻撃法への対応方法を検討し始めている。CNS 攻撃法を発表した Coron、Naccache、Stern も SC27/WG2 の活動に協力するとしており²²、SC27/WG2 においてどのような判断が下されるのか、注目されるところである。

また、CNS 攻撃法によって影響を受ける可能性のある業界標準として、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 が挙げられる (Europay et al.[1996])。EMV'96 では、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述されている。EMV'96 に準拠した IC カードにおけるデジタル署名においては、本手順が利用されることとなっている。したがって、仮に、今後 ISO/IEC 9796-2 の改訂が行われることになれば、EMV'96 を利用するシステムへの影響が生じる可能性がある。

²² Silverman and Naccache[1999]を参照。

9. おわりに

ISO/IEC 9796-2 のデジタル署名方式は、主に IC カードにおける実装を想定して設計されており、メッセージ復元機能を有する特殊な署名変換対象データのフォーマットが採用されていた。CNS 攻撃法は、こうした署名変換対象データのフォーマットの特徴を利用した攻撃法であり、暗号アルゴリズム自体が高い信頼性を有していたとしても、利用方式によって安全性が損われる可能性があることを改めて示唆するものである。

デジタル署名をはじめとする暗号技術の実装方法の多様化が進む中、1998年6月に SSL に対して能動的攻撃が可能となることが Bleichenbacher によって示されたように²³、CNS 攻撃法が現実化する可能性は否定できない。このため、デジタル署名方式を実装する際には、その実装環境を十分に考慮した上で安全性を評価する必要がある。

また、能動的攻撃に対する安全性を高める方法として、PSS 署名や PSS-R 署名といった安全性が証明されているデジタル署名方式を採用することも検討する必要がある。安全性が証明されている公開鍵暗号方式としては OAEP や EPOC²⁴が提案されているほか、デジタル署名方式としては TSH-ESIGN²⁵が提案されている。OAEP は一部の技術仕様に既に採用されており、デジタル署名方式の分野においても、安全性が証明されている方式を採用する動きが広がる可能性がある。ただし、こうした暗号方式は、一定の仮定の下での安全性を証明するものであり、実装環境次第でそうした仮定が満足されなくなる可能性もある点には留意が必要である。

このように、実装環境を含めた暗号方式の安全性に関する評価は非常に重要である。今後も、安全性証明等の理論研究と実装技術に関する研究の両者の動向について注視していく必要がある。

以 上

²³ 詳細については、Bleichenbacher[1998]、宇根・岡本[1999a]を参照。

²⁴ EPOC については、Okamoto, Uchiyama and Fujisaki[1998]、宇根[1998]を参照。

²⁵ TSH-ESIGN については、Okamoto, Fujisaki and Morita[1998]を参照。

参考文献

- 岩下直行・谷田部充子、「金融分野における情報セキュリティ技術の国際標準化動向」、『金融研究』第18巻第2号、日本銀行金融研究所、1999年4月
- 宇根正志、「公開鍵暗号方式 EPOC について」, IMES Discussion Paper Series 98-J-19、日本銀行金融研究所、1998年8月
- ・岡本龍明、「安全性証明付きの公開鍵暗号方式を巡る動向について」、『1999年暗号と情報セキュリティシンポジウム予稿集』, pp.881-886、1999年1月 (1999a)
 - ・ 、「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 (1999b)
- 田中初一、「ID情報に基づく岡本鍵配送方式の解析」、『電子情報通信学会技術報告』ISEC88-7、電子情報通信学会、1988年5月
- American National Standards Institute, "X 9.31 Digital Signatures Using Reversible Public Key Cryptography (rDSA)," 1998.
- Bellare, M., and P. Rogaway, "Optimal asymmetric encryption," *Advances in Cryptology Proceedings of EUROCRYPT '94*, Lecture Notes in Computer Science, Vol. 950, pp.92-111, Springer-Verlag, 1995.
- and , "The Exact Security of Digital Signatures How to Sign with RSA and Rabin," *Advances in Cryptology Proceedings of EUROCRYPT '96*, Lecture Notes in Computer Science, Vol. 1070, pp.399-416, Springer-Verlag, 1996.
- Bleichenbacher, D., "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," *Advances in Cryptology Proceedings of CRYPTO '98*, Lecture Notes in Computer Science, Vol. 1462, pp.1-12, Springer-Verlag, 1998.
- Coron, J. S., D. Naccache, and P. Stern, "On the Security of RSA Padding," to appear *CRYPTO '99*, 1999.
- Desmedt, Y. G., and A. M. Odlyzko, "A chosen text attack on the RSA cyptosystem and some discrete logarithm schemes," *Advances in Cryptology CRYPTO '85*, Lecture Notes in Computer Science, Vol. 218, pp.516-522, Springer-Verlag, 1986.
- 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>)
- Girault, M., and J. F. Misarsky, "Selective Forgery of RSA Signatures Using Redundancy," *Advances in Cryptology Proceedings of EUROCRYPT '97*, Lecture Notes in Computer Science, Vol. 1233, pp. 495-507, Springer-Verlag, 1997.

- Guillou, L. C., J.-J. Quisquater, 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, Lecture Notes in Computer Science, Vol. 473, pp. 465-473, Springer-Verlag, 1991.
- International Organization for Standardization, "ISO 11166-2 Banking Key management by means of asymmetric algorithms Part 2: Approved algorithms using the RSA cryptosystem," 1994.
- , "ISO/TR 13569 Banking and related financial services Information security guidelines," 1997.
- 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.
- and , "ISO/IEC 9796 Information technology Security techniques Digital signature scheme giving message," 1991.
- Kaliski, B., and M. Robshaw, "The Secure Use of RSA," *CryptoBytes*, Vol. 1, No. 3, RSA Laboratories, 1995.
- Menezes, A. J., P. C. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- Misarsky, J. F., "A Multiplicative Attack Using LLL Algorithm on RSA Signatures with Redundancy," *Advances in Cryptology CRYPTO '97, Lecture Notes in Computer Science*, Vol. 1294, pp. 221-234, Springer-Verlag, 1997.
- , "How (not) to design RSA signature schemes," *Proceedings of Public Key Cryptography '98, Lecture Notes in Computer Science*, Vol. 1431, pp.14-28, Springer-Verlag, 1998.
- National Institute for Standards and Technology, "Specifications for a digital signature standard," *Federal Information Processing Standard Publication 186-1*, 1999.
- Okamoto, T., E. Fujisaki, and H. Morita, "TSH-ESIGN: Efficient Digital Signature Scheme Using Trisection Size Hash," *Submission to IEEE P1363a*, November 1998.
- Okamoto, T., S. Uchiyama, and E. Fujisaki, "EPOC: Efficient Probabilistic Public-Key Encryption," *Submission to IEEE P1363a*, November 1998.
- 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.
- RSA Laboratories, "PKCS #1: RSA Encryption Standard Version 1.5," November 1993.
- RSA Laboratories, "PKCS #1: RSA Cryptography Specifications Version 2.0," September 1998. (<ftp://ftp.rsa.com/pub/pkcs/ascii/pkcs-1v2.asc>)

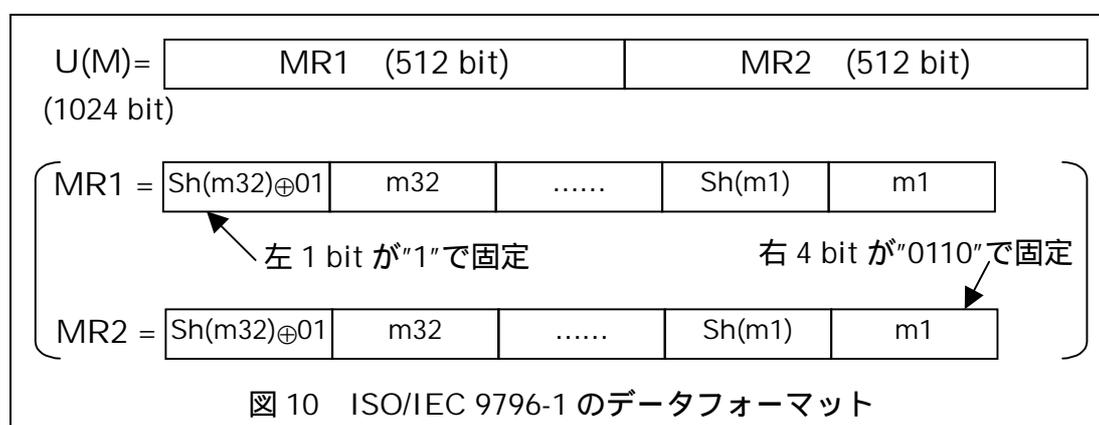
Silverman, R. D., and D. Naccache, "Recent Results on Signature Forgery," April 1999. (<http://www.rsa.com/rsalabs/html/sigforge.html>)

補論 1 : ISO/IEC 9796-1,2,3 の署名変換対象データのフォーマット

1. ISO/IEC 9796-1 の署名変換対象データ

ISO/IEC 9796-1 では、メッセージ全体を署名から復元可能にするため、比較的短いメッセージのみ利用可能となっている。公開鍵のサイズを 1024 bit とする場合、利用可能なメッセージのサイズは 256 bit 以下となる。以下では、メッセージのサイズが 256 bit、公開鍵のサイズが 1024 bit のケースを想定して説明する (ISO/IEC[1991])。

署名変換対象データ $U(M)$ のサイズは公開鍵のサイズと同一の 1024 bit となり、図 10 に表されるフォーマットとなる。ただし、メッセージ M を 8 bit 毎に分割し、右から $m_1, m_2, \dots, m_{31}, m_{32}$ とする。



このように、署名変換対象データ $U(M)$ は MR1 と MR2 (いずれも 512 bit) によって構成されている。

MR1 と MR2 には、8 bit ごとに分割されたメッセージ m_i ($i=1, \dots, 32$) と、 m_i を換字変換 Sh (入出力値はともに 8 bit) によって変換したデータ $Sh(m_i)$ が含まれており、これら 2 種類のデータが交互に配置されている。ただし、 m_{32} に対しては、換字変換したデータ $Sh(m_{32})$ と "0000 0001" との排他的論理和が計算される。

最後に、MR1 のデータについては、左から 1 bit が "1" に固定されるほか、MR2 のデータについては、右から 4 bit が "0110" に固定される。

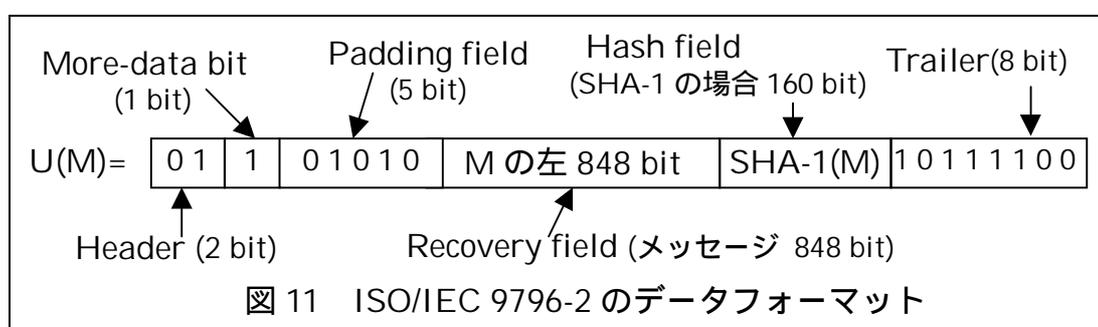
このように生成された $U(M)$ に対して、秘密鍵 d による以下の変換によって署名 S が生成される。

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

2. ISO/IEC 9796-2 の署名変換対象データ

ISO/IEC 9796-2 では、署名変換対象データを生成する際にハッシュ関数²⁶が利用されており、ISO/IEC 9796-1 よりも大きなサイズのメッセージの署名を生成可能である。例えば、公開鍵が 1024 bit の場合、848 bit までであればメッセージ全体を復元することが可能である。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、メッセージのうち左から 848 bit 分のデータに対する署名を生成し、残りは署名に添付して送信するという方法が採用されている。

以下では、公開鍵およびメッセージが 1024 bit、ハッシュ関数に SHA-1(ハッシュ値は 160 bit)を利用する場合を想定し、署名変換対象データ U(M)のフォーマットを説明する。署名変換対象データは公開鍵のサイズと同一で 1024 bit となり、フォーマットは図 11 の通り (ISO/IEC[1997])



U(M)は以下の 6 つのパートから構成される。

- (i) Header : U(M)の左から 2 bit 分のデータであり、常に“01”となる。
- (ii) More-data bit : Header に続く 1 bit のデータ。この例では、メッセージ全体が復元できないため、“1”となる。
- (iii) Padding field : U(M)を 1024 bit にするためのパディングデータ。この例では 5 bit (=1024 bit - 2 bit[Header] - 1 bit[More-data bit] - 848 bit[Recovery field] - 160 bit[Hash field] - 8 bit[Trailer]) となり、“01010”となる。

²⁶ 本標準規格において利用されるハッシュ関数としては、ISO/IEC 10118 (Information technology Security techniques Hash-functions、ISO/IEC) に規定される方式を利用するよう記載されている。具体的には、Matyas-Meyer-Oseas 方式、MDC-2、RIPEMD-160、SHA-1 等の方式が挙げられている。

(iv) Recovery field: メッセージ M のうち左から 848 bit 分のデータが入る。残りの 176 bit (=1024 bit - 848 bit) 分のデータは署名とともに送付される。

(v) Hash field: メッセージ M に対するハッシュ値 SHA-1(M)が入る。ハッシュ関数として SHA-1 を想定しているため、ハッシュ値のサイズは 160 bit となる。

(vi) Trailer: ハッシュ関数の属性を表すデータ。SHA-1 を利用することを想定しているため、8 bit のデータ"1011 1100"となる。

こうして生成された $U(M)$ に対して、秘密鍵による署名生成変換によって署名 $S=U(M)^d \text{ mod } n$ が生成される。

3. ISO/IEC 9796-3 の署名変換対象データ

ISO/IEC 9796-3 のデジタル署名方式には、ハッシュ関数が利用されず、 $C = M \text{ mod } (2^{79} + 1)$ という検証式によって生成される 80 bit の検証コード C が利用されている (Misarsky[1998])。以下では、公開鍵のサイズを 1024 bit、メッセージのサイズが 929 bit の場合を例として、署名変換対象データのフォーマットを説明する (図 12 参照)。

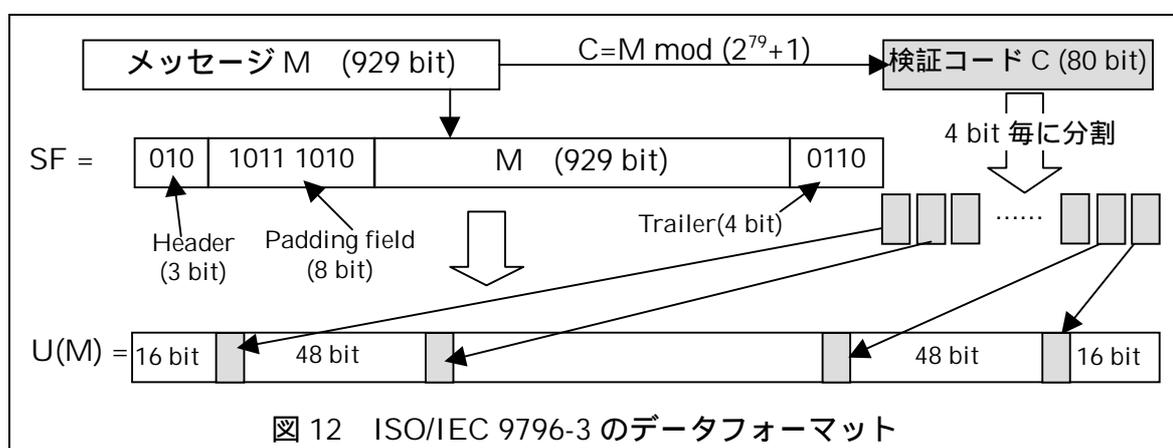


図 12 ISO/IEC 9796-3 のデータフォーマット

署名変換対象データの生成は以下の 3 段階で行われる。

(i) 中間データ SF (944 bit) が生成される。SF は 4 つの部分から構成されており、左から、3 bit の Header、8 bit の Padding field、929 bit のメッセージ、4 bit の Trailer となっている。

(ii)80 bit の検証コード C を生成する。検証コードは $C = M \bmod (2^{79} + 1)$ によって生成される。

(iii)検証コード C を 20 個の 4 bit データに分割し、中間データ SF の特定の位置に埋め込まれる。具体的には、 SF の左から 16 bit 目に 1 個挿入された後、48 bit 毎に 1 個ずつ挿入される。この結果生成される 1024 bit (=944 bit + 80 bit) のデータが署名変換対象データ $U(M)$ となる。

こうして生成された $U(M)$ に対して、秘密鍵による署名生成変換によって署名 $S = U(M)^d \bmod n$ が生成される。

補論 2 : CNS 攻撃法の概要

< 以下の説明に利用される記号の意味 >

- U : メッセージ M から署名変換対象データ U(M)を生成する関数。U(M)のサイズは公開鍵のサイズと同一となる。
- L : 攻撃の際に素因数分解が必要となる数値のサイズ (bit 数)
- p_k : 最小の素数 2 から順番に数えて k 番目に大きな素数
- (n, e) : RSA 署名の公開鍵
- d : RSA 署名の秘密鍵

1. 攻撃に利用されるメッセージ M の選択

(1)メッセージ M の条件

攻撃者に利用可能なメッセージ M の条件は、その署名変換対象データ U(M) が p_k -smooth となることである。すなわち、M は

$$U(M) = \prod_{j=1}^k p_j^{v_j} \dots$$

を満足する (ただし、 p_j は p_k 以下の素数)。

(2)攻撃に必要な素因数分解の計算量を削減するための工夫

式を満足する M を 1 つみつける際に、式を満足する U(M)を直接素因数分解して p_k -smooth となっているかどうかを検証するという方法では、U(M)のサイズが公開鍵のサイズと同一であることから、必要な計算量は公開鍵の素因数分解と同程度となり、計算量を削減することは困難である。

このため、U(M)を直接素因数分解する方法の代わりに、以下の方法を採用する。

(a)公開鍵 n の形態や署名変換対象データのフォーマットに着目して、 $a \cdot n - b \cdot U(M)$ のサイズ (L bit) が、公開鍵のサイズに比べて十分小さくなるように a、b および M の一部を定める²⁷。ただし、b の値は p_k -smooth となるように選択される。

(b) $a \cdot n - b \cdot U(M)$ が p_k -smooth となるように M の残りの部分を定める。

²⁷ ISO/IEC 9796-1 に本攻撃法を適用する場合 (公開鍵のサイズを 1024 bit、メッセージのサイズを 256 bit に設定)、 $a=0$ かつ $b=(-1/)$ となるほか、メッセージのうちの 64 bit 分が一定の値に定められる。このとき、L の値は 64 bit となる。一方、ISO/IEC 9796-2 に本攻撃法を適用する場合 (公開鍵、メッセージともに 1024 bit に設定)、 $b=2$ または 2^8 となるほか、a の値は n の形態に応じて決定される。メッセージについては、左から 848 bit 分が n の値に依存して決定される。このとき、L の値は高々 176 bit となる。

上記の(a)、(b)の手順に沿って a 、 b 、 M を定めることにより、以下の等式が成立する。

$$a \cdot n - b \cdot U(M) = \prod_{j=1}^k p_j^{v_j} \dots$$

‘式が成立する M によって $U(M)^d \bmod n$ が p_k -smooth となることを以下の手順で確かめることができる。

まず、 $a \cdot n - b \cdot U(M)$ を d 乗して展開し、 $\bmod n$ を計算する。

$$(a \cdot n - b \cdot U(M))^d \bmod n = \left\{ (a \cdot n)^d + d(a \cdot n)^{d-1}(-b \cdot U(M)) + \dots + (-b \cdot U(M))^d \right\} \bmod n$$

右辺括弧内の項のうち、 n がかかっている項の $\bmod n$ を計算すると 0 となることから、

$$(a \cdot n - b \cdot U(M))^d \bmod n = (-b \cdot U(M))^d \bmod n = (-b)^d U(M)^d \bmod n$$

上記の式を $U(M)^d \bmod n$ について整理すると、

$$U(M)^d \bmod n = (-b)^{-d} (a \cdot n - b \cdot U(M))^d \bmod n$$

これに ‘式を代入すると、

$$U(M)^d \bmod n = (-b)^{-d} \left(\prod_{j=1}^k p_j^{v_j} \right)^d \bmod n$$

となる。 b の値が p_k -smooth となることから、 $U(M)^d \bmod n$ は必ず p_k -smooth となる。

(3)メッセージ M の選択に必要な計算量

攻撃に利用するメッセージを 1 つみつけるために必要な計算量は、攻撃者が定めた p_k に対して、 $a \cdot n - b \cdot U(M)$ の取り得る値の中に p_k -smooth な値が存在する確率 X と、 $a \cdot n - b \cdot U(M)$ が p_k -smooth であるか否かを検証するために必要な計算量 Y を求め、 $X \cdot Y$ を計算する必要がある。

(a) $a \cdot n - b \cdot U(M)$ の取り得る値の中に p_k -smooth な値が存在する確率

$a \cdot n - b \cdot U(M)$ の取り得る値の数は、 $a \cdot n - b \cdot U(M)$ のサイズが L bit であることから 2^L となる。 L bit のデータの中で p_k -smooth となる合成数の割合は、

p_k のサイズ ($\log_2(p_k)$ bit) と L の値に依存しており、以下の関数 $\rho(t)$ によって与えられる。ただし、 $t = L/\log_2(p_k)$ である。

$$\rho(t) \cong (2\pi t)^{-1/2} \exp\left(\gamma - t\zeta + \int_0^\zeta \frac{e^s - 1}{s} ds\right) \quad (e^\zeta - 1 = t\zeta, \gamma: \text{正定数})$$

このように、 ρ 式を満足する M を 1 つみつけるためには、平均 ($1/\rho(L/\log_2(p_k))$) 個の数をランダムに選択する必要がある。

例えば、 L が 160 bit、 p_k のサイズが 24 bit の場合 ($t=160/24$ の場合) には、 $\rho(t) \approx 2^{-19}$ となる。これは、 2^{24} -smooth となる 160 bit の数を見つめるためには平均 2^{19} 個の自然数を選択する必要があることを意味する。

(b) $a \cdot n - b \cdot U(M)$ が p_k -smooth であるか否かを検証するための計算量

選択した M が ρ 式を満足するか否かを検証するための計算量のオーダーは、 $a \cdot n - b \cdot U(M)$ のサイズが L bit であることから $O(L\sqrt{k \ln k})$ となる²⁸。

以上の(a)と(b)より、 ρ 式を満足する M を 1 つみつけるために必要となる平均的な計算量 $C_{L,k}$ は、

$$C_{L,k} = O\left(\frac{L\sqrt{k \ln k}}{\rho(L/\log_2(k \ln k))}\right)$$

となる。攻撃者は、 $C_{L,k}$ が最小となるように p_k の値を設定する。

2. 署名の偽造方法

(1) 選択した M に対する署名入手

攻撃者は、 ρ 式を満足する r 個のメッセージ ($M_1, M_2, \dots, M_{r-1}, M_r$) を入手し、正当な署名者に送信する。署名者は、自分の秘密鍵を利用して ρ 式で表される署名 $U(M_i)^d \pmod n$ を生成し、攻撃者に返信する。

$$U(M_i)^d = \prod_{j=1}^k p_j^{(v_{i,j})^d \pmod n} \dots$$

署名偽造の対象となるメッセージを M_{r+1} とすると、 M_{r+1} も ρ 式を満足する必要がある。

²⁸ ここで利用されている素因数分解アルゴリズムは Pollard-Brent のアルゴリズムである (Coron et al.[1999])。Coron らは、Pollard-Brent のアルゴリズムが比較的小さな素数からなる合成数の素因数分解を行う際に有効なアルゴリズムであるとして、CNS 攻撃法に利用している。

(2)署名の偽造方法

攻撃者は、入手した署名 $U(M_i)^d \bmod n$ から M_{r+1} に対する署名を偽造するために、 $U(M_i)^d \bmod n$ を説明変数、 $U(M_{r+1})^d \bmod n$ を従属変数とする方程式を以下の手順で導出する。

<1> $U(M_i)$ を、以下のように表される k 次元ベクトル V_i に対応させる。ただし、 e は RSA 署名方式の公開鍵の一部である。

$$U(M_i) \mapsto V_i = \{v_{i,1} \bmod e, v_{i,2} \bmod e, \dots, v_{i,k} \bmod e\}$$

V_i を要素とするベクトルの集合の要素は e^k 個存在し、基底となるベクトルは $\{p_1, p_2, p_3, \dots, p_k\}$ である。

<2>偽造対象メッセージ M_{r+1} にも、 $U(M_{r+1})$ に対応する k 次元ベクトル V_{r+1} が存在し、 V_{r+1} は他の r 本のベクトル V_i の線形結合で表される。すなわち、 V_{r+1} は適当な定数 b_i ($i=1, \dots, r, 0 \leq b_i \leq e-1$) によって、

$$V_{r+1} = \sum_{i=1}^r b_i V_i \bmod e = \{v_{r+1,1} \bmod e, v_{r+1,2} \bmod e, \dots, v_{r+1,k} \bmod e\} \dots$$

と表される。

式に $V_i = \{v_{i,1} \bmod e, v_{i,2} \bmod e, \dots, v_{i,k} \bmod e\}$ を代入すると、

$$\begin{aligned} \sum_{i=1}^r b_i V_i \bmod e &= \sum_{i=1}^r b_i \{v_{i,1} \bmod e, v_{i,2} \bmod e, \dots, v_{i,k} \bmod e\} \bmod e \\ &= \left\{ \sum_{i=1}^r b_i v_{i,1} \bmod e, \sum_{i=1}^r b_i v_{i,2} \bmod e, \dots, \sum_{i=1}^r b_i v_{i,k} \bmod e \right\} \dots \end{aligned}$$

となる。

<3> 式と 式の右辺を比較すると、 $v_{r+1,j}$ について以下の等式が得られる。

$$v_{r+1,j} = \sum_{i=1}^r b_i v_{i,j} \bmod e$$

さらに、適当な a_j を選び、 $v_{r+1,j}$ を 式のように表す。

$$v_{r+1,j} = \sum_{i=1}^r b_i v_{i,j} - a_j e \dots$$

<4>偽造対象メッセージ M_{r+1} に対する署名 $U(M_{r+1})^d \bmod n$ は、以下のように表される。

$$U(M_{r+1})^d = \prod_{j=1}^k p_j^{(v_{r+1,j})^d} \bmod n \dots$$

式に 式を代入し、以下のように変形する。

$$\begin{aligned} U(M_{r+1})^d &= \prod_{j=1}^k p_j^{\left(\sum_{i=1}^r b_i v_{i,j} - a_j e \right)^d} \pmod n \\ &= \prod_{j=1}^k p_j^{\left(\sum_{i=1}^r b_i v_{i,j} \right)^d} \times \prod_{j=1}^k p_j^{(-a_j e d)} \pmod n \dots \end{aligned}$$

RSA 署名方式の演算の性質 $e \times d \pmod{(p-1)(q-1)} = 1$ を、 式の右辺第 2 項に適用すると、

$$U(M_{r+1})^d = \prod_{j=1}^k p_j^{\left(\sum_{i=1}^r b_i v_{i,j} \right)^d} \times \prod_{j=1}^k p_j^{(-a_j)} \pmod n$$

さらに、式を展開すると、

$$U(M_{r+1})^d = \prod_{j=1}^k p_j^{(b_1 v_{1,j} + b_2 v_{2,j} + \dots + b_r v_{r,j})^d} \times \prod_{j=1}^k p_j^{-a_j} \pmod n$$

$$\begin{aligned} U(M_{r+1})^d &= \prod_{j=1}^k \prod_{i=1}^r p_j^{(v_{i,j})^d \times b_i} \times \prod_{j=1}^k p_j^{-a_j} \pmod n \\ &= \prod_{i=1}^r \left(\prod_{j=1}^k p_j^{(v_{i,j})^d} \right)^{b_i} \times \prod_{j=1}^k p_j^{-a_j} \pmod n \dots \end{aligned}$$

式に 式を代入すると、

$$U(M_{r+1})^d = \prod_{i=1}^r (U(M_i)^d)^{b_i} \times \prod_{j=1}^k p_j^{-a_j} \pmod n \dots$$

式が、 M_{r+1} の署名を偽造するための方程式である。

<5> 式に入手した署名 $U(M_i)^d \pmod n$ を代入することによって、 $U(M_{r+1})^d \pmod n$ を生成することができる。

計算に必要となる M_i の個数は k と e の値に依存し、 $O(k \log e)$ と表される。必要となる計算量は $C_{L,k}$ に $O(k \log e)$ を乗じた値となり、

$$O\left(\frac{Lk(\log e)\sqrt{k \ln k}}{\rho(L/\log_2(k \ln k))}\right)$$

と表される。