RSA（PKCS＃１ v1.5）に関する評価報告書

2002 年 9 月 30 日

電気通信大学

太田 和夫

RSA (PKCS#1 v1.5) に関する評価報告書
(2002 年度外部評価のまとめ)

2002 年 9 月 30 日

要旨

CRYPTREC 公開鍵暗号評価小委員会では，RSA Primitive を利用した暗号／署名として RSA-OAEP (Optimal Asymmetric Encryption Padding), RSA-PSS (Probabilistic Signature Schemes) について 2001 年度までに安全性の評価等を終了し，電子政府推奨暗号リスト素案に含めることを確認している．

本評価期間（2002 年 4 月から 9 月）では，2002 年度になってから暗号技術評価委員会から,「使用実績のある暗号技術」ということで採用が要請された暗号／署名である RSA 法の一利用方法である PKCS#1 v1.5 について安全性の評価を行なった．

PKCS#1 v1.5 の署名法は,「電子署名法に係る指針に記載された方式」ということで従来より評価を行なってきたが，暗号法については評価期間が始まった後に暗号技術評価委員会より採用が要請されたために，外部評価を依頼できなかった．

本報告書は，署名法に関しては外部評価報告書（2 件) の内容をとりまとめたものである．暗号法に関しては，報告者が文献 [3] などにより調査したものである．

署名法に関して: 評価者#1: 「For PKCS#1 v1.5 (and ANSI x9.31), we have seen that the attack of [4] does not apply. To our knowledge, no attack better than factoring the modulus or finding a collision in the hash function, is known for PKCS#1 v1.5 (and ANSI x9.31). 」の記述，ならびに

評価者#2: 「The attacks are not a threat to the practical security of schemes described in the (ANSI X9.31 and) PKCS#1 v1.5 standards. 」の記述の妥当性を，評価者#1 の報告書を詳細に検討することで確認した．

暗号法に関して: 公開されている文献を精査して，PKCS#1 v1.5 で規定された暗号法に実効上有効な攻撃法は発見されなかったものの，選択暗号文攻撃，選択平文攻撃が理論的に存在することを確認した．

報告者は，評価者#2 の結論「The best candidate is the RSA PSS signature scheme which has been proven secure in the random oracle model. 」にも明言されているように,（暗号法/署名法にかかわらず,）証明可能安全性を持つ方式を採用すべきとの見解を持っている．

目次

第 1 章

PKCS#1 v1.5 で規定された署名法の安全性

1.1 はじめに

RSA 法は 1977 に Rivest, Shamir, Adleman [13] によって発明されて以降，最も広く使用されている公開鍵暗号である．

RSA 署名法は多くの場合，文書にハッシュ関数を施した後に，その値にパディングを付加して，秘密鍵でべき乗演算を行なう．この枠組みは，PKCS#1 v1.5, ANSI X9.31[1], ISO 9796-1[8], および ISO 9796-2[9], として標準化されてきた．

以下では，「Evaluation of Cryptographic Technology」，「Evaluation of Security Level of Cryptology: RSA Signature Schemes」の報告のなかから，特に，PKCS#1 v1.5 に関連する部分に注目して整理して報告する．

1.2 RSA 署名に対する攻撃法

PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 および ISO 9796-2 で規定された RSA 署名法は，一般的に次のように表記できる：

$$s = \mu(m)^d \bmod N.$$

ここで m は文書，$\mu(m)$ は文書 m のエンコード関数の値，N は RSA 法で使用する法の値を，d は秘密鍵の指数を表す．

$\mu(m)$ は固定値のパディングから構成される場合と，さらにハッシュ関数値も含めて構成される場合とがある．

1.2.1 パディングのみが付加される場合

文書にパディング値 P のみが先頭に付加される場合には

$$s = (P|m)^d \bmod N$$

と表記できる．さらに一般には，

$$R(m) = \omega \cdot m + a \tag{1.1}$$

$$\text{ただし} \begin{cases} w \text{ は 乗法的冗長性 (the multiplicative redundancy)} \\ a \text{ は 加法的冗長性 (the additive redundancy)} \end{cases}$$

と表記できる．

このとき，m の署名は

$$s = R(m)^d \bmod N$$

で計算できる．

Lenstra-Shparlinski's attack

パディングのみを用いる方式に対して現在知られている最も強力な攻撃法はLenstra と Shparlinski の攻撃法である．文献 [10] に示されている．

パディングのみが付加された場合の攻撃法は，拡張 Euclid 互除法（あるいは，連分数法）を利用したものであり，$\mu(m)$ 中で文書成分 m が占める割合が低いほど困難となる．また，式 (1.1) で規定された a と ω の制約条件が緩くなるほど困難になる．

Lenstra-Shparlinski の攻撃が有効となる $\mu(m)$ の文書のサイズは

$$|\text{message}| \succ \frac{1}{3}|N|$$

となっている（図 1.1).

文献 [10] では，この攻撃法の計算量はヒューリスティックには漸近的に

$$\exp\left((1 + o(1))(\log N)^{1/3}(\log\log N)^{2/3}\right)$$

であることが示されており，選択文書に対して 1024-bit の RSA 法での偽造例が示されている．この攻撃は多項式時間アルゴリズムではないが，選択文書攻撃である点が興味深い．

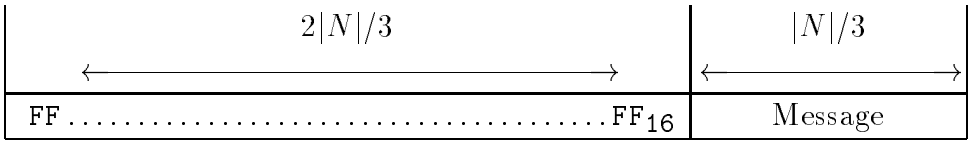

図 1.1: 偽造可能な RSA パディング形式（$\omega = 1$ かつ $a = \texttt{FF...FF 00...00}_{16}$

1.2.2 ハッシュ値も付加される場合

PKCS#1 v1.5, ANSI X9.31, ISO 9796-1 および ISO 9796-2 では，ハッシュ値が関数 $\mu(m)$ で用いられている．よって，前述の攻撃方法は，これらのハッシュ値を付加する署名方式には直接は適用できないと考えられる．

定義 1.2.1 整数のすべての素因数が y より小さいとき，その整数を y-smooth な数とよぶ．

現在知られているハッシュ関数値をエンコード関数を用いる方式に対する攻撃法は，y-smooth の概念を利用した次の攻撃法 [5, 12] をもとにしている．攻撃の目標は，d を知らずに，$\mu(m)^d \bmod N$ を求めることである．

Desmedt-Odlyzko attack [5]

1. $\mu(m)$ を小さな素数 p_i のみの積に因数分解する．
2. 小さな素数 p_i のみの積に因数分解できる $\mu(m_j)$ を用いて $p_i^d \bmod N$ の値を求める．
3. $\mu(m)$ の小さな素数 p_i に対する $p_i^d \bmod N$ の値の積を計算して，m の署名を求める．

攻撃の計算量は $\mu(m)$ のサイズのみに依存して決まる．攻撃は $\mu(m)$ のサイズが小さいときのみに有効である．(そうでない場合には，$\mu(m)$ が小さな素数の積で表せる確率は非常に小さくなる．).

Coron-Naccache-Stern attack [4]

Coron, Naccache および Stern はDesmedt と Odlyzko　の攻撃を拡張して，ISO ISO 9796-2 の攻撃に成功した [4].

その後，Coppersmith，Halevi　および Jutla はISO 9796-1 の攻撃に成功した [2]．さらに Grieu はこの攻撃の効率を高めた [7].

以下,「Evaluation of Cryptographic Technology」にしたがいながら，攻撃に必要となる計算量の評価を行なう.

一連の攻撃方法は,

$$t = a \cdot \mu(m) + b \cdot N \tag{1.2}$$

が小さな値となるような a と b を見つける，あるいは,

$$\mu(m) = c \cdot t \tag{1.3}$$

に現れる t が小さな値となるように c を見つけるとき，次の手順として一般的にとらえることができる.

式 (1.2)　の場合には，$c = a^{-1} \bmod N$ とおくことで,

$$\mu(m) = c \cdot t \bmod N$$

と表せるので，式 (1.3) に帰着できる．ここで，t が小さな整数となることを期待していることに注意.

攻撃の第一フェーズでは，まず,

$$\mu(m_i) = c \cdot t_i \bmod N \tag{1.4}$$

の中に現れる t_i が y-smooth となるように，多数の文書 m_i を入手する．ここで y はパラメータであり，計算量とメモリ量を考慮して最適値を求める（最適値の選択方法は付録 A を参照.）.

y より小さい素数のリストを $(p_1, \ldots, p_k)$ で表す.

$$\mu(m_i) = c \cdot \prod_{j=1}^{k} p_j^{v_{i,j}} \bmod N \quad \text{for } 1 \le i \le \tau$$

となるとき，$\mu(m_i)$ に対して

$$\mu(m_i) \longmapsto \vec{V_i} = \{1, v_{i,1} \bmod e, \ldots, v_{i,k} \bmod e\}$$

で，$k+1$-次元の vector $\vec{V_i}$ を対応付ける．

攻撃の第二フェーズでは，Gauss の消去法を用いて，ある vector $\vec{V_\tau}$ を他の voctor の一次結合で，

$$\vec{V_\tau} = \sum_{i=1}^{\tau-1} \beta_i \vec{V_i} \mod e \tag{1.5}$$

と表す．このとき，式 (1.5) で用いられている β_i と $\bmod e$ を γ_j を用いて書く下すことで

$$v_{\tau,j} = \sum_{i=1}^{\tau-1} \beta_i \cdot v_{i,j} - \gamma_j \cdot e \quad \text{for all} \quad 1 \leq j \leq k$$

をえる．

$$\delta = \prod_{j=1}^{k} p_j^{-\gamma_j}$$

とおくと，

$$\mu(m_\tau) = \delta^e \cdot \prod_{i=1}^{\tau-1} \mu(m_i)^{\beta_i} \mod N$$

となる．これより攻撃者は，始めの $\tau-1$ 個の文書 m_i とその署名を用いて，第 τ 番目の文書 m_τ の署名を

$$\mu(m_\tau)^d = \delta \cdot \prod_{i=1}^{\tau-1} \left(\mu(m_i)^d\right)^{\beta_i} \mod N$$

で偽造できる．

ここで，

$$y = L_x[\beta] = \exp\left(\beta \cdot \sqrt{\log x \log\log x}\right)$$

と表すと，$\beta = 1/\sqrt{2}$ のとき，計算量が最適となることが示せて，上記の攻撃法の計算量は，

$$L_x[\sqrt{2} + \circ(1)]$$

$\lvert x\rvert$	$\log_2$ **time**	$\log_2$ **space**
64	26	13
96	34	17
128	41	20
192	52	26
256	62	31
368	77	38

表 1.1: 一般化された CNS 攻撃の複雑度

メモリ量は

$$L_x\left[\frac{\sqrt{2}}{2}+\circ(1)\right]$$

となる.

従って，攻撃の複雑度は，整数 t_i のサイズの準指数時間アルゴリズムとなる．t_i を小さく選べるときにのみ有効となることに注意.

$L_x[\sqrt{2}]$ と $L_x[\sqrt{2}/2]$ の関数値を表 1.1 に示す．これより，t_i のサイズが 128 より小さいときには，攻撃が実行可能と考えられる.（注：この表は，あくまでも目安であり，実際には計算量の定数項までの影響を吟味しなければならないことを注意しておく.）

1.3 PKCS#1 v1.5 への適用

前の章で示した手順を PKCS#1 v1.5 に適用する．文献 [4] では RSA の法の値が特殊な場合にのみ有効な攻撃手順が示されていたが，前章の手順は法の値が任意でよいことに注意.

1.3.1 PKCS#1 v1.5

PKCS#1 v1.5 [14] で規定された関数 $\mu(m)$ の形式は以下のとおり:

$$\mu(m) = \mathtt{0001}_{16}\|\mathtt{FFFF}_{16}\ldots\mathtt{FFFF}_{16}\|\mathtt{00}_{16}\|c_{\mathrm{SHA}}\|H(m)$$

ℓ	$\log_2$ **time**	$\log_2$ **space**
128	100	50
160	102	51

表 1.2: PCKS#1 v1.5 に対する CNS 攻撃の計算複雑度 ($n = 1024$ の場合)

ここで c_{SHA} は定数 $c_{\mathrm{SHA}} = \texttt{3021300906052B0E03021A05000414}_{16}$ であり，$H(m) = \mathrm{SHA}(m)$ である．

値 t を小さくするために次の工夫をする．

関数 $\mu(m)$ を

$$\mu(m) = c + H(m)$$

と表す．ここで，c は定数であり，H は出力値が ℓ ビットとする．以下では，n で法 N のビット数を表す．

ビットサイズが $(n-\ell)/2$ となる整数 a と，$(n+\ell)/2$ となる b が，関係式

$$a \cdot c = b \bmod N$$

をみたすように選ぶ．これは，拡張 Euclid の互除法（あるいは，c/N の連分数近似）を実行すればよく，効率よく実行可能である [6]．(付録 ?? を参照.)

求めた a と b を用いて,

$$a \cdot \mu(m) = b + a \cdot H(m) = t \tag{1.6}$$

を計算すると，サイズの選び方から，b のサイズは $(n+\ell)/2$ ビットであり，$a \cdot H(m)$ のサイズは $(n-\ell)/2 + \ell = (n+\ell)/2$ ビットなので，t は $(n+\ell)/2$ ビットとなる．

これより，章 1.2.2 で示した攻撃手順を適用すると，表 1.1 の値をえたのと同様にして，法のサイズが 1024-ビットの場合，計算複雑度の表 1.2 をえる．一般化された CNS 攻撃が非現実的なことが確認できる．

1.4 まとめ

以上により，報告者# 1 の「Conclusion」に記述された,「For PKCS#1 v1.5 and ANSI x9.31, we have seen that the attack of [4] does not apply. 」を確認した．ま

た，報告者＃2 も「The attacks are not a threat to the practical security of schemes described in the (ANSI X9.31 and) PKCS#1 v1.5 standards. 」と同様の結論に至っているので，現時点では（2002 年 9 月末）「To our knowledge, no attack better than factoring the modulus or finding a collision in the hash function, is known for PKCS#1 v1.5 and ANSI x9.31. 」と判断できる．

参考文献

[1] ANSI X9.31. *Digital signatures using reversible public-key cryptography for the financial services industry (rDSA)*, 1998.

[2] Coppermith, D. and Halevi, S. and Jutla, C. *ISO 9796-1 and the new forgery strategy. Research contribution to P1363*, 1999. `http://grouper.ieee.org/groups/1363/contrib.html`.

[3] Jean-Sébastian Coron, Marc Joye, David Naccache, and Pascal Paillier. New attacks on PKCS# 1 v1.5 Encryption. In *Advances in Cryptology — EUROCRYPT'2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 369–379. Springer-Verlag, 2000.

[4] J.S. Coron, D. Naccache, and J.P. Stern. On the security of RSA padding. In *Advances in Cryptology — CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 1–18, Berlin, 1999. Springer-Verlag.

[5] Y. Desmedt and A. M. Odlyzko. A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes. In Hugh C. Williams, editor, *Advances in Cryptology - Crypto '85*, pages 516–522, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 218.

[6] M. Girault, P. Toffin, and B. Vallee. Computation of approximate L-th roots modulo n and application to cryptography. In Shafi Goldwasser, editor, *Advances in Cryptology - Crypto '88*, pages 100–118, Berlin, 1989. Springer-Verlag. Lecture Notes in Computer Science Volume 403.

[7] F. Grieu. A chosen message attack on the ISO/IEC 9796-1 signature scheme. In *Advances in Cryptology — EUROCRYPT'2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 70–80. Springer-Verlag, 2000.

[8] ISO/IEC 9796. *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 1 : Mechanisms using redundancy*, 1999.

[9] ISO/IEC 9796-2. *Information technology - Security techniques - Digital signature scheme giving message recovery, Part 2 : Mechanisms using a hash-function*, 1997.

[10] Arjen K. Lenstra and Igor E Shparlinski. Selective forgery of RSA Signatures with fixed-padding. In *PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 228–236, Berlin-Heidelberg, 2002. Springer-Verlag.

[11] H.W. Lenstra. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

[12] J.-F. Misarsky. How (not) to design rsa signature schemes. In Hideki Imai and Yuliang Zheng, editors, *Public-key cryptography*, pages 14–28, Berlin, 1998. Springer-Verlag. Lecture Notes in Computer Science Volume 1431.

[13] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[14] RSA Laboratories. *PKCS # 1 Version 1.5: RSA Cryptography Standard*, November 1993. `http//www.rsa.com/rsalabs/pubs/PKCS/`.

付録 A

The Complexity of Generalized Coron-Naccache-Stern's Attacks

(この章は評価者# 1 からの評価レポートをもとに報告者が編集したものである[†1].)

A.1 The Properties of y-smooth

The attack complexity depends on the probability that the integers t_i are y-smooth. Defining $\psi(x,y) = \#\{v < x, \text{ such that } v \text{ is } y\text{-smooth}\}$, it is known that, for large x, the ratio $\psi(x, \sqrt[t]{x})/x$ is equivalent to Dickman's function defined by :

$$\rho(t) = \begin{cases} 1 & \text{if } \ 0 \leq t \leq 1 \\ \\ \rho(n) - \displaystyle\int_n^t \frac{\rho(v-1)}{v} dv & \text{if } \ n \leq t \leq n+1 \end{cases}$$

$\rho(t)$ is thus an approximation of the probability that a u-bit number is $2^{u/t}$-smooth. In particular, denoting:

$$y = L_x[\beta] = \exp\left(\beta \cdot \sqrt{\log x \log\log x}\right)$$

the probability that an integer between one and x is $L_x[\beta]$-smooth is:

$$\frac{\psi(x,y)}{x} = L_x\left[-\frac{1}{2\beta} + \circ(1)\right]$$

[†1]明らかな誤りについて修正を加えたことと，A.7 節の証明は報告者によるものである.

If we assume that the integers t_i in (1.4) are uniformly distributed between one and x, we have to generate on average $L_x[1/(2\beta) + o(1)]$ integers t_i.

A.2 The Complexity of Phase I

Using the ECM factorization algorithm [11], a prime factor p of an integer n is extracted in time:

$$L_p[\sqrt{2} + o(1)]$$

A y-smooth integer can thus be factorized in time:

$$L_y[\sqrt{2} + o(1)] = L_x[o(1)] \tag{A.1}$$

The complexity to find an integer t_i which is y-smooth using ECM is thus:

$$L_x\left[\frac{1}{2\beta} + o(1)\right]$$

A.3 The Complexity of Phase II

Moreover, the number τ of integers which are necessary to find a vector which is a linear combination of the others is $\mathcal{O}(y \cdot \log e)$ (see [4] and A.6 for more details). Therefore, one must solve a system with $\tau = L_x[\beta + o(1)]$ equations in $\tau = L_x[\beta + o(1)]$ unknown. Using Lanczos iterative algorithm, the time required to solve this system is $\mathcal{O}(r^2)$ and the space required is roughly $\mathcal{O}(r)$.

A.4 The Total Complexity of Attacks

To summarize, the time required to obtain the $L_x[\beta + o(1)]$ necessary equations is

$$L_x\left[\beta + \frac{1}{2\beta} + o(1)\right]$$

This system is solved in time

$$L_x[2\beta + o(1)]$$

and space

$$L_x[\beta + \circ(1)]$$

The complexity is minimal by taking $\beta = 1/\sqrt{2}$, since $\beta = \frac{1}{2\beta}$, i.e., $\beta^2 = \frac{1}{2}$.

A.5 The Proof of Equation (A.1)

Let

$$c(y) = L_y[\sqrt{2} + o(1)]$$

This means that there exits a function $f(y)$, which limits is 0 when y goes to infinity, such that:

$$c(y) = L_y[\sqrt{2} + f(y)]$$

We have $y = L_x[\beta] = \exp(\beta\sqrt{\log x \log\log x})$.

Let $g(x) = f(y) = f(L_x[\beta])$. The limit of g is also 0 when x goes to infinity. This gives:

$$\begin{aligned} c(y) &= \exp((\sqrt{2} + g(x))\sqrt{\log y \log\log y}) \\ &= \exp\left((\sqrt{2} + g(x))\sqrt{\beta\sqrt{\log x \log\log x}\log\left(\beta\sqrt{\log x \log\log x}\right)}\right) \end{aligned}$$

which gives

$$c(y) = \exp(r(x)\sqrt{\log x \log\log x})$$

where

$$r(x) = (\sqrt{2} + g(x))\frac{\sqrt{\beta\sqrt{\log x \log\log x}\log\left(\beta\sqrt{\log x \log\log x}\right)}}{\sqrt{\log x \log\log x}}.$$

It is easy to see that the limit of $r(x)$ is 0 when x goes to infinity, which gives:

$$c(y) = L_y[\sqrt{2} + o(1)] = L_x[o(1)].$$

A.6 The Proof of $\mathcal{O}(y \cdot \log e)$

τ is the number of integers which are required in order to find a vector which is a linear combination of the others.

In the most simple setting, the public exponent e is prime and the set of vectors with $k+1$ coordinates modulo e is a $k+1$-dimensional linear space; $\tau = k+2$ vectors are consequently sufficient to guarantee that (at least) one of the vectors can be expressed as a linear combination (easily found by Gaussian elimination) of the other vectors.

When e is the r-th power of a prime p, $\tau = k+2$ vectors are again sufficient to ensure that (at least) one vector can be expressed as a linear combination of the others. Using the p-adic expansion of the vectors' coefficients and Gaussian elimination on $k+2$ vectors, we can write one of the vectors as a linear combination of the others.

Finally, the previous argument can be extended to the most general case :

$$e = \prod_{i=1}^{\omega} p_i^{r_i}$$

where it appears that $\tau = 1 + \omega(k+1) = \mathcal{O}(k \log e)$ vectors are sufficient to guarantee that (at least) one vector is a linear combination of the others; modulo each of the $p_i^{r_i}$, the attacker can find a set T_i of $(\omega - 1)(k+1) + 1$ vectors, each of which can be expressed by Gaussian elimination as a linear combination of $k+1$ other vectors. Intersecting the T_i and using Chinese remaindering, one gets that (at least) one vector must be a linear combination of the others modulo e.

Since k is the number of primes below y, we have $k \leq y$ which shows that in any case, the number of integers which are required is always

$$\mathcal{O}(y \log e)$$

A.7 Additional Proof of General Case when $e = \prod_{j=1}^{\omega} p_j^{r_j}$

Denote the target vector $\mathcal{V}$ *((k+1)*-dimensional) .

Step 1 First , we collect $k+1$ of *(k+1)*-dimensional vectors, and solve

$$\mathcal{V} = \sum_{i=1}^{k+1} c_i^{(1)} \mathrm{V}_i^{(1)} \qquad mod\ p_1^{r_1} \tag{A.2}$$

Step 2 Then, for each $p_j^{r_j}, 1 < j \leq \omega$, we continue collecting $k+1$ of *(k+1)*-dimensional vectors to solve the similar equation.

$$\mathcal{V} = \sum_{i=1}^{k+1} c_i^{(j)} \mathrm{V}_i^{(j)} \qquad mod\ p_j^{r_j} \qquad 1 < j \leq \omega \tag{A.3}$$

Step 3 Finally, we get :

$$\mathcal{V} = \sum_{i=1}^{k+1} c_i^{(j)} \mathrm{V}_i^{(j)} \qquad mod\ p_j^{r_j} \qquad 1 \leq j \leq \omega \tag{A.4}$$

by using $1 + (k+1) + (\omega - 1)(k+1) = 1 + \omega k$ of *(k+1)*-dimensional vectors. Applying *Chinese Remainder Theorem*, we can find the linear dependency relation of $\mathcal{V}$ on modulus $e = \prod_{j=1}^{\omega} p_j^{r_j}$.

Hence, the number of vectors which are necessary to express $\mathcal{V}$ as linear combinations of others is $1 + \omega k$.

(Q.E.D)