ESIGN 署名攻撃評価報告書

（株）東芝

2001/01/12

1. はじめに

本文では、ESIGN 署名の詳細評価（攻撃評価）の結果についてまとめる。本文の構成は、２章で提案者による自己評価書の概要をまとめ、以下、３、４、５章でこの評価結果の検証を行う。３章では、プリミティブの安全性に関わるｅ乗根近似問題の困難性について既知の攻撃論文のサーベイとともに考察する。また、４章では、もう１つのプリミティブの安全性に関わる素因数分解に対する困難性について考察する。５章では、プリミティブをベースに構築された署名スキームの安全性に関して考察する。

2. 自己評価書の記述概要

ESIGN 署名に関する自己評価書の内容概略は次の通りである。

2.1 設計方針

ESIGN 署名は以下のような要求条件に答えるために作られた署名目的の公開鍵暗号方式である。

(1) 適当な仮定の下に最強の意味での安全性を保証する理論的証明があること。
(2) RSA 署名や楕円 DSS 署名等の代表的なディジタル署名方式のいずれより優れた性能を保持すること。

これらの要求のうち(1)は以下のように実現されており、(2)は署名生成や署名検証の処理量が RSA 署名他と比べて少ないことが自己評価書に示されている。

2.2 安全性評価

ESIGN 署名は、RSA 仮定の近似版である e 乗根近似仮定とランダムオラクルモデルの下で最強の意味で安全（適応的選択文書攻撃に対し存在的偽造不可）であることが理論的に証明できる。安全性の評価に対するこのような理論的アプローチは、1994 年の Bellare, Rogaway の OAEP や PSS 署名の提案以来、標準的なアプローチとされているものである。

定義 2.1 *Gen* を鍵生成アルゴリズムとする。e 乗根近似問題とは、$pk = \{n,e\} \leftarrow Gen(1^k)$ と $y \leftarrow_R \{0,1\}^{k-1}$ が与えられたとき、$0 \| \| y = [x^e \bmod n]^k$ となるような $x \in (Z/nZ) \setminus pZ$ を見つける問題である。

e 乗根近似問題が難しいという仮定を e 乗根近似仮定と呼ぶ。

e 乗根近似問題については１５年前に ESIGN を発表して以来、様々な研究が行われてきたが、e が４以上の場合については有効な攻撃が発見されておらず、提案者らは素因

数分解以外に有効な攻撃法が無いと予想している。

ESIGN については、自己評価書添付の論文にて次のことが証明されている。

定理 2.2 ESIGN は、e 乗根近似仮定が正しいという条件下で、ランダムオラクルモデルにおいて、適応的選択文書攻撃に対して存在的に偽造不可である。

3. プリミティブの安全性：e 乗根近似問題の困難性

法 n が素因数分解されれば秘密鍵が分かるので ESIGN 署名を偽造できる。素因数分解アルゴリズムについては４章で解説するので、ここでは素因数分解を行わずに ESIGN 署名を偽造する方法に関して、ｅ乗根近似問題の困難性に着目して検討する。

3.1 Brickell らによるアタック

Brickell らは k = 2 の ESIGN に対するアタックを発表している[1]。彼らのアタックの原理は以下のようなものである。x を $n^{1/2}$ に近い整数とする。このとき、$x^2 \bmod n$ は $O(n^{1/2})$ であり、m = 0 の場合の ESIGN 基本関数の検証式を満足する。この原理を任意の m に対して適用できるように、連分数展開を用いて平方根の近似値を求めるようにしたのが Brickell らの方法である。この方法は k = 3 にも容易に拡張できるが、k ≧ 4 に対しては適用できない。

3.2 Vallee らによるアタック

Vallee らは LLL アルゴリズムのような格子基底縮小アルゴリズムを利用した ESIGN に対するアタックを発表している[2]。論文中では k = 2 に対するアタックを記述し、k > 2 に対しても簡単に拡張できるとしている。彼らの方法は以下のような原理に基づくものである。有限体上の 2 次不等式を解くかわりに以下のような有限体上の 2 変数多項式を解こうとする。

$$(ax + b)^2 + c - y \equiv m \pmod n$$

ここで a, b, c は定数であり、求まった y が十分 c に近いとき($y - c < O(n^{2/3})$)、ax + b は正当な署名となる。Vallee らの方法は 2 変数多項式を解くために格子基底縮小アルゴリズムを使用する。

ESIGN では、一つのメッセージに対して少なくとも $O(n^{1/3})$個の正当な署名が存在する。ESIGN では署名生成時に p 以下の乱数を使用するが、p 以下であれば、どの値を使っても署名が生成できる。このような冗長性が Vallee らのアタックを許す原因となっている。

安藤らは、Vallee のアルゴリズムに関する数値的な検討を行っている[5,6]。文献[5]は格子基底の縮小に LLL アルゴリズムを使用し、文献[6]では Lenstra アルゴリズムを使用している。文献[5]によると k = 2 の場合には無視できない確率で ESIGN 署名の偽造に成功しているが、k = 4 に対しては偽造に成功しなかったとのことである。

3.3 Vallee らのアタック再考

Vallee らのアタックは、1)ESIGN の偽造を有限体上の多変数多項式を解く問題とみなすというコンセプトの部分と、2)格子基底縮小アルゴリズムを使って実際に有限体上の多変数多項式を解くという 2 つの部分に分けて考えることができる。Vallee らの研究は 10 年以上前のものであり、最近の様々な研究の進歩を織り込んでいないという問題がある。特に 2)の部分に関しては 1990 年代後半に Coppersmith によるブレイクスルーがあり、近年この結果を応用した革新的な成果が登場しはじめている[7]。例えば、有限体上の多変数多項式を解くのに、Howgrave-Graham 行列を縮小した結果に終結式を利用して解を求めるといったことが考えられる。

Brickel の方法は $k \geqq 4$ に適用できないことが分っているが、Vallee の方法を $k \geqq 4$ に適用できるかどうかは不明である。また、上で述べたように、Vallee の方法は様々な方面に拡張できる可能性が残されている。

Vallee の方法の理論的な限界は分っていないが、Vallee の方法は LLL アルゴリズムを使うため、LLL アルゴリズムの限界を考えれば Vallee らの方法の適用限界を推定できる。並列実行を考えないとすると Vallee らの攻撃にかかる時間計算量は $O(k^3)$なので $O(2^{128})$程度の安全性を保証するためには $k>2^{43}$ 程度に選ばなければならない。並列実効を考えたとすると時間計算量は $O(k^2)$程度にまで下がる可能性があり、同じく 2^{128} 程度の安全性を保証するためには $k > 2^{64}$ 程度に選ばなければならない。空間計算量で考えると $O(k^2)$程度は最低でも必要とされるが、2^{60} 以上の記憶域は物理的に困難であると仮定すると $k > 2^{30}$ 程度あれば安全であることになる。

以上をまとめると、実際には検証されていないが、ｅ乗根近似問題を解く可能性が最も高いのは Vallee らの方法を拡張することである。しかし、どんなに改良を重ねても $k > 2^{30}$ 程度であればｅ乗根近似問題は解けないことになる。

3.4 e 乗根近似問題の困難性のまとめ

以上考察したように、ESIGN 基本関数のベースとなるｅ乗根近似問題の困難性に関してはあまり多くの研究が行われていないのが現状である。最も有効と考えられるのは Valee のアルゴリズムを拡張する方法であるが、それでも $k > 2^{30}$ 程度あれば原理的に安全と予想される。逆に k の値が小さい場合にはｅ乗根近似問題は破られる可能性があり、k の設定には注意が必要と考える。

[1] E. F. Brickell and J. DeLaurentis, “An Attack on a Signature Scheme Proposed by Okamoto and Shiraishi,” Crypto'85, LNCS 218, Springer-Verlag, pp.28-32, 1986.
[2] B. Vallee, M. Girault, P. Toffin, “How to Break Okamoto's Cryptosystem by Reducing Lattice Bases,” Eurocrypt'88, LNCS 330, pp.281-291, 1988.
[3] M. Girault, P. Toffin, B. Vallee, “Computation of Approximate L-th Roots Modulo n and Application to Cryptography,” Crypto'88, LNCS 403, Springer-Verlag, pp.100-117, 1990.
[4] B. Vallee, M. Girault, P. Toffin, “How to Guess L-th Roots Modulo n by Reducing Lattice Bases,” ISSAC-88 and AAECC-6, 1988.
[5] 安藤心, 荒木純道, “LLL アルゴリズムによる ESIGN の偽造確率に対する数値的検討,” 1997 年電子情報通信学会基礎・境界ソサイエティ大会, A-7-15, p.140, 1997.
[6] 安藤心, 荒木純道, “有限環上の2次不等式解法に関する数値的検討,” 1999年電子情報通信学会総合大会, A-7-3, p.231, 1999.
[7] P. Q. Nguyen, J. Stern, “Lattice Reduction in Cryptology: An Update,” ANTS-IV, to appear.

4. プリミティブの安全性：素因数分解に対する評価

ESIGN の安全性は、e 乗根近似仮定の下で、法 $n=p^2q$ の素因数分解に依存している。このため、本章ではプリミティブの安全性評価として、各種素因数分解法でどの程度まで攻撃できるかについて考察する。

まず、以下に素因数分解の主な方法について簡単にまとめておく（表１）。
表１でｎは素因数分解の対象となる数であり、p、q はその素因数とする。
また関数 Lx[u,v]は、次のように定義される：

$Lx[u,v] = \exp((v+o(1))(\log x)^u(\log\log x)^{1-u})$

素因数分解法	計算量	特徴
試行割算法	N	ｎが小さい素因数を持つ場合は有効。
p-1 法、p+1 法	$O(p_{max})$	それぞれ p-1、p+1 が小さい素因数の積に分解するときに有効。
Fermat 法		素因数の差の絶対値が小さいときに有効。
楕円曲線法	Lp[1/2,1.414]	計算量は p のサイズに依存する。
２次ふるい法	Ln[1/2,1.020]	計算量は n のサイズに依存する。
数体ふるい法	Ln[1/3,1.901]	２次ふるい法の一般化

表１.

表１に挙げた素因数分解法のうち、試行割算法による攻撃は、原始的なものであって p、q を十分大きく取ることによって避けることが常識的であり、p-1 法、p+1 法、Fermat 法に対しても、p、q として、それらが有効となる特殊な型の素因数を取らないことによって回避することができる。従って以下素因数の特殊性によらない一般的な素因数分解法を考えることとし、現在最高速のものとして、計算量が n のサイズに依存する数体ふるい法と、素因数のサイズに依存する楕円曲線法を取り上げる。また最近 Boneh らによって開発された $p^r q$ に対する素因数分解アルゴリズムについても取り上げる。

4.1 サイズ依存の素因数分解法（数体ふるい法）

4.1.1 方法概要

数体ふるい法は２次ふるい法の拡張として、90 年に Lenstra 兄弟と Menasse、Pollard[LMP]によって開発されたもので、一般的な素因数分解法としては現在最高速のものである。最近では、99 年に RSA 暗号の 512 ビットの法が数体ふるい法によって素因数分解されたことが記憶に新しい。ここではこのアルゴリズムについて説明するが、一般的なものはかなり複雑になるため、ここでは使われる数体（有理数体上の有限次代数拡大体）が類数１で単数群の構造等もよくわかっている特殊な場合に限定して概要を以下解説することとする。

今 N を素因数分解の対象となる合成数とする。

代数体 K を適当に取り、その整数環を Z_K とする。また K=Q(θ)とするとき、 $[Z_K : Z[\theta]] = f$ とし、$(f, N)=1$ を仮定する。更に T(X)をθの整係数モニック最小多項式とし、T(m) = kN（k は小さい整数）なる整数 m が既知であるとする。

このとき Z[θ]から $Z_N=Z/NZ$ への環準同型φを、φ(θ) = m とすることにより定義できるが、これを次のようにして Z_K に拡張できる：Z_K の任意の元αに対し、fα は Z[θ]の元であるから、$\phi(\alpha) = f^{-1}\phi(f\alpha)$とする（$f^{-1}$ は f の mod N での逆元）。

次に、ノルムが B-smooth であるような（B は正の整数）

$\alpha \in Z_K$ を（Z_K が一意分解整域であるから）素元分解して次の通りとなったとする：

$$\alpha = \prod_{u \in U} u^{\lambda_u} \prod_{g \in G} g^{\mu_g}$$

（ここで U は単数群、G はその下の素数 p が $p \le B$ を満たす素元の集合）

もしφ(α)も B-smooth であったとするならば、

$$\phi(\alpha) \equiv \prod_{p \le B} p^{\nu_p}$$

よって次の合同式が成立する：

$$\prod_{u \in U} \phi(u)^{\lambda_u} \prod_{g \in G} \phi(g)^{\mu_g} \equiv \prod_{p \le B} p^{\nu_p}$$

P を B 以下の素数の集合とすると、上記の型の合同式を|U|+|G|+|P|個以上集めれば、合同式の両辺の対数を取った式達に Z/2Z 上での Gauss の消去法を適用することにより、$s^2 \equiv t^2 (\mathrm{mod}\, N)$ の型の式を得て、２次ふるい法等と同様に(s±t, N)を計算して N の因数を得る。

4.1.2　考察

数体ふるい法は、法 N のサイズにのみ依存しており、その素因数のサイズには依存していない。従って ESIGN の法を RSA の法と同じサイズにしておけば、その困難性は、RSA の法の素因数分解と同等になる。

4.2　因数サイズ依存の素因数分解法（楕円曲線法）

4.2.1　方法概要

楕円曲線法（Elliptic Curve Method；ECM）は 1987 年に Lenstra[L]によって提案された素因数分解法である。方法は次の通り。

N：６と素な合成数とする。これを素因数分解の対象とする。

B：整数

p[i]：大きさの順番で i 番目の素数（i は自然数）。

ｋ：p[k] ≤ B なる最大の整数。p[1], p[2],…,p[k]までの事前計算テーブルがあると仮定する。

［楕円曲線法アルゴリズム］

Input：　合成数 N

Output：　N の因数

Step１：（曲線の初期化）set a ← 0

楕円曲線 E を射影座標(x, y, t)を用いて $y^2t = x^3 + axt^2 + t^3$ とする。

Step２：（初期化）Set x ← (0, 1, 1)、i ← 0

Step３：Set i ← i + 1

If　i > k , set a ← a + 1 and go to step2.

Else , set　q ← p[i], q_1 ← q, $l \leftarrow \lfloor B/q \rfloor$ and go to step2

Step４：（楕円倍数算）While $q_1 \leq l$, q_1 ← q・q_1。

Then　x ← q_1・x の楕円倍数算を実行。

楕円倍数算が常に問題なく行われ、破綻する（t mod N が可逆元でなくなる）ことが起きなければ、Step３に戻る。

Step５：T を mod N の非可逆元とする。

Set　g ← (t, N)

If g < N, output g and terminate

Else , set a ← a + 1 and go to step2

4.2.2 考察

楕円曲線法で素因数が見つけられるのは、p 上の楕円曲線の位数が B を割り切る場合、従って位数が smooth な場合である。漸近的には初めに挙げた表のように最大素因子 p のサイズに依存する。

ここで RSA の法と ESIGN の法を同じサイズに取ったとし、RSA の法の素因子 p に対して ESIGN の法の素因子を 2/3 のサイズに取ったと考え、その計算量の比を考える。即ち Lp[1/2,1.414]/Lp$^{2/3}$[1/2,1.414]を考える。

p は大きい素数なので、

$\log\log(p^{2/3}) = \log(\log p + \log(2/3)) \approx \log(\log p - 0.8) \approx \log\log p$ と近似すると、

$$L_p[1/2,1.414]/L_{p^{2/3}}[1/2,1.414] \approx \exp(1.414(1-\sqrt{2/3})(\log p)^{1/2}(\log\log p)^{1/2})$$

であって、この比は p が大きくなればなるほど大きくなる。例えば logp=512 とすると、この比は 18779337.8 程度になる。従って RSA の法と ESIGN の法を同じサイズにとっても楕円曲線法に対する強度は RSA の方が強く、この差は法のサイズが大きくなるほど顕著になる。楕円曲線法に対して同じ強度とするためには ESIGN の法のサイズは RSA の法の 3/2 倍にすることが必要である。RSA 社は 99 年の時点で RSA が安全であるためには、法のサイズは 768 ビット必要と述べている。これに従えば ESIGN が安全であるためには、法のサイズは 1152 ビット必要ということになる。

4.3 N= $p^r q$ に対する素因数分解法（LFM）

4.3.1 方法概要

この方法は N= $p^r q$ という特殊な形の合成数に対して、Boneh 等[B]によって 99 年に提案した方法であり、LFM（Large Factoring Method）と呼ばれている。以下方法概要を述べる。

f(x) = (x + P)^r とおく。x_0＝p－P とおくと、 $f(x_0) \equiv 0 (\mathrm{mod}\ p^r)$ なので、

f(x)の mod p^r の解 x_0 を $|x_0| < X$（X はある整数）で求めることを考える。
しかしながら、p^r が未知であるため、代わりに $p^r q$(= N)を用いることを考え、更に f(x)を因数に持つ多項式を用いて次の命題での h(x)を構成することを考える：

命題 3.1. h(x)を d 次整係数多項式とし、次を仮定する：

a) ある正数 r, m に対し、 $h(x_0) \equiv 0 (\mathrm{mod}\ p^{rm})$ （$|x_0| < X$ ）

b) $\|h(xX)\| < p^{rm}/\sqrt{d}$ （ここで $h(x) = \sum_i a_i x^i$ に対し ）

このとき h(x_0) = 0 となる。

この h(x)を作るため、条件 a)を満たす多項式（f(x)を因子に持つ式の変形）をいくつか取って条件 b)を満たすように線形結合することを考える。具体的には次のアルゴリズムを実行して h(x)を構成し、その整数解を使って N の因数 p を求める。

［LFM アルゴリズム（概要)］

Input： 合成数 N、素因数 p,q のサイズ k、正整数 r, c（$N = p^r q$, $q < p^c$ とする）

Output： N の因数

Step 0： $\varepsilon = \frac{1+c}{r+c}$, $X = \left\lfloor 2^{(1-\varepsilon)k} \right\rfloor + 1$ を計算する。Set j = 1

Step 1：Set $P_j = 2^k + j X$

Step 2：多項式の列 $g_{i_1,i_2}(x) = N^{m-i_2} x^{i_1} (x+P)^{ri_2}$ を計算し、それらの係数を並べた

行列 A を作る（$0 \le i_1 \le r-1, 0 \le i_2 \le m-1$ or $0 \le i_1 \le d-mr-1, i_2 = m$）。

Step 3：A に LLL アルゴリズムを実行し、LLL 縮小基底の中の最短ベクトルを取り出して、その係数から多項式 h(x)を作る。

Step 4：h(x)が整数解を持つかどうか調べる。

もし持っていなければ j → j + 1 として Step1 へ戻る。

Step 5：h(x)の整数解 x_0 が($x_0 + P_j$) | N を満たしていれば $p = x_0 + P_j$ を出力して終了。

満たしていなければ、j → j + 1 として Step1 へ戻る。

4.3.2 考察

このアルゴリズムの計算量は Boneh 等の論文[B]の定理２によると、

$$\exp\left(\frac{c+1}{r+c} \cdot \log p\right) \cdot O(LLL)$$

（ここで exp(n) = 2n, また O(LLL)は、成分のサイズが O(rlogN)、次数 $O(r^2)$の行列についての LLL アルゴリズムの計算量。）である。

(1) O(LLL)について。

Cohen[C]によれば、次数 n、成分のノルムの２乗の最大値 B に対し、LLL の計算量は $O(n^6 \log B)$である。従って $O(LLL) = O((r^2)^6 \log((r \log N)^2)) = O(r^{12} (\log N)^2)$。

このため、計算量は exp(*)の部分が支配的となる。

(2) exp(*)について。

c があまりにも大きい（q が p に比して大きい）場合、exp(*)は指数時間のアルゴリズムになってこのアルゴリズムは適用できなくなる。しかしながら p 又は q のサイズがあまりにも小さければ、他の、例えば楕円曲線法のターゲットとなるであろう。このため p 又は q のサイズが一定以上で、しかもサイズの増加を抑えようとすれば、必然的に p、q のサイズが近くなり、c が小さくなる。そこで例えば c = 1 で p、q のサイズがほぼ同じケースを考えてみる。このとき r の大きさによって計算量がどうなるかについては Boneh 等[B]で議論されており、以下これに従って説明すると、

$\frac{c+1}{r+c} = O(\frac{1}{r})$ となるため、r が大きくなればなるほど、このアルゴリズムは高速になって

・$r \approx \log^{1/2} p$ ならば、計算量はほぼ $\exp(\log^{1/2} p)$ となり、楕円曲線法よりやや高速

・r = ε log p（ε は固定値）なら、このアルゴリズムは多項式時間まで達する。

しかしながら、r が小さいときは exp(*)は指数時間であるため、ESIGN のように r=2 の場合には LFM アルゴリズムは有効ではないことがわかる。

4.4　プリミティブの安全性評価のまとめ

ESIGN の安全性は、e 乗根近似仮定の下で、法 $n=p^2q$ の素因数分解に依存している。

３での検討から判断すると素因数分解法の中で有力な方法は、数体ふるい法と楕円曲線法である。数体ふるい法は計算量が法のサイズに依存するため、ESIGN の法のサイズを RSA の法のサイズと同じに取れば、この方法に対して ESIGN は RSA と同じ安全性を持つ。楕円曲線法の計算量は最大素因子のサイズに依存するため、ESIGN の法のサイズを RSA の法のサイズと同じに取った場合、楕円曲線法に対して ESIGN は RSA より安全性が劣る可能性がある。しかしながら、現在最速な素因数分解法は数体ふるい法であるため、この方法に対して安全であれば、素因数分解に対して安全と考えられる。

結論として ESIGN は、法のサイズを RSA の法のサイズと同じに取った場合、安全と判断する。

参考文献

[LMP] Lenstra, A.K., Lenstra, H.W.Jr., Manasse, M.S. and Pollard, J.M.：”The Number Field Sieve”, Proc. of STOC, pp.564-572(1990)

[L] Lenstra, H.W.Jr ： ”Factoring Integers with Elliptic Curves”, Annals of Math.126(1987),649-673

[C] H.Cohen：A Course in Computational Algebraic Number Theory, GTM 138, Springer-Verlag

[B] D.Boneh, G.Durfee and N. Howgrave-Graham： “Factoring $N = p^r q$ for Large r”, Proc of Crypto'99, LNCS 1666, Springer-Verlag, pp.326-337

5. スキームの安全性評価

本章では、プリミティブである e 乗根近似仮定は正しい(3 章で評価済み)という条件のもとでスキームの安全性を評価する。

5.1 適応的選択文書攻撃に対する安全性について

提案方式は「e 乗根近似仮定が正しいという条件下で、ランダムオラクルモデルにおいて、適応的選択文書攻撃に対して存在的に偽造不可である」(自己評価書 定理 2.2)と主張しており、その理論的な証明は添付資料に記述されている。

証明には背理法を用いており、ランダムオラクルモデルにおいて、適応的選択平文攻撃により提案方式の署名の存在的偽造が可能なアルゴリズム F が存在すると仮定すると、そのアルゴリズム F をサブルーチンとして用いることにより、e 乗根近似問題を解くアルゴリズム B が構成可能であることを示している。この証明方法は、Bellare,Rogaway により提案された FDH-RSA 署名方式の安全性証明方法と同様なアプローチであり、証明とその結果について誤りはないと思われる。

しかし、この証明方法の様に署名偽造問題をプリミティブな問題へ還元する際、より tight な確率(理想的には確率 1)で還元する必要があり、その時に限り署名偽造問題はプリミティブな問題と同程度の安全性をもつといえる。また、ランダムオラクルへの query 数を q_h、署名オラクルへの query 数を q_s としたとき、現実世界においてランダムオラクルの代わりにハッシュ関数を用いた場合、q_h は敵の計算能力にのみ依存する値であり、q_s に比べて q_h は十分大きな値となる可能性があることが指摘されている。

FDH-RSA 署名の場合、FDH-RSA 署名方式の偽造成功確率が RSA の解読成功確率の q_h+q_s 倍になることから、この 2 つの問題には差があるのではないかと懸念されている。そこで Bellare,Rogaway は、偽造成功確率が q_h,q_s に依存しない値となるような署名方式 PSS を提案している。また FDH-RSA については RSA の解読成功確率から q_s 倍程度の確率で還元可能であることが 2000 年 Coron により示されている。しかし FDH-RSA と PSS では安全性の還元効率の面でまだ差があると考えられている。

一方、提案方式の偽造成功確率は e 乗根近似問題を解くアルゴリズムの成功確率の q_h 倍であり、上記の理由により 2 つの問題には差があると考えられる。Coron と同様の手法によ

り q_s 倍程度の確率で還元可能であることを示すことは可能と思われるが、それでも、例えRSAとe乗根近似問題が同強度の問題だとしても、PSSと提案方式の安全性については還元効率の面で差があると思われる。

以上まとめると、RSAとe乗根近似問題が同強度の問題と仮定した場合、提案方式はFDH-RSAと同程度の安全性をもつと言えるが、PSSとは差があると考えられる。

5.2 安全性証明の前提(ランダムオラクルモデル)について

ランダムオラクルモデルとは、1993年Bellare,Rogawayにより提案された安全性証明の一手法である。この手法では、方式で用いられるハッシュ関数を理想的なランダム関数と仮定したモデル(ランダムオラクルモデル)において安全性を証明することにより、実用的なハッシュ関数を用いた方式に対する安全性の指標を与えている。ランダムオラクルモデルを用いた安全性の証明手法は、Bellare, Rogawayにより提案されたOAEP,PSS署名の安全性証明に用いられて以来一般的な証明手法となっており、ランダムオラクルモデルにおいて安全ならば、実用的なハッシュ関数を用いた方式も安全であると広く信じられている。

実用的なハッシュ関数を用いた方式と、そのハッシュ関数を理想的なランダム関数と仮定した方式の間に大きな隔たりがあることは事実であるが、提案者が主張するように「もしこのような仮定の下で安全性の証明のついた方式に対して何らかの攻撃があれば、それはそのハッシュ関数のランダム関数にない性質(非ランダム性)の一つが見つかったことを意味する」(仕様書 2 章)ものであり、方式の安全性は実際に用いるハッシュ関数の安全性に依存すると考えられる。提案方式では、ハッシュ関数の例としてSHA-1からの構成したハッシュ関数を示している(仕様書8章)。この構成法は、Bellare,RogawayによりOAEPを構成する際に示されたものであり、安全性に問題はないと思われる。

一方、1998年Canettiらにより、ランダムオラクルモデルにおいて安全性が証明できても、どのような計算量的仮定の下でも、安全な方式を実現するハッシュ関数を構成できないような暗号プロトコルを示している。この例についても、提案者の主張通り「極めて人工的に作られたものであり、自然に構成された方式に対しては、Canett たちの結果に基づく懸念は意味がない」(仕様書p.2脚注) と思われる。よって、自然に構成された提案方式について特に影響はないと考えられる。

以上より、ランダムオラクルモデルにおいて安全性の証明がなされた提案方式について、適切に構成したハッシュ関数を用いれば、安全性に問題はないと思われる。

5.3　ハッシュ関数の衝突探索に関する安全性

ESIGN 署名で利用されるハッシュ関数 H は値域のサイズが、2^pLen-1 と規定されている。ここで pLen は法ｎのビット長の 1/3 である。従って、$O(n^{1/6})$程度の試行によるバースディアタックによりハッシュ値の衝突ペアが得られ、これが成功すると ESIGN 署名が偽造されることになる。しかし、仕様書では、ｎのサイズは 1152 ビットで性能評価されており、これを推奨値と考えると、バースデイアタックに必要な試行は O(2^192)となり、現実的な攻撃法とはならない。

6. まとめ

以上、ESIGN 署名の安全性について検討した。プリミティブである ESIGN 基本関数はｅ乗根近似問題と素因数分解問題をベースとしており、素因数分解に関しては提案者の考察のとおり法が 1024 ビット以上であれば困難と考えられる。一方、ｅ乗根近似問題は多くの研究がないが、k > 2^{30} 程度で原理的に安全と思われるが、k が小さい場合には問題になる可能性があり、設定には注意すべきである。

一方、ESIGN 署名のスキームとしての安全性は提案者の評価のとおり、e 乗根近似仮定が正しいという条件下で、ランダムオラクルモデルにおいて、適応的選択文書攻撃に対して存在的に偽造不可といえる。但し、ESIGN 署名の e 乗根近似問題への還元効率を RSA-PSS の e 乗根問題への還元効率と比較するとあまり効率的とはいえない。