HITACHI
Inspire the Next

# RSA暗号秘密指数の中央部分の未知ビットのLLLによる推定

2008/07/05

株式会社 日立製作所
中央研究所

遠藤　隆

HITACHI
Inspire the Next

# Contents

# 1-1　RSA暗号の概要(1)

HITACHI
Inspire the Next

大きな素数p, qを用意し，秘密情報とする。p, qの積Nと、(p-1)および(q-1)と互いに素となる指数eを公開する。Nと同程度のビット長の平文Mがあたえられたとき，暗号文Cは

$$C = M^{e} \bmod N$$

で計算される。
暗号文Cの復号化は、

$$d \equiv e^{-1} \bmod \varphi(N) \equiv e^{-1} \bmod (p-1)(q-1)$$

で定義される秘密指数dを用いて、

$$M \equiv C^{d} \bmod N \equiv \left(M^{e}\right)^{d} \bmod N \equiv M^{ed} \bmod N \equiv M \bmod N$$

と復号化することができる

HITACHI
Inspire the Next

## RSA暗号へのアタック

■CのNを法としたe乗根から求める。

C, N, eが与えられているとき，次式を満たすMを求める。

$$C = M^e \bmod N$$

→多項式時間で解くアルゴリズムは知られていない。

■Nを素因数分解し，φ(N)を求め，eより秘密指数dを求める

$$d \equiv e^{-1} \bmod \varphi(N) \equiv e^{-1} \bmod (p-1)(q-1)$$

大きな素数の積の素因数分解を多項式時間で解くアルゴリズムは知られていない。

⇒十分な鍵ビット長が得られていれば、安全か？

HITACHI
Inspire the Next

## ハードウエアに暗号アルゴリズムを実装した場合の懸案事項

- 暗号処理を実際に実装する場合，物理的な制約から，電流，電磁波，実行時間などに情報が漏洩する。
  - 1024[bit]長の演算は一度に実行できない
    - 処理を部分に分けて実行しなければならない
    - データや演算の違いが消費電流に現れる
    - 実行内容の違いが実行時間に現れる
    - etc.

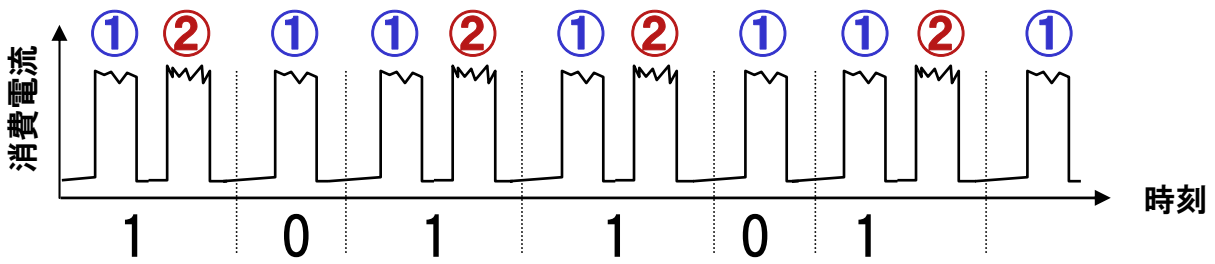
# 2-2　サイドチャネルアタック(2)

HITACHI
Inspire the Next

- ベキ乗剰余演算
  - 指数部をビット毎に分けて演算　$d = \sum_{i=0}^{n-1} d[i] \cdot 2^i$

*Calculate* $x^d \bmod N$

```
result = 1
for i:=n-1 down to 0 do
    result := result² mod N          ・・・①
    if (d[i]==1) then
        result := result * x mod N   ・・・②
    end if
end for
return result
```



# 2-3　サイドチャネルアタック(3)

HITACHI
Inspire the Next

- CRT(Chinese Remainder Theorem)アルゴリズム
  - mod p, mod q 上で演算し，それぞれの結果を統合

**事前計算**

$$d_p = d \bmod (p-1)$$
$$d_q = d \bmod (q-1)$$
$$V = p^{-1} \bmod q$$

*Calculate* $x^d \bmod N$

$x_p = x \bmod p$
$x_q = x \bmod q$
$S_p = x_p^{d_p} \bmod p$
$S_q = x_q^{d_q} \bmod q$
$m = \{(S_q - S_p) \cdot V \bmod q\} \cdot p + S_p$
return S

秘密素数が
演算に陽に現れる

# 3-1　LLL法を用いたアタック(1)

HITACHI
Inspire the Next

## Theorem (Howgrave-Graham)

f(x)を1次でモニックな式とする。Nは既知で，bはNの未知の約数とする。
また，$b=N^{\beta}$で，$\beta<1$である。このとき，
方程式

$$f(x) = 0 \mod b$$

に対して，$|x_0| < N^{\beta^2}$となる全ての解を log N の多項式時間で求めることができる

# 3-2　LLL法を用いたアタック(2)

RSAの公開モジュラスNを素因数分解する場合，素数p,素数qがほぼ同じサイズである場合，$\beta$ =1/2であるので，bをpとおき、Pを素数pの上位部分の既知の値とし、$x_0$を未知の値とすると、

$$f(x) = (P + x_0) \bmod p = 0$$

$$\left|x_0\right| < N^{\frac{1}{4}}$$

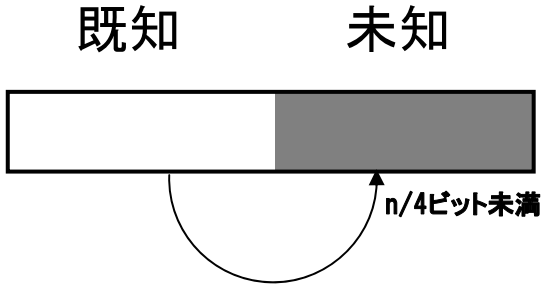
となる。これは、未知部分のビット長が、モジュラスNのビット長nの1/4以下であれば、多項式時間で求められることを示す

⇨ CoppersmithのHigh Bit knownの素因数分解

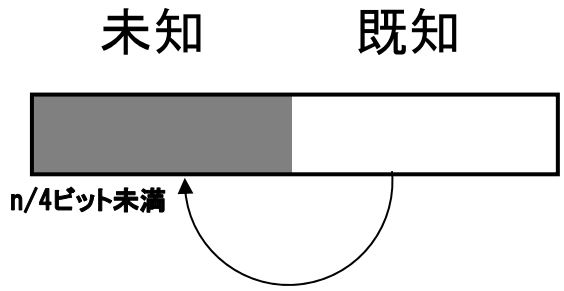
# 3-3　LLL法を用いたアタック(3)

HITACHI
Inspire the Next

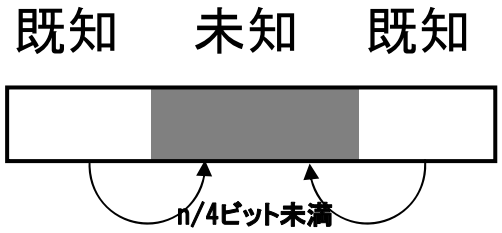
## 秘密素数の一部が既知である場合のRSAのNの素因数分解




Coppersmith(1996)




Boneh, Durfee and Frankel(1998)




本発表

HITACHI
Inspire the Next

## 最上位が既知の場合

秘密素数pの上位ビット側の既知の値をPとすると、

$$f(x_0) = (P + x_0) \bmod p = 0$$

$f(x_0) = 0 \bmod p$ より $^{\forall} x_0 \le X$ に対して $g(x_0) = 0$ over $\mathbf{Z}$ を構成

(1). 未知ビット長に応じて、mを決定

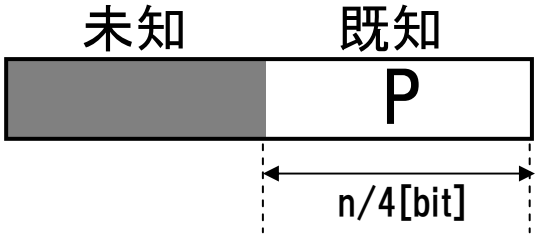
(2). $g_{i,k}(x_0) = N^{m-k} x^i f^k(x_0)$ としたとき、以下の多項式の係数行列を構成し、

$$g_{0,k}(Xx) \quad | k = 0,\ldots,m-1$$
$$g_{m+i,m}(Xx) \,|\, i = 0,\ldots,m$$

LLLアルゴリズムにより、リダクションを行う。得られる最小ノルムの
ベクトルの係数より、$g(x_0) = 0$ を解く。

HITACHI
Inspire the Next

## 最下位側が既知の場合




秘密素数pの下位ビット側の既知の値をPとし、$R=2^{n/4}$とすると、

$$(Rx_0 + P) \bmod p = 0$$

$a \equiv R^{-1} \bmod N$ を両辺にかけると,

$$(aRx_0 + aP) \bmod p = (x_0 + aP) \bmod p = 0$$

ここで,

$$f(x_0) = (x_0 + aP) \bmod p = 0$$

とおく。

# 3-6 Boneh Durfee Frankel method

$f(x_0) = 0 \bmod p$ より $^{\forall} x_0 \leq X$ に対して $g(x_0) = 0$ over $\mathbf{Z}$ を構成

(1). 未知ビット長に応じて、mを決定

(2). $g_{i,k}(x_0) = N^{m-k} x^i f^k(x_0)$ としたとき、以下の多項式の係数行列を構成し、

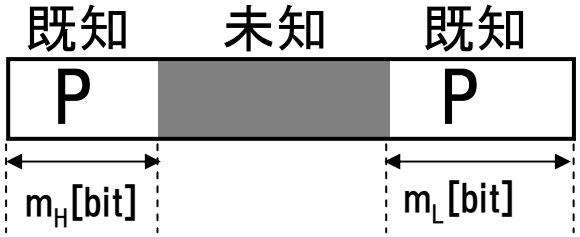
$$g_{0,k}(Xx) \quad | k = 0,\ldots,m-1$$

$$g_{m+i,m}(Xx) \,| \, i = 0,\ldots,m$$

LLLアルゴリズムにより、リダクションを行う。得られる最小ノルムの
ベクトルの係数より、$g(x_0) = 0$ を解く。

HITACHI
Inspire the Next

## 最下位側および最上位が既知で、中間が未知の場合




$m_H+m_L>n/4$

秘密素数pの下位ビット側の既知の値をPとし、$R=2^{m_L}$とすると、

$$(Rx_0 + P) \bmod p = 0$$

$a \equiv R^{-1} \bmod N$ を両辺にかけると,

$$(aRx_0 + aP) \bmod p = (x_0 + aP) \bmod p = 0$$

ここで,

$$f(x_0) = (x_0 + aP) \bmod p = 0$$

とおく。

# 4-2 Proposed method

HITACHI
Inspire the Next

$f(x_0) = 0 \bmod p$ より $\forall x_0 \le X$ に対して $g(x_0) = 0$ over $\mathbf{Z}$ を構成

(1). 未知ビット長に応じて、mを決定

(2). $g_{i,k}(x_0) = N^{m-k} x^i f^k(x_0)$ としたとき、以下の多項式の係数行列を構成し、

$$g_{0,k}(Xx) \quad | k = 0,\ldots,m-1$$

$$g_{m+i,m}(Xx) \,| \, i = 0,\ldots,m$$

LLLアルゴリズムにより、リダクションを行う。得られる最小ノルムの
ベクトルの係数より、$g(x_0) = 0$ を解く。

# 5-1　数値例(1)

HITACHI
Inspire the Next

## (1). 秘密素数の中央部が未知の場合

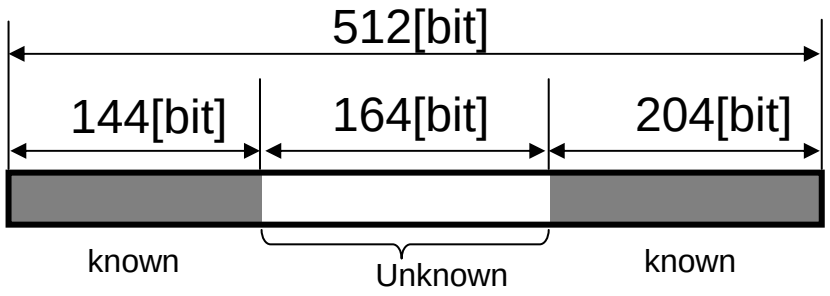
秘密素数p，qおよびpの既知部分をPとする。pqの積をNとし、既知とする。

p =
fa9a890577fbce39194133aa9d936c98eac5472bd11da3f0dc51ba5731098c24
4abec8701a28dc4cb42546c7532bf07a7860ae2bb28266ac21393f6d511390ff

q=
ce3bf64b9706c7043c76c0efa21a835370daecaad2a3da331f4cadc282a8a61f
4441a702347b450383580739c0c4a6d341ec4550425f4db4a987c2d9d6832c5b

P=
fa9a890577fbce39194133aa9d936c98eac5000000000000000000000000000
000000000000c4cb42546c7532bf07a7860ae2bb28266ac21393f6d511390ff

ここで、$X = 2^{164}$, $R = 2^{204}$ , $a = R^{-1} \bmod pq$ とする



# 5-2　数値例(1)

HITACHI
Inspire the Next

## Lattice行列

$$
\begin{array}{cccccccccccc}
1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 & x^9 & x^{10} & x^{11} \\
N^5 & & & & & & & & & & & \\
aPN^4 & XN^4 & & & & & & & & & & \\
(aP)^2N^3 & 2aPXN^3 & X^2N^3 & & & & & & & & & \\
(aP)^3N^2 & 3(aP)^2XN^2 & 3aPX^2N^2 & X^3N^2 & & & & & & & & \\
(aP)^4N & 4(aP)^3XN & 6(aP)^2X^2N & 4aPX^3N & X^4N & & & & & & & \\
(aP)^5 & 5(aP)^4X & 10(aP)^3X^2 & 10(aP)^2X^3 & 5aPX^4 & X^5 & & & & & & \\
0 & (aP)^5X & 5(aP)^4X^2 & 10(aP)^3X^3 & 10(aP)^2X^4 & 5aPX^5 & X^6 & & & & & \\
0 & 0 & (aP)^5X^2 & 5(aP)^4X^3 & 10(aP)^3X^4 & 10(aP)^2X^5 & 5aPX^6 & X^7 & & & & \\
0 & 0 & 0 & (aP)^5X^3 & 5(aP)^4X^4 & 10(aP)^3X^5 & 10(aP)^2X^6 & 5aPX^7 & X^8 & & & \\
0 & 0 & 0 & 0 & (aP)^5X^4 & 5(aP)^4X^5 & 10(aP)^3X^6 & 10(aP)^2X^7 & 5aPX^8 & X^9 & & \\
0 & 0 & 0 & 0 & 0 & (aP)^5X^5 & 5(aP)^4X^6 & 10(aP)^3X^7 & 10(aP)^2X^8 & 5aPX^9 & X^{10} & \\
0 & 0 & 0 & 0 & 0 & 0 & (aP)^5X^6 & 5(aP)^4X^7 & 10(aP)^3X^8 & 10(aP)^2X^9 & 5aPX^{10} & X^{11}
\end{array}
$$

**0**

# 5-3　数値例(1)

HITACHI
Inspire the Next

$$
\begin{aligned}
h(x) = {} & +0 \cdot x^{11} \\
& -597583304066030445127000 \cdot x^{10} \\
& +16332246186698069375211552734180506458840136427800793816755950729574486571 \cdot x^{9} \\
& +31020099783938061018230030412805042102888584362202211098858711721702043255063826048608399612060253286194880585817787308053 \cdot x^{8} \\
& -70142813835532297196016379686257517385923348479349326985617399727754704880350254873171289951808660950823928418459839209261841461865004337637408445909618700953526618106500850 \cdot x^{7} \\
& -14102962379348341749780049564609359822671225733624239943413477012588827448771061519967969481817710508342221986662505175022605260322916623031833834346407140477045782190604840475656343207207511358515946373891612583511538443 2 \cdot x^{6} \\
& +88893817256222895500774017065415755477822149114179411579267226570354994406160436670252253128304706353684774156514790900856247861253890887567966557220782793799961493078612270229538177898474003900420370780387813746695006457918365070505314289772796652151597065947766271943335 \cdot x^{5} \\
& +16475179265996210954719068524660560240514810064353344995060900759360037329281139812994422833120077106145923554718829200144564904839489938641108904009827783407291615440954729332561374907668145677233415300327616282147417752669525549491383724411401531000952773794969693505653924379138802220253395381201100103162115457884919 0 \cdot x^{4} \\
& -24887480133810125751275175525320449525362919168603887936848530851712806536352672600676209838471303148573445306419427703897289769101812649122818824815703436332439511275775751201243014061649841703701545366834596562546906420239970145254773423185364018994522875924377835692857162272441430716236537478142623573970457662896660194223412932535647099342124292973596161869936882307223634865005572652739456005156426846935763025696175979560335198798588281586797649415310807901455736294629227179422157863712611243854747070817598569063902479592225020783162288894025504208323620970978883047379408047745386007566675600700982051903098600993430114555872341524686369911731331056685711852419325101291071802425215439089813884091183436098562227152560727550230675508716122434326769817848894458797788008004680252 61 \cdot x^{2} \\
& +11034382700543328943165743517162855318173716551108568125049855318446186914232367376168601281283726294032254621189671078327586440731923620188988679482153282112017276556405655405108738552513402577758404581160105717018753922834981005585039531064895421404653503107404226701687443047758110272836815828917314301978232272628432558145458392327716905166753662758734688771690768439303553459851565795800925804772828239644203265691174210990191237603465890161995322584919745598981318131 \cdot x^{1} \\
& -99264477561284700726108454153290294446055037688295122161632149199441118458712171556723162345049431472807979397257413079595390318991286657856506468747607058387486427392036388853741735092921686076069670612817853208881025951612024429791882029482552950233707619836537023532983162813527046364785009875452173357935530339091777940471168503772872100257194495199987626140289859426513499008710999446647693677378994489731416040047288719990183530506561608170290557140878399043899037606440148496128903129925685602988762620267471 30
\end{aligned}
$$

# 5-4　数値例(1)

HITACHI
Inspire the Next

ニュートン法により根を求め、$x_0$を得る。

```
x0 = 472bd11da3f0dc51ba5731098c244abec8701a28d(16)
```

pの近似値Pで0となっていた部分が得られる。

```
p =
fa9a890577fbce39194133aa9d936c98eac5472bd11da3f0dc51ba5731098c24
4abec8701a28dc4cb42546c7532bf07a7860ae2bb28266ac21393f6d511390ff
```

LLLによるリダクションに　2.672[秒]
ニュートン法による求根に 0.047[秒]

# 5-5　数値例(2)

HITACHI
Inspire the Next

## (2). 秘密指数の中央部が未知の場合

p =
b992f7a82717d82b0ac87c36985ab32144f6933a53297616540e59a43849ed18
e710ed75c7a6c7121d31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7(16)
q =
80213cea31f32a4ac0a0c6aa45ed40871b9c8cebfe087e9c15eef46fa93e4240
e520696c2550f88a623ef1c832f5a86116514788dde52e4cd352cee5c99d7409(16)
d =
167a8da7c86245c681c7b744911a6c0c66954f07f5dd4691046adbe1beeaa225
1d9b5c92c7d740516d72de162410c1f13145d76a163bd4fda4511fad64bc95b2
371a3068694f306e92852349a343636611eb314e7055fd13c80160f943a2b7bf
edadc563ff95e45295a5ea34cf281b5564857345f9eb7f5060c23bd582b08ef1(16)

公開指数
e =65537

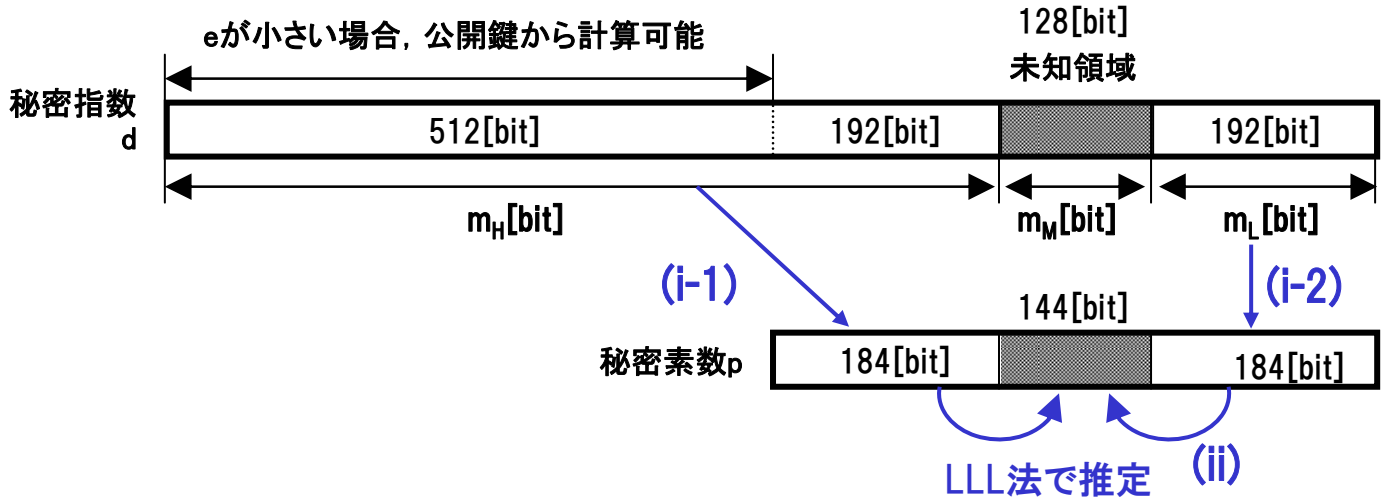
既知秘密指数値
$d_H$ =
167a8da7c86245c681c7b744911a6c0c66954f07f5dd4691046adbe1beeaa225
1d9b5c92c7d740516d72de162410c1f13145d76a163bd4fda4511fad64bc95b2
371a3068694f306e92852349a343636611eb314e7055fd130000000000000000
0000000000000000000000000000000000000000000000000000000000000000(16)
$d_L$ =95a5ea34cf281b5564857345f9eb7f5060c23bd582b08ef1(16)

# 5-6　数値例(2)

HITACHI
Inspire the Next

## 推定手順

(i)　秘密指数の既知部分から、秘密素数の対応部分の値を求める。
(ii)　秘密指数の未知部分に対応する秘密素数の未知部分をLLL法で推定



# 5-7　数値例(2)

HITACHI
Inspire the Next

## (i-1)秘密指数dの上位部$d_H$より，秘密素数の上位部分を求める

秘密指数$y$の上位部分が既知の場合，秘密素数をp, q，公開指数をe，公開モジュラスをN(= pq) とすると，

$$ed = k(p-1)(q-1)+1$$

ここで，kはk<e を満たす整数。この式から、(p+q)は

$$(p+q) = N+1-\frac{ed-1}{k}$$

となる。s=(p+q) とおくと、

$$p = \frac{(p+q)+(p-q)}{2} = \frac{s+\sqrt{(p-q)^2}}{2} = \frac{s+\sqrt{(p+q)^2-4pq}}{2} = \frac{s+\sqrt{s^2-4N}}{2}$$

となる。kの値は，dの上位の既知ビット部分$d_H$より求まる。

$$k = 1+\left\lfloor \frac{ed_H-1}{N} \right\rfloor$$

数値例の場合，$k = 15861$ となる。

# 5-8　数値例(2)

HITACHI
Inspire the Next

## (i-2)秘密指数dの下位部$d_L$より，秘密素数の下位部分を求める

dの下位から$m_L$[bit]分の値を$d_L$が既知の場合，

$$ed_L \equiv 1 + k(N - p - N/p + 1) \bmod 2^{m_L}$$

であるので，

$$kp^2 + (ed_L - k(N+1) - 1)p + kN \equiv 0 \bmod 2^{m_L}$$

である。ここで，

$$b = (ed_L - k(N+1) - 1)$$

と置き，$k_{odd}$を奇数整数として

$$k = k_{odd} 2^t$$

とすると，2次方程式の解の公式より，pの下位のビット候補得られる

$$p \equiv (-b \pm \sqrt{b^2 - 4k^2 N}) \cdot k_{odd}^{-1} \cdot 2^{-t} \bmod 2^{m_L - t}$$

# 5-9　数値例(2)

HITACHI
Inspire the Next

## (i) 秘密素数の$d_H$, $d_L$に相当する部分を推定

### (i-1) 秘密素数上位推定値

```
p′ =
 b992f7a82717d82b0ac87c36985ab32144f6933a532976 20c1caed16c1e4047e
 f3c7d72fe30b5232638a4f4529948a83d8667ba29b693fc4850b9f5542e85760(16)
q′ =
 80213cea31f32a4ac0a0c6aa45ed40871b9c8cebfe087e9 4e29c5f0ff3f30d05
 a3d6ac2032039a6fef6492e6d4a5c4a09c801584ee9eb8c49b194a88c55072d3(16)
```

最上位ビットから184[bit]分の秘密素数が得られた。

### (i-2) 秘密素数下位推定値

```
result[0] =623ef1c832f5a86116514788dde52e4cd352cee5c99d7409(16)
result[1] =e23ef1c832f5a86116514788dde52e4cd352cee5c99d7409(16)
result[2] =a23ef1c832f5a86116514788dde52e4cd352cee5c99d7409(16)
result[3] =223ef1c832f5a86116514788dde52e4cd352cee5c99d7409(16)
```

} qの下位

```
result[0] =1d31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7(16)
result[1] =9d31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7(16)
result[2] =dd31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7(16)
result[3] =5d31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7(16)
```

} pの下位

秘密素数の最下位から184[bit] 分が得られた。

# 5-10 数値例(2)

HITACHI
Inspire the Next

## (ii) 秘密素数p, q の残る部分の推定は，秘密指数の中央部が未知の場合と同様。

p, qの推定値が上位で2通り，下位で2通り現れ、組み合わせると4通りのP，Qの中央部分が不明な値が得られる。そのうちの2通りで、P， Qの正しい値が復元される。

**P'=**
**b992f7a82717d82b0ac87c36985ab32144f6933a5329760000000000000000000**
**0000000000000000031a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7**

**Estimated_P=**
**b992f7a82717d82b0ac87c36985ab32144f6933a532976[16540e59a43849ed18](#)**
**[e710ed75c7a6c7121d](#)31a4e80ea7a9eb284e5eeb0334694ca6217d50f157aaa7**

**P'=**
**80213cea31f32a4ac0a0c6aa45ed40871b9c8cebfe087e000000000000000000**
**0000000000000000003ef1c832f5a86116514788dde52e4cd352cee5c99d7409**

**Estimated_P=**
**80213cea31f32a4ac0a0c6aa45ed40871b9c8cebfe087e[9c15eef46fa93e4240](#)**
**[e520696c2550f88a62](#)3ef1c832f5a86116514788dde52e4cd352cee5c99d7409**

# 6-1　計算時間

HITACHI
Inspire the Next

LLL法を用いた素因数分解は、次の2つの処理に分けられる。

(1). LLL法によるLattice Reduction処理
(2). ニュートン・ラプソン法による求根

このうち、(1)の処理が占める割合がほとんどを占める。
LLL法のLattice Reductionの計算量は，モジュラスNが一定であれば、Lattice行列の行数の5乗に比例することが知られている。

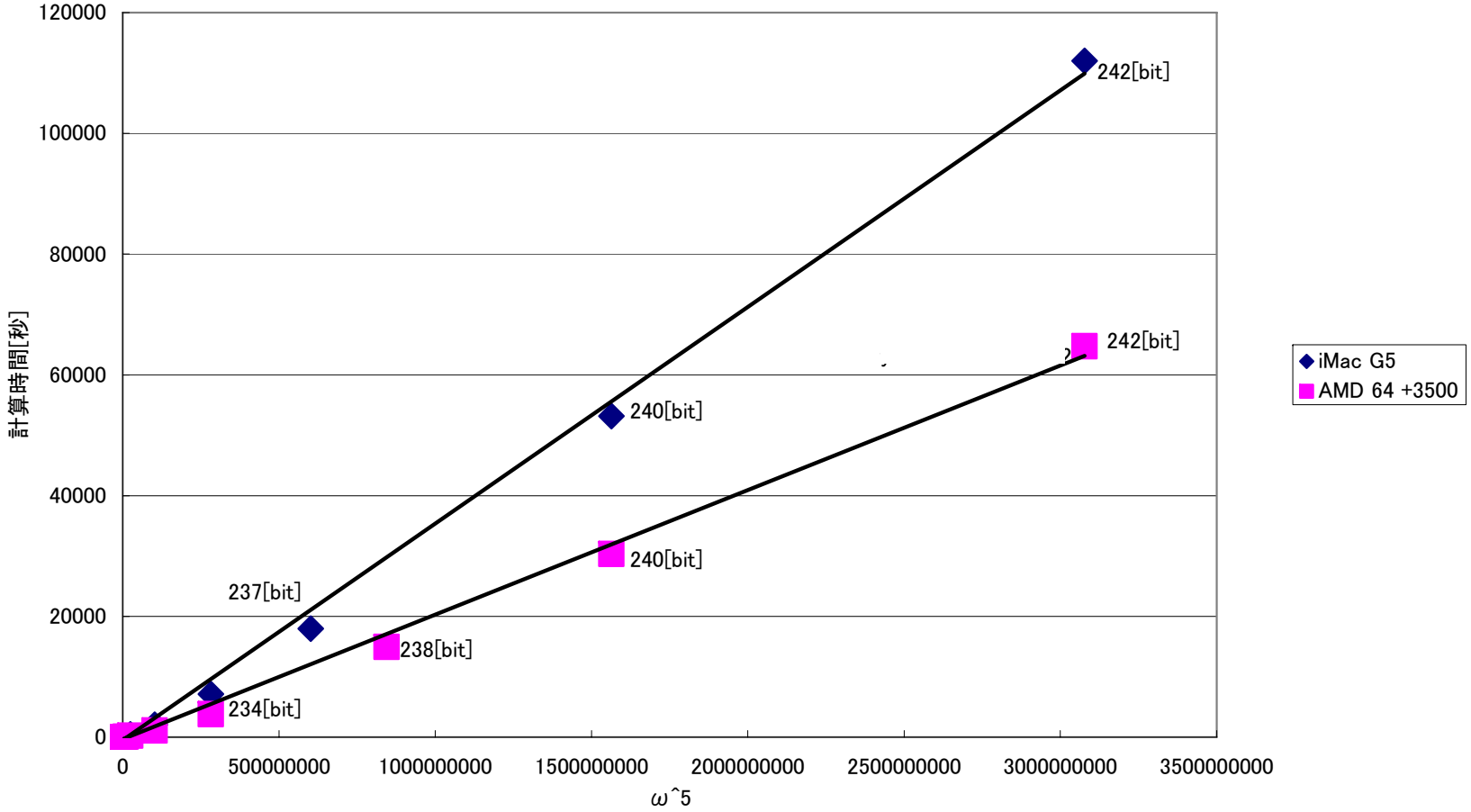
Lattice 行列の行数$\omega$は、モジュラスNのビット数をn，未知部分のビット数をy とすると，次式で与えられている。

$$\omega = \left\lceil \frac{4n}{n-4y} \right\rceil$$

# 6-2　計算時間

HITACHI
Inspire the Next

鍵長N:1024[bit], p,qは512[bit]について、NTL(+GMP) 上に実装したプログラムを用いて、未知ビット数と計算時間の関係を実測した。



# 7　まとめ

HITACHI
Inspire the Next

(1). 最上位側と最下位側からそれぞれ1/4ビット未満づつ既知で、両者のビット数の合計が1/4を超えた場合，中央部の未知ビットを推定可能であることを示した。

(2). 秘密指数の中央部が1/4ビット未知の場合、秘密素数を復元できることを示した。

(3). 実際に推定を行うコードをAMD64+3500のWindowsマシン上に実装し、p,qが512[bit]で，そのうちyビットが未知の場合の計算時間を評価。
計算時間はLattice行列の行数の5乗にオーダーとなることを確認。
計算時間は、

$$T = 2\times 10^{-5}\left(\frac{4096}{1024-4y}\right)^{5} \quad [\mathrm{s}]$$

で近似される結果が得られた。