素因数分解問題調査研究報告書

2001 年 12 月

素因数分解問題

調査研究報告書

2001年12月

目 次

1　まえがき

本調査研究報告書は、特に暗号プリミティブの評価という視座から有理整数の素因数分解問題 (integer factoring problem) を捉え、現在においてこの問題がいかなる困難さを有する問題であるかを、具体的な素因数分解アルゴリズムと計算量理論の両面から明らかにし、適切な鍵サイズを設計する際の基礎資料となることを目指すものである。

一般の素因数分解問題は、与えられた自然数 n を $n = p_1^{e_1} \cdots p_k^{e_k}$ (p_i: 素数, $e_i \geq 1$) と分解することであるが、暗号理論というコンテクストではもっと限定的に考えるのが普通であり、例えば $n = pq$ (p, q: 素数) から p (または q) を求める問題、もしくは $n = p^r q$ (p, q: 素数, $r > 1$) から p (または q) を求める問題などを考えている。言うまでもなく、これらの形の合成数を分解する効率的なアルゴリズムが発見されれば、代表的な暗号スキームの安全性は破綻するからである。また、新しいアルゴリズムの発見がなくとも、利用可能な計算資源によっては、実際に分解可能な合成数のサイズ (ビット長) は大きく変動する。例えば、これまで難しいとされていたサイズの合成数でも、莫大な数の人々の参加による分散処理 (massively distributed computing) によって現実的に分解可能となる場合もあるし、新たな高速 CPU または新しいアーキテクチャの高速計算機の出現によって分解可能になる場合もある (ただし、量子計算機と量子アルゴリズムは本稿では考慮していない。量子計算機が出現すれば素因数分解問題に基づく暗号系が破綻するのは明らかだからである)。さらに、素因数分解問題より上位の問題 (いままで関連性が注目されていなかったような意外な問題を含む) に対する効率的なアルゴリズムが発見されれば、からめ手を突かれて、結果として素因数分解問題が現実的に簡単な問題に変化してしまう可能性もある。

したがって、本調査研究の目的に照らせば、素因数分解アルゴリズムに限定した議論だけでは不十分であり、計算機の処理能力に関する技術動向や、素因数分解問題に関係する周辺の諸問題との関係などを議論しなければ、総合的な評価は不可能である。そこで本報告書では、そのような多角的な評価を行うよう努めたつもりである。

以下、第 2 節では、既知の素因数分解アルゴリズムの種類と実行時間を概観し、楕円曲線法 (Elliptic Curve Method, ECM)、一般数体ふるい法 (General Number Field Sieve, GNFS)、格子利用法 (Lattice Factoring Method, LFM) が最も有力なアルゴリズムであることを確認する。さらに、ECM と GNFS については、将来の計算環境で現実的に分解可能な合成数のサイズ (または発見可能な最小素因数のサイズ) の予想値についても数値計算に基づいて議論する。第 3 節では計算量理論の立場から、素因数分解問題に関連の深い他の数論的問題との帰着関係を議論し、計算量理論という枠組のなかでの素因数分解問題の位置付けを明らかにする。第 4 節はこれらの議論をまとめ、適切な鍵サイズに関する考察を行う。

2　素因数分解アルゴリズムに基づく検討

2.1　既知のアルゴリズム

2.1.1　概要

素因数分解アルゴリズムのうち有力なものは、分解しようとする合成数 n のサイズ (ビット長) に主に依存して実行時間が決まるものと、n に含まれる最小素因数サイズに主に依存して実行時間が決まるものに分類される。前者に属するのは、Lehman 法 [15]、連分数法 [19]、複数多項式 2 次ふるい法 (MPQS)[22, 25]、数体ふるい法 (NFS) [18] などである。後者に属するのは、試行割算法、ρ (rho) 法 [21]、楕円曲線法 (ECM)[16] などである。

これらのアルゴリズムの実行時間をまとめると次のようになる。数体ふるい法には、合成数の型を特定した特殊数体ふるい法 (SNFS) と、特定の型を仮定しない一般数体ふるい法 (GNFS) があるが、ここでは GNFS のみを示している。

アルゴリズム	実行時間	備 考
Lehman 法	$O(n^{1/3})$	
連分数法	$O(exp(c\sqrt{\ln n \ln\ln n}))$	
MPQS	$O(exp(c\sqrt{\ln n \ln\ln n}))$	$c = 1.020$
GNFS	$O(exp(c(\ln n)^{1/3}(\ln\ln n)^{2/3}))$	$c = 1.901$ [8]
試行割算法	$O(p \cdot (\log n)^2)$	p: 最小素因数
ρ 法	$O(p^{1/2} \cdot (\log n)^2)$	p: 最小素因数
ECM	$O(exp(c\sqrt{\ln p \ln\ln p}) \cdot (\log n)^2)$	p: 最小素因数, $c = 1.414$

ただし、Lehman 法、試行割算法、ρ 法を除く他のアルゴリズムの実行時間は厳密なものではなく、整数がスムーズになる確率に関する漸近的な性質を有限の範囲で近似した平均的なものである。すなわち、整数 a が B-smooth であるとは、a の素因数がすべて B 以下のときをいい、ここで

$$L_x[\nu, \lambda] = \exp(\lambda(\ln x)^{\nu}(\ln\ln x)^{1-\nu})$$

という記号を使うと、$L_x(\nu, \lambda)$ 以下の自然数が $L_x(\omega, \mu)$-smooth である確率は

$$L_x[\nu - \omega, -(\nu - \omega)\lambda/\mu + o(1)] \quad (x \to \infty)$$

であることが知られているが、この性質を $x \to \infty$ というわけではない範囲で使って実行時間を評価している、という意味である。

上の表から明らかなように、実行速度の観点からは、分解しようとする合成数のサイズに主に依存したアルゴリズムの代表は一般数体ふるい法 (GNFS)、合成数の最小素因数サイズに主に依存したアルゴリズムの代表は楕円曲線法 (ECM) であると言える。GNFS とは、簡単に言えば、$x^2 \equiv y^2 \pmod{n}$ なる $x, y (x \neq \pm y)$ を

発見するという 2 次ふるい法以来の基本方針を発展させて、数体の整数環上での分解と $\mathbf{Z}_n$ への同型を使って効率よく x, y を発見して分解するものである。楕円曲線法とは、本来は体上で定義される楕円曲線を環 $\mathbf{Z}_n$ の上で定義し、その位数が n の最小素因数の倍数になるような曲線を発見することで n を分解するものである。GNFS と ECM の具体的なアルゴリズムについては暗号理論や数論アルゴリズムなどの教科書に平易に解説されている [7, 13]。

このほか、合成数の型を特定してはいるが、暗号理論の観点から注目すべきアルゴリズムとして格子利用法 (Lattice Factoring Method, LFM) [4] がある。これは対象を $n = p^r q$ $(r \geq 1)$ の型に特化した素因数分解アルゴリズムであって、文献 [9] と [10] の結果を巧みに応用して構成されたものである。具体的には、 $n = p^r q$ (ただしある定数 c に対して $q < p^c$) として、 n, r, c が与えられたとき、LFM が素因数 p を発見するための実行時間は

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

と見積もられている。ここに γ は、成分の数が $O(r \log n)$ で $O(r^2)$ 次元の格子についての LLL アルゴリズム [17] の計算量である。

上の計算量の式をもう少し見通しよくするため、違う表現をする。いま、p, q を k ビットの素数とし、$r = k^\epsilon$ と表したとき、実行時間は

$$T(k) = 2^{(k^{1-\epsilon}) + O(\log k)}$$

と書くことができる。このとき、もし $\epsilon = 1$ (つまり r が $\log p$ 程度) ならば、この式から明らかなように、実行時間は多項式時間となる。また、$\epsilon = 1/2$ (つまり $r = \sqrt{\log p}$) のときは楕円曲線法よりもやや高速である。しかし、$n = p^2 q$ のように r が小さい場合には、ϵ は非常に小さくなり、したがって無視できるほどになり、実行時間は指数時間となってしまう。

2.1.2 考察

以上のように既知の素因数分解アルゴリズムを見る限り、暗号系を設計するという観点、すなわち n の型やサイズに関して安全性を確保するという観点からは、次のような自明な結論しか出てこない。

1. n のサイズは GNFS が現実的に実行不可能な範囲にとるべきである。
2. n の最小素因数 p のサイズは ECM が現実的に実行不可能な範囲にとるべきである。
3. $n = p^r q$ $(r > 1)$ の場合、LFM が現実的に実行不可能となるよう、r は小さく (例えば $r = 2$ と) とるべきであり、r を $\sqrt{\log p}$ 程度にしなければならない場合は、相応の p のサイズにすべきである。

$n = pq$ でも $n = p^r q$ でも、最初の 2 項目は必須である。ただし、実際には ECM の実行時間は漸近的に速くはないので、GNFS に対して安全なように $n = pq$ を選べば、ECM に対しても耐性をもつのが普通である。注意すべきは、$n = pq$ について、 k ビットもあれば GNFS に対して安全と見積もったからといって、$n = p^r q$ も同じく k ビットで安全である保証はない点である。言うまでもなく、最小素因数の小ささが効いて、ECM で分解される可能性があるからであるが、このような現象が発生する n と r の範囲については理論的な考察は難しいのが現状である。つまり、n のサイズ $|n|$ に対して最小素因数のサイズを $|n|/(r+1)$ $(r \geq 1)$ としたとき、

$$\begin{aligned}
&\exp\left(1.414\left(\frac{|n|\log_e 2}{r+1}\right)^{1/2}\left(\log_e\left(\frac{|n|\log_e 2}{r+1}\right)\right)^{1/2}\right) \\
&\leq \exp\left(1.901\left(|n|\log_e 2\right)^{1/3}\left(\log_e\left(|n|\log_e 2\right)\right)^{2/3}\right)
\end{aligned}$$

が成立する $|n|$ の上限を明らかにする必要があるが、これは理論的には難しい。おそらく、数値実験によって小さな $|n|$ について調査し、それを外挿して推定するしか方法はないと考えられる。特定の n を競って分解するような場当たり的な数値実験ではなく、系統的な実験がぜひとも必要である。

ところで、「現実的に実行不可能な範囲」とはどのような範囲を指すのか、非常に曖昧ではある。また、LFM はアルゴリズム自体が開発されて間もないことから、これから改良型研究が盛んに行われれば、$n = p^r q$ 型には安全性のためにさらに多くの制約条件が追加される可能性が潜在的にある。そこで次節では、GNFS と ECM に関して、現実的に「実行可能」な範囲の予想値を示すことにする。

2.2 分解可能サイズの数値予想

Brent [5] は、1960 年代から現在までに素因数分解された合成数のサイズと年次推移を詳しく調査した結果、それらのサイズの年次増加曲線は、新アルゴリズムの開発と Moore の法則 [23] (半導体素子の集積度が 18 か月ごとに 2 倍になるという経験則) の両方に依存していることを確認した。そこで、このような傾向が今後も継続するであろうという仮定[1]のもとでこの曲線を外挿し、将来、その時点で現実的に素因数分解可能な合成数 (またはその合成数の最小素因数) のサイズを与える予想式 (実験式) を導出している。本節では、その式に基づく考察を述べる。

2.2.1 楕円曲線法

実行時間が素因数サイズに依存するアルゴリズムの代表である楕円曲線法 (ECM) については、Y を年 (西暦)、D を ECM で発見される素因数のサイズ (10 進桁数: digits) として、

$$\sqrt{D} = (Y - 1932.3)/9.3$$

という実験式が与えられている。D は 10 進桁数なので、これをビットに換算して計算すると、例えば今後 50 年では次のような表になる。

年	bits	年	bits	年	bits	年	bits	年	bits
2001	182	2011	238	2021	303	2031	375	2041	454
2002	187	2012	244	2022	310	2032	382	2042	463
2003	192	2013	251	2023	316	2033	390	2043	471
2004	198	2014	257	2024	323	2034	398	2044	480
2005	203	2015	263	2025	331	2035	406	2045	488
2006	209	2016	270	2026	338	2036	414	2046	497
2007	215	2017	276	2027	345	2037	422	2047	506
2008	221	2018	283	2028	352	2038	430	2048	515
2009	226	2019	289	2029	360	2039	438	2049	524
2010	232	2020	296	2030	367	2040	446	2050	533

今後 100 年については、図 1 にグラフを示した。

$n = pq$ と $n = p^2q$ がともに $|n| = 1024$ で $|p| = |q|$ の場合、この予想によれば、前者の最小素因数 (512 ビット) が ECM で現実的に計算できるようになるのは 2048 年頃、後者の最小素因数 (342 ビット) が ECM で現実的に計算できるようになるのは 2027 年頃ということになる。

[1] シリコン半導体に関する限り Moore の法則は限界を迎えているという説が有力であった。これは、シリコン半導体素子の絶縁膜 (SiO_2) の厚みが平均的に原子 12 個分に達しており、およそ 9 個から 10 個分が限界であろうと考えられていたからである。しかし、これを 6 個分まで薄くできることが 2000 年になって判明し [2]、この結果、Moore の法則の「寿命」は少なくとも 2005 年以降まで延びたと見られている。

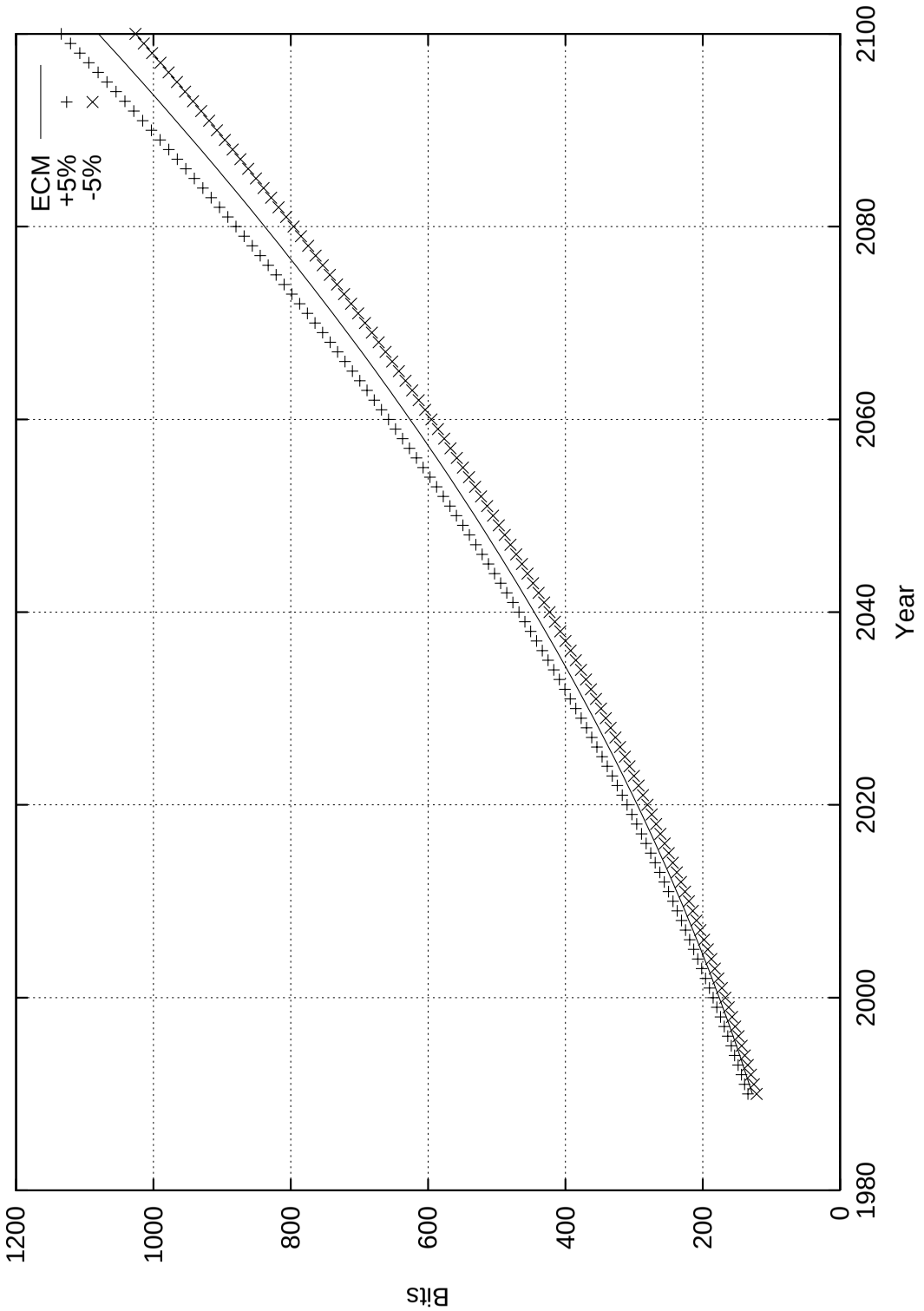

図 1 ECM で発見される素因数サイズの年次推移予想
横軸: 年 (西暦), 縦軸: ビット
($\sqrt{D} = (Y - 1932.3)/9.3$ からビット換算. 誤差 $\pm 5\%$ を参考表示.)

2.2.2 一般数体ふるい法

実行時間が合成数サイズにのみ依存するアルゴリズムの代表である一般数体ふるい法 (GNFS) については、Y を年 (西暦)、D を NFS で分解される合成数のサイズ (10 進桁数: digits) として、

$$D^{1/3} = (Y - 1928.6)/13.24$$

という実験式が与えられている。D は 10 進桁数なので、これをビットに換算して計算すると、例えば今後 50 年では次のような表になる。

年	bits	年	bits	年	bits	年	bits	年	bits
2001	544	2011	801	2021	1130	2031	1537	2041	2033
2002	566	2012	831	2022	1167	2032	1583	2042	2088
2003	590	2013	861	2023	1205	2033	1629	2043	2143
2004	614	2014	892	2024	1243	2034	1676	2044	2200
2005	639	2015	924	2025	1283	2035	1725	2045	2258
2006	664	2016	956	2026	1323	2036	1774	2046	2316
2007	690	2017	989	2027	1364	2037	1824	2047	2376
2008	717	2018	1023	2028	1406	2038	1875	2048	2437
2009	744	2019	1058	2029	1449	2039	1926	2049	2499
2010	772	2020	1093	2030	1493	2040	1979	2050	2561

今後 100 年については、図 2 にグラフを示した。

合成数 n について、$|n| = 1024$ の場合、この予想によれば n が GNFS で現実的に計算できるようになるのは 2018 年頃ということになる。

なお、現在のところ一般数体ふるい法により分解された合成数の最大サイズは 512 ビット (10 進 155 桁: いわゆる RSA-155) である [6]。すなわち、

$$\begin{aligned} n = \quad & 109417386415705274218097073220403576120037329454492059909138421314763499842889347847179972578912673324976257528997818337970765372440271467435315933543338897 \end{aligned}$$

に対して一般数体ふるい法を適用し、

$$\begin{aligned} p = \quad & 102639592829741105772054196573991675900716567808038066803341933521790711307779 \end{aligned}$$

$$\begin{aligned} q = \quad & 106603488380168454820927220360012878679207958575989291522270608237193062808643 \end{aligned}$$

と分解した。計算コストは約 8400 MIPS·years であった。

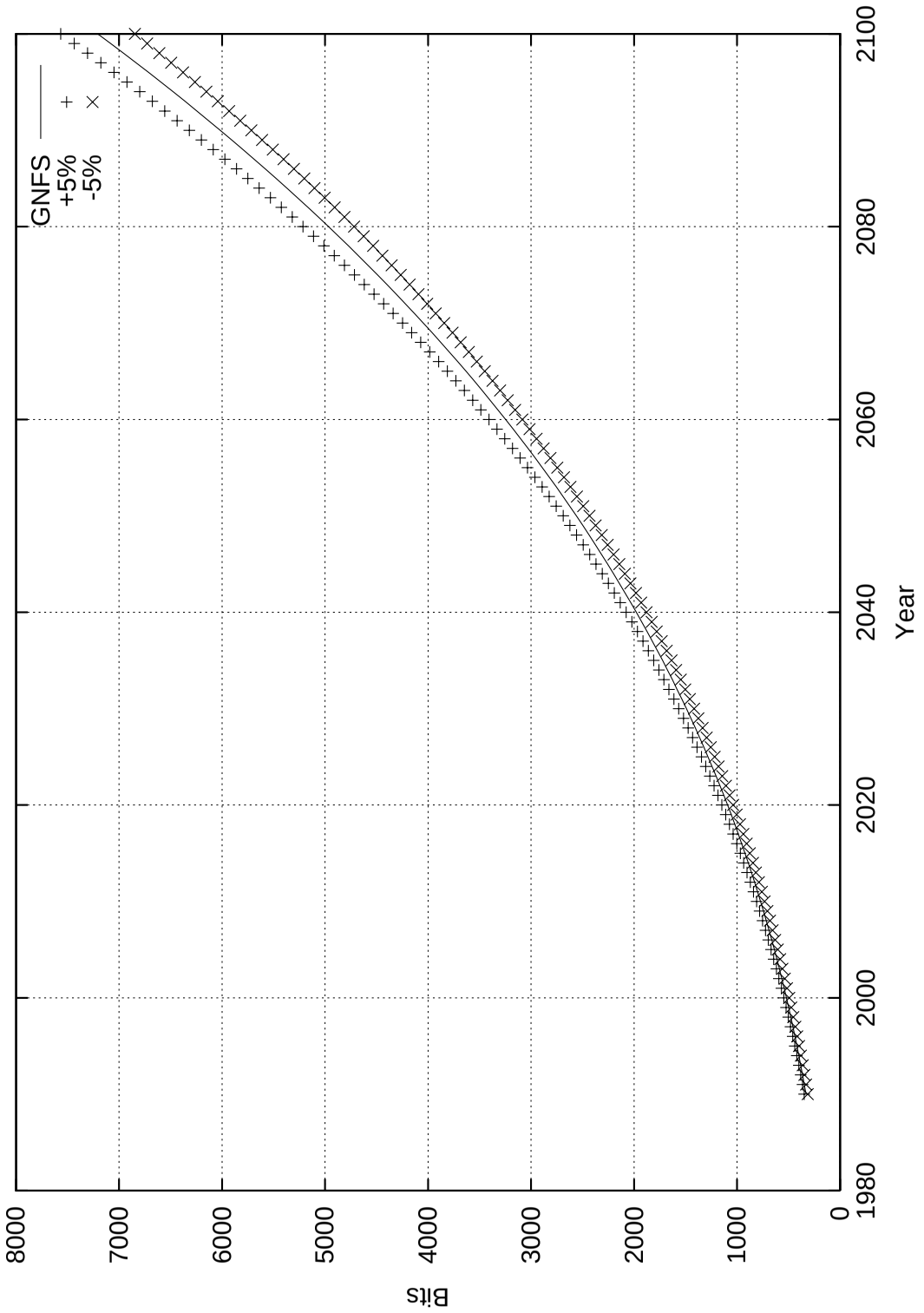

図 2 GNFS で分解される合成数サイズの年次推移予想
横軸: 年 (西暦), 縦軸: ビット
($D^{1/3} = (Y - 1928.6)/13.24$ からビット換算. 誤差 $\pm 5\%$ を参考表示.)

2.2.3 考察

本小節で示した具体的な数値は (RSA-155 の分解を除いて)、あくまで Brent [5] の実験式に基づく数値計算で得られた予想値である。これらの実験式が示す曲線が過去のデータに対する正確なフィッティングを与えるものであるとしても、将来を正確に外挿している保証はない。実際、本小節冒頭の脚注で述べたように Moore の法則は、寿命は延びたとはいえ、限界に近付いている。さらに、将来発見される可能性のある新しい素因数分解アルゴリズムは、素因数分解問題を多項式時間で解くかもしれないし (もちろん古典的 Turing 機械の計算モデルの上で)、そのような劇的な事態とならないまでも、楕円曲線法 (ECM) や一般数体ふるい法 (GNFS) の高速化手法の発見は、 その時点で図 1 や図 2 の曲線をステップ状に著しく上昇させる可能性がある。

しかしながら、上記のような多くの不確定要素を含むとはいえ、安全な暗号系を設計するために n のサイズを選定するという観点からは、本小節で示した予想値のほかは系統的なデータは見られないという事実もまた正視すべきである。そこで、 n のサイズ $|n|$ を選定したら、とりあえずこの予想値に依拠してその n が分解可能となる年 (西暦) を逆引きで読み取り、現在年との差 y_d を使って、「今後新たな高速アルゴリズムが発見されない限りにおいて y_d 年程度は安全と見積もられる暗号系」と考えるのも、一つの方策であろう。

3　計算量理論に基づく検討

本節では帰着の概念を使って問題の難しさの上下関係を扱い、そのような切り口で素因数分解問題を検討する。問題 A が問題 B に帰着するとは、問題 B が解ければ、そのアルゴリズムを使って問題 A が簡単に解けてしまうような場合をいう。難しさが同等とは、互いに帰着する関係をいう。

合成数 $n = pq$ や $n = p^r q$ を分解する問題は、一般の (合成数の形を限定しない) 素因数分解問題に帰着する。しかし n の形には特徴があり、細かに見れば、それぞれの分解問題は、一般の素因数分解問題との間に入る別々の中間的な問題に帰着しており、微妙な差異が見られる。その様子を明らかにするのが本節の目的である。

3.1　準備

最初に帰着の概念を定義するが、この定義は文献 [26] に概ね準拠している。問題の間の帰着関係を考えるにあたり、「問題」は「関数」(部分関数) として抽象化している。

- 関数 f が関数 g に多項式時間 Turing 帰着するとは、g の値に高々多項式回アクセスできる多項式時間 Turing 機械が f の値を計算できることをいう。これを $f \leq_T^p g$ とかく。
- 関数 f が関数 g に多項式時間 truth-table 帰着するとは、高々多項式個の g の値に一括して一度だけアクセスできる多項式時間 Turing 機械が f の値を計算できることをいう。これを $f \leq_{tt}^p g$ とかく。一括してアクセスする際の値の数が高々定数 k 個であるときは、特に $f \leq_{k-tt}^p g$ とかく。
- 関数 f が関数 g に多項式時間 many-one 帰着するとは、多項式時間計算可能な関数 h が存在し、g の値に一度だけアクセスできる多項式時間 Turing 機械が f の値を $f(x) = g(h(x))$ と計算できることをいう。これを $f \leq_m^p g$ とかく。

以上で、$f \leq_\alpha^p g$ かつ $g \leq_\alpha^p f$ のとき、$f \equiv_\alpha^p g$ とかく ($\alpha \in \{T, tt, m\}$)。また定義から明らかなように、$f \leq_m^p g \Rightarrow f \leq_{tt}^p g \Rightarrow f \leq_T^p g$ である。

次に、本節で扱う関数 (問題) を定義する。

- `IntegerFactoring` (一般の素因数分解問題):
  入力 $n(> 1)$ に対して、$n = p_1^{e_1} \cdots p_k^{e_k}$ となる $\{(p_1, e_1), \ldots, (p_k, e_k)\}$ (p_i: 素数, $e_i \geq 1$) を出力する。

- `EulerFunction` (Euler 関数の計算問題):
  入力 $n(>1)$ に対して Euler 関数の値 $\phi(n)$ を出力する。

- `SquarefreePart` (平方無縁部分の抽出問題):
  入力 $n(>1)$ に対して、$n = r^2 s$ かつ s が平方無縁 (squarefree) となる $\{r, s\}$ を出力する。
  (註: s が平方無縁とは、s が a^2 型の因数 $(a > 1)$ をもたないことである。)

- `Factorpq`($n = pq$ 型合成数の分解問題):
  入力 $n(>1)$ に対して、$n = pq$ となる素数の組 $\{p, q\}$ を出力する。

- `Factorprq` (k は定数 > 1, $n = p^r q$ 型合成数の分解問題):
  入力 $n(>1)$ に対して、$n = p^r q$ となる素数の組 $\{p, q\}$ を出力する。

- `DLmodn`(自然数 > 1 を法とする離散対数問題): 入力 (n, g, y) に対して、$y \equiv g^x \pmod{n}$ となる x を出力する。

- `DHmodn`(自然数 > 1 を法とする DH 鍵共有を破る問題):
  入力 (n, g, A, B) に対して、$A \equiv g^a \pmod{n}$, $B \equiv g^b \pmod{n}$, $C \equiv g^{ab} \pmod{n}$ となる C を出力する。

- `RSA` (RSA を破る問題):
  入力 (n, e, c) に対して、$c \equiv m^e \pmod{n}$ となる m を出力する。

3.2 帰着関係

これらの関数の帰着関係は図 3 のとおりである。

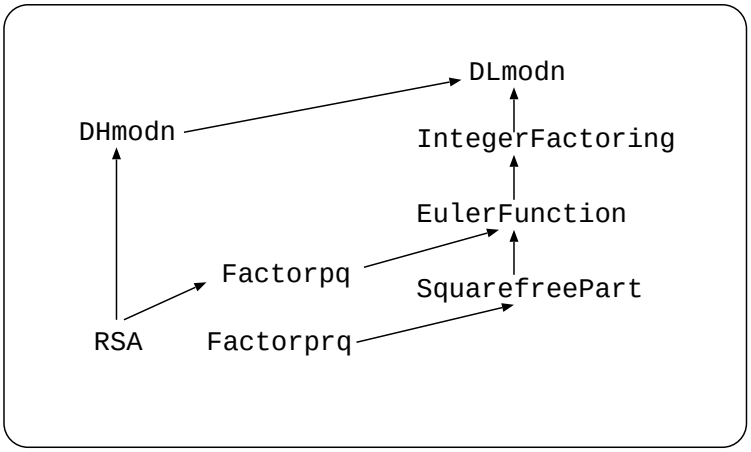

図 3 帰着関係の概要
($A \longrightarrow B$ は A が B に帰着することを表す)

以下、各帰着を解説する。

1. $\texttt{Factorpq} \leq^p_{1-tt} \texttt{EulerFunction}$
   入力 $n(=pq)$ に対して、$\texttt{EulerFunction}(n)$ は $s = \phi(n) = (p-1)(q-1) = n-p-q+1$ を返す。これにより、p を未知数として $p^2-(n-s+1)p+n=0$ なる 2 次方程式を p について解けばよい。
2. $\texttt{Factorprq} \leq^p_{1-tt} \texttt{SquarefreePart}$
   入力 $n = p^r q$ に対して、$\texttt{SquarefreePart}(n)$ は $r(>1)$ が偶数ならば $\{p^{r/2}, q\}$ を、r が奇数ならば $\{p^{(r-1)/2}, pq\}$ を返す。偶数のときは $n/q = p^r$ から p を計算する (一般に、正整数 a が与えられて、ある k に対して $a = b^k$ なる b が存在するとき、その最小の b とそのときの k を求めることは 2 分探索による決定性多項式時間で可能である)。奇数のときは、まず $p^{(r-1)/2}$ から p を求め、それから q を求める。
3. $\texttt{EulerFunction} \leq^p_{1-tt} \texttt{IntegerFactoring}$
   自明である。
4. $\texttt{SquarefreePart} \leq^p_T \texttt{EulerFunction}$
   これは自明ではない。文献 [14] による。
5. $\texttt{IntegerFactoring} \leq^p_T \texttt{DLmodn}$
   文献 [27] による。
6. $\texttt{DHmodn} \leq^p_{1-tt} \texttt{DLmodn}$
   自明である。
7. $\texttt{RSA} \leq^p_m \texttt{DHmodn}$
   文献 [20] による。
8. $\texttt{RSA} \leq^p_{1-tt} \texttt{Factorpq}$
   自明である。なお、秘密鍵 d が小さいという特別な場合 ($d < n^{0.292}$ の場合) は多項式時間で d が求まるので、結局 n の分解も (Factorpq のオラクルの助けを借りずに) 計算できる [3]。

もちろん、途中の関数を経由しない直接的な帰着も存在する。例えば、次の帰着である ($\leq^{ep}_T$ とは、平均的多項式時間の意味で $\leq^p_T$ 帰着を表す)。

$\texttt{Factorpq} \leq^{ep}_T \texttt{DLmodn}$

入力 $n(=pq)$ に対して、$g \in \mathbf{Z}^*_n$ をランダムに選ぶ。もし、g の位数が $p+q-1$ より大きければ、$s = \texttt{DLmodn}(g^n) = p+q-1$ である。なぜならば、

$$g^n = g^{n-\phi(n)} = g^{n-(n-p-q+1)} = g^{p+q-1}$$

だからである。このとき、p, q を未知数として $n = pq$ と $s = p+q-1$ から p または q について解けばよい。

問題は、都合のよい g が選ばれる確率であるが、それについては次の事実が知られている [11]。

$$\mathrm{Prob}\left[ord_n(g) < \frac{1}{|n|^k} \cdot (p-1)(q-1)\right] \leq O\left(\frac{1}{|n|^{(k-4)/3}}\right)$$

すなわち、ランダムに選んだ g の位数が異常に小さいことはほとんどない、という意味である。これによりこの帰着が成立する。

このほか、素因数分解問題を含む代表的な数論的問題の帰着関係については文献 [1] に網羅的に記載されている。

3.3 考察

3.3.1 帰着関係について

Factorpq($n = pq$ の分解問題) と Factorprq($n = p^r q$ の分解問題) については、平方無縁部分抽出問題 (SquarefreePart) に帰着するかどうかが焦点である。現時点では Factorpq が SquarefreePart に帰着するとは知られていないし、帰着しそうには見えない。しかし、Factorprq はほぼ自明に帰着する。すなわち、SquarefreePart に対する効率的アルゴリズムが発見された場合、$n = p^r q$ は効率的に分解されるが、$n = pq$ は難しい問題として生き残る可能性がある。これは注意すべき事項である。

もっとも、SquarefreePart が効率的に解けるようなアルゴリズムが発見されれば、そこには、一般の素因数分解問題 (IntegerFactoring) 自体に対するアルゴリズムを大きく進展させるような根本的なアイデアが必ず含まれると予想されるので、SquarefreePart だけが独立に簡単になることは現実的には考えにくい。したがって、SquarefreePart を介した $n = pq$ と $n = p^r q$ の難しさの差異は、「理論的には」違いがあるように見える程度と考えるのが自然であろう。

一方、DLmodn(自然数を法とする離散対数問題) は、ここで考察してきた問題のなかでは最上位に位置する問題である。これに対する効率的なアルゴリズムが発見されれば、離散対数問題に基づく暗号系はもとより、素因数分解問題に基づく暗号系もすべて破綻することになる。帰着性の定義から明らかなように、DLmodn は素因数分解問題と同等かそれ以上の難しさをもつ問題であるから、素因数分解問題ですら難しい現状では、そう簡単に効率的なアルゴリズムなど発見されそうにない。しかし、理論上は起こり得ることであるから、これも注意すべき事項ではある。

なお、素数を法とする離散対数問題と素因数分解問題の直接的な帰着関係は解明されていない。それぞれに対する従来のアルゴリズムが似たような実行時間をもつことから、両者の難しさは同等であることを示唆してはいるが、計算量理論の観点からは、両者が帰着関係の意味で同等かどうかは未解明であるどころか、片

方からもう一方への帰着も知られていない。したがって、素数を法とする離散対数問題に対する効率的なアルゴリズムの発見が、素因数分解問題に基づく暗号系にとって直ちに脅威になるとは言えない。しかし、合成数を法とする離散対数問題を解くアルゴリズムに影響を与え、それを通して素因数分解問題に何らの影響が及ぶことは十分に考えられる。

3.3.2 計算量のクラスについて

もし、P=NP のような劇的な事態が発生した場合は、素因数分解問題はもはや難しい問題ではなくなる。それどころか、公開鍵系のほとんどの暗号系は破綻する。

P=NP よりは弱いステートメントであるが、例えばもし、P= NP ∩ co-NP であったならば、やはり素因数分解問題は簡単になってしまう。これは、集合

$$L = \{(n,k)|\exists a[(1 < a < n) \wedge (a \leq k)]\}$$

が NP ∩ co-NP に属することと、L の判定問題が素因数分解問題と等価であることによるものである。しかし、NP ∩ co-NP というクラスには完全集合の存在が知られていないことから、NP ∩ co-NP がまるごと P まで潰れるような事態は考えにくい。

また、拡張リーマン予想 (Extended Riemann Hypothesis, ERH) が肯定的に解決された場合、素数判定が決定性多項式時間になったり、$\bmod p$ の平方非剰余が決定性多項式時間で発見できるようになるだけで、素因数分解問題に大きな影響は与えないであろうと予想される。ただし、計算量理論の面では `IntegerFactoring` という関数の位置付けが少し変化する。現在のところ、この関数は NPSV という関数のクラスに属している。これは、NP マシンで計算できる一価部分関数のクラスである。ERH が肯定的に解決されると、`IntegerFactoring` は NPSV_g というクラスに入ることになる ($\mathsf{NPSV}_g \subseteq \mathsf{NPSV}$)。一般に、関数のクラス $\mathcal{C}$ に対して、関数 $f \in \mathcal{C}$ が $\mathcal{C}_g$ に入るとは、$graph(f) = \{(x,y)|f(x) \mapsto y\} \in \mathsf{P}$ という意味である。つまり、現在のところ素数の集合が P に属してはいないので、n とその素因数分解を見せられても、それが正しい素因数分解になっているかどうかを決定性多項式時間では判定できない。それで `IntegerFactoring` $\in$ NPSV であっても、$\in \mathsf{NPSV}_g$ ではないのが現状である。

なお、多項式時間計算可能な関数のクラスは PF と呼ばれ、PF $\subseteq$ NPSV ではあるが、NPSV $\subseteq$ PF は P=NP と同値であることが知られている [24]。したがって、関数のクラスの包含関係に関する劇的な事態から素因数分解問題が簡単になることも考えにくい。

4　むすび

これまでの考察に基づき、$n = pq$ と $n = p^2q$ のサイズについては、安全性確保の観点からは次のように結論される。

(1) $n = pq$ については、$|p| = |q|$ であって $|n| = 1024$ であれば、より高速な新しいアルゴリズムが発見されない限り、今後 20 年程度は安全と考えられる。
(2) $n = p^2q$ については、$|p| = |q|$ であって $|n| = 1024$ であれば、より高速な新しいアルゴリズムが発見されない限り、今後 20 年程度は安全と考えられる。

また、数論的な問題の間の帰着関係という観点から注意すべき事項は次のとおりである。

(3) 平方無縁部分抽出問題に対する効率的なアルゴリズムの発見は $n = p^rq$ の分解を容易にするが、$n = pq$ は難しいままで生き残る可能性が理論上はある。
(4) 合成数を法とする離散対数問題に対する効率的なアルゴリズムの発見は、$n = pq$ や $n = p^rq$ に限らず、一般の素因数分解問題を容易にする。

このほか、計算環境という観点からは次の事実にも注意する必要がある。

(5) Moore の法則の寿命は 2005 年以降に延びたが、限界に近付いていることに変わりはないので、将来の計算機の処理能力を過去の実績から外挿するのは過大評価になる可能性がある。
(6) しかし一方で、素因数分解問題に対する関心が高まり、多数の人々の協力による分散処理が世界的規模で実施された場合、素因数分解問題にとって脅威となる計算機は、単体の高速計算機ではなく、そのような分散処理機械の総体そのものであり、同時にこれは、理論上は地球上のすべての計算機にまで拡大する可能性がある。

特に (6) については、典型例として地球外生命の探査プロジェクト (SETI) がある。現在、このプロジェクトの参加者は約 300 万人であり、これらの人々によって提供されている計算機 (ほとんどは PC) の全体的な平均処理能力は 10 TFLOPS (=10000 GFLOPS) に達している [12]。これは、世界のスーパーコンピュータのトップ 500 に入る日本のスーパーコンピュータ全部の処理能力の総和に匹敵するものである。

参考文献

[1] L. M. Adleman, "Algorithmic number thoery – The complexity contribution," Proc. 35th FOCS, pp.88–113, 1994.

[2] M. A. Alam, J. Bude, A. Ghetti, "Field acceleration for oxide breakdown – Can an accurate anode hole injection model resolve the E vs. 1/E controversy?" Proc. IEEE 2000 International Reliability Physics Symposium, pp.16–20, 2000.

[3] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," Notices of the AMS, vol.46, pp.203–213, 1999.

[4] D. Boneh, G. Durfee, N. Howgrave-Graham, "Factoring $N = p^r q$ for large r," Proc. Crypto'99, LNCS 1666, Springer-Verlag, pp.326–337, 1999.

[5] R. P. Brent, "Recent progress and prospects for integer factorisation algorithms," Proc. COCOON 2000, LNCS 1858, Springer-Verlag, pp.3–22, 2000.

[6] S. Cavallar, W. Lioen, H. te Riele, B. Dodson, A. K. Lenstra, P. L. Montgomery, B. Murphy, K. Aardal, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. C. Putnam, P. Zimmerman, "Factorization of a 512-bit RSA modulus," Report MAS-R0007, CWI, Feb. 29, 2000.

[7] H. Cohen, A Course in Computational Algebraic Number Theory, GTM 138, Springer-Verlag, 1993.

[8] D. Coppersmith, "Modifications to the number field sieve," J. Cryptology, vol.6, pp.169–180, 1993.

[9] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," J. of Cryptology, vol.10, pp.233–260, 1997.

[10] N. Howgrave-Graham, "Finding small roots of univariate modular equations revisited," Proc. Cryptography and Coding, LNCS 1355, Springer-Verlag, pp.131–142, 1997.

[11] J. Håstad, A. W. Schrift, A. Shamir, "The discrete logarithm modulo a composite hides $O(n)$ bits," JCSS, vol.47, pp.376–404, 1993.

[12] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M. Lebofsky, "SETI@ home: Massively distributed computing for SETI," IEEE Computer Society, 2000.
(`http://www.computer.org/cise/articles/seti.htm`)

[13] 小山謙二, "暗号と素因数分解," 情報理論とその応用学会 (編)『暗号と認証』, 培風館, pp.29–45, 1996.

[14] S. Landau,"Some remarks on computing the square parts of integers," Information and Computation, vol.78, pp.246–253, 1988.

[15] R. S. Lehman, "Factoring large integers," Math. Comp., vol.28, pp.637–646, 1974.

[16] H. W. Lenstra, Jr., "Factoring integers with elliptic curves," Annals of Mathematics, vol.126, pp.649–673, 1987.

[17] A. K. Lenstra, H. W. Lenstra, Jr., L. Lovasz, "Factoring polynomial with rational coefficients," Mathematiche Annalen, vol.261, pp.515–534, 1982.

[18] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, "The number field sieve," Proc. 22nd STOC, pp.564–572, 1990.

[19] M. A. Morrison, J. Brillhart, "A method of factorisation and the factorisation of F_7," Math. Comp., vol.29, pp.183–205, 1975.

[20] M. Mambo, H. Shizuya, "A note on the complexity of breaking Okamoto-Tanaka ID-based key exchange scheme," IEICE Trans. Fundamentals, vol.E82-A, pp.77–80, 1999.

[21] J. M. Pollard, "A Monte Carlo method for factorisation," BIT, vol.15, pp.331–334, 1975.

[22] C. Pomerance, "The quadratic sieve factoring algorithm," Proc. Eurocrypt'84, LNCS 209, Springer-Verlag, pp.169–182, 1984.

[23] R. S. Schaller, "Moore's law: past, present and future," IEEE Spectrum, vol.34, no.6, pp.52–59, 1997.

[24] A. L. Selman, "A taxonomy of complexity classes of functions," JCSS, vol.48, pp.357–381, 1994.

[25] R. D. Silverman, "The multiple polynomial quadratic sieve," Math. Comp., vol.48, pp.329–339, 1987.

[26] K. Sakurai, H. Shizuya, "A Structural Comparison of the Computational Difficulty of Breaking Discrete Log Cryptosystems," J. of Cryptology, vol.11, pp.29–43, 1998.

[27] H. Woll, "Reductions among number theoretic problems," Information and Computation, vol.72, pp.167–179, 1987.