

情報セキュリティ

インターネットで安全に情報交換するための
暗号技術について

2008年7月20日

東京工芸大学 工学部 コンピュータ応用学科

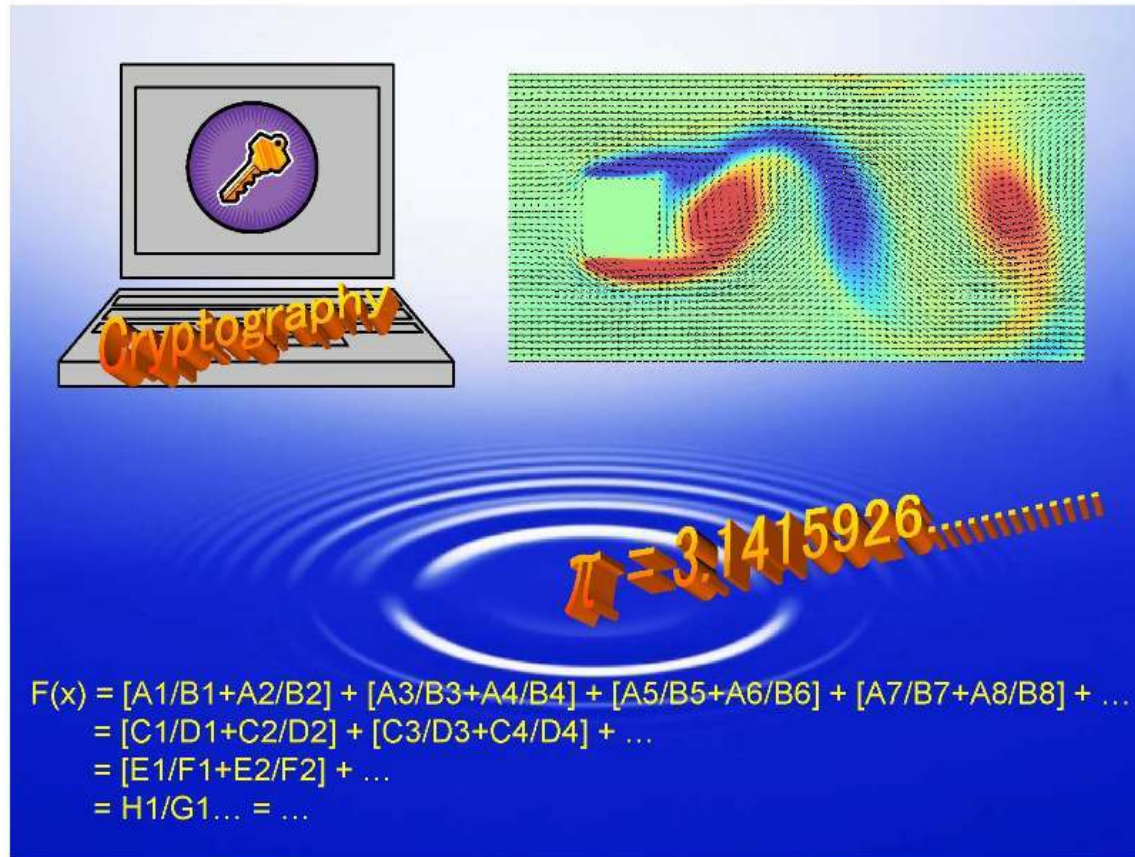
数値シミュレーション研究室

後 保範 (Ushiro Yasunori)

目次

1. 数値シミュレーション研究室
2. インターネットでの情報伝達
3. 情報を安全に送るには何が必要か
4. 使用されている暗号を調べる
5. 共通鍵暗号と公開鍵暗号
6. RSA暗号の仕組み
7. RSA暗号の強度と暗号解読法

1. 数値シミュレーション研究室

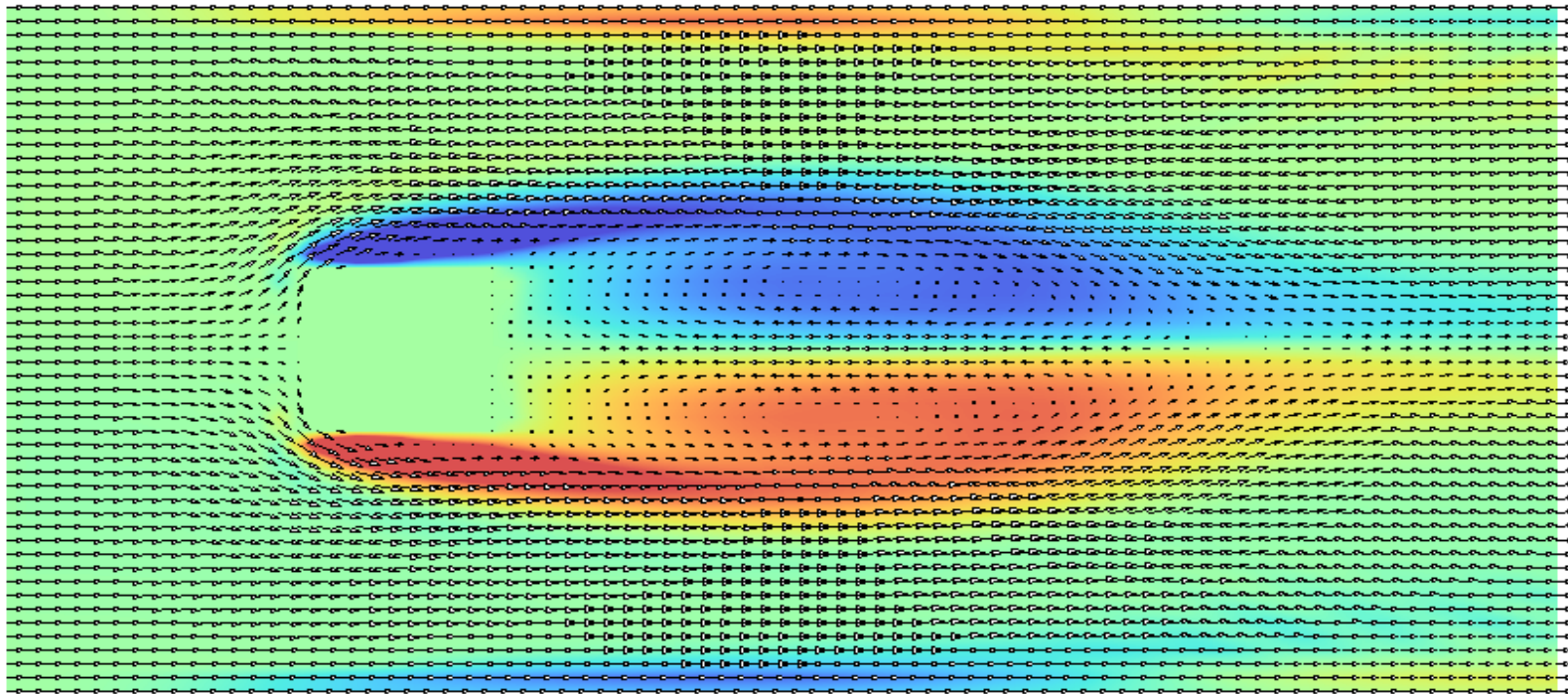


$$F(x) = [A1/B1+A2/B2] + [A3/B3+A4/B4] + [A5/B5+A6/B6] + [A7/B7+A8/B8] + \dots$$
$$= [C1/D1+C2/D2] + [C3/D3+C4/D4] + \dots$$
$$= [E1/F1+E2/F2] + \dots$$
$$= H1/G1\dots = \dots$$

1.1 研究室の研究内容

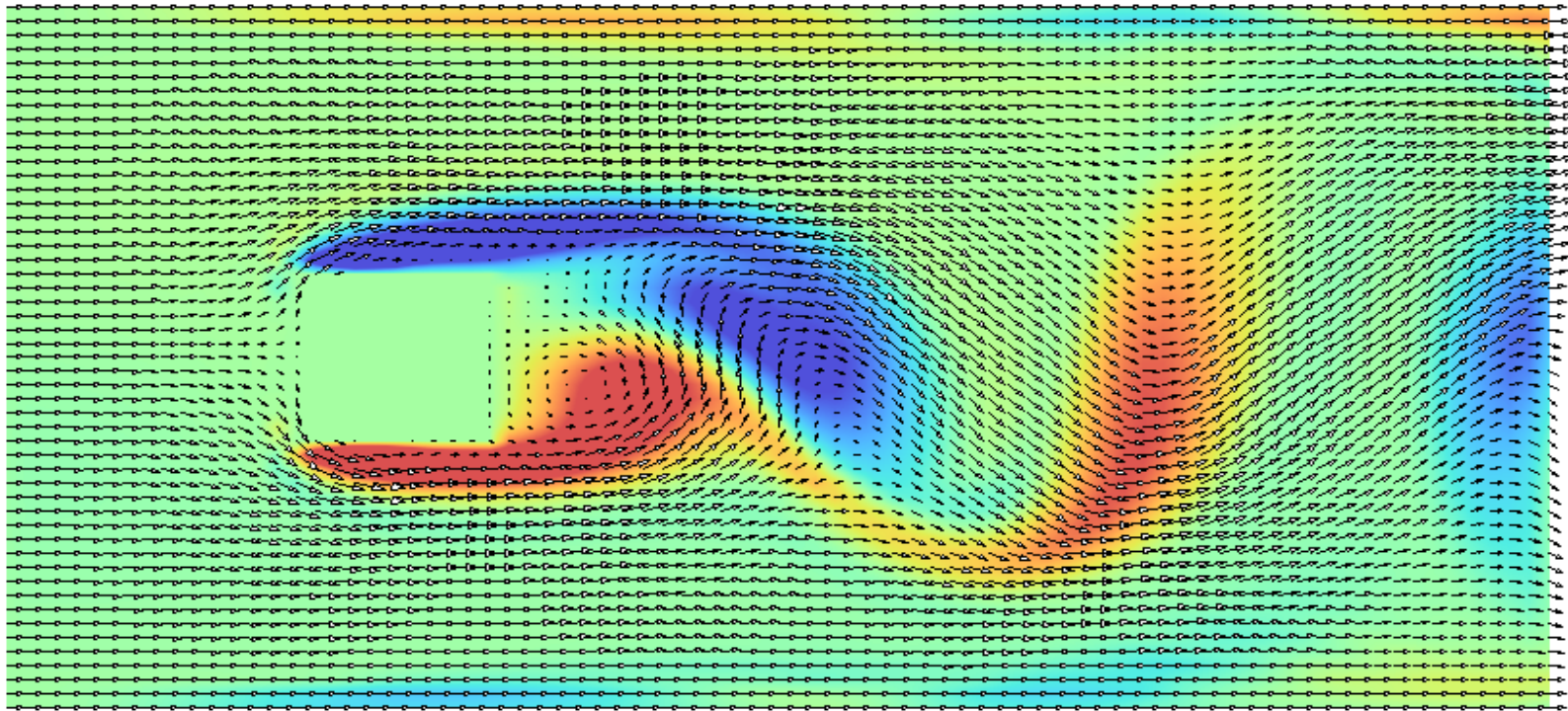
- (1) 物理現象の数値シミュレーション
流れ解析、行列計算、並列計算
- (2) 電子社会の安全を保つ暗号技術
暗号の仕組み、RSA暗号計算、
多数桁計算(2002年に π 1.2兆桁の世界記録)
- (3) 数学的手法を用いた投資分析
投資の統計処理、最小二乗近似、
テクニカル分析

1.2 流れ解析の例(1/2)



角柱の周りの流れ解析:カルマン渦の発生原因

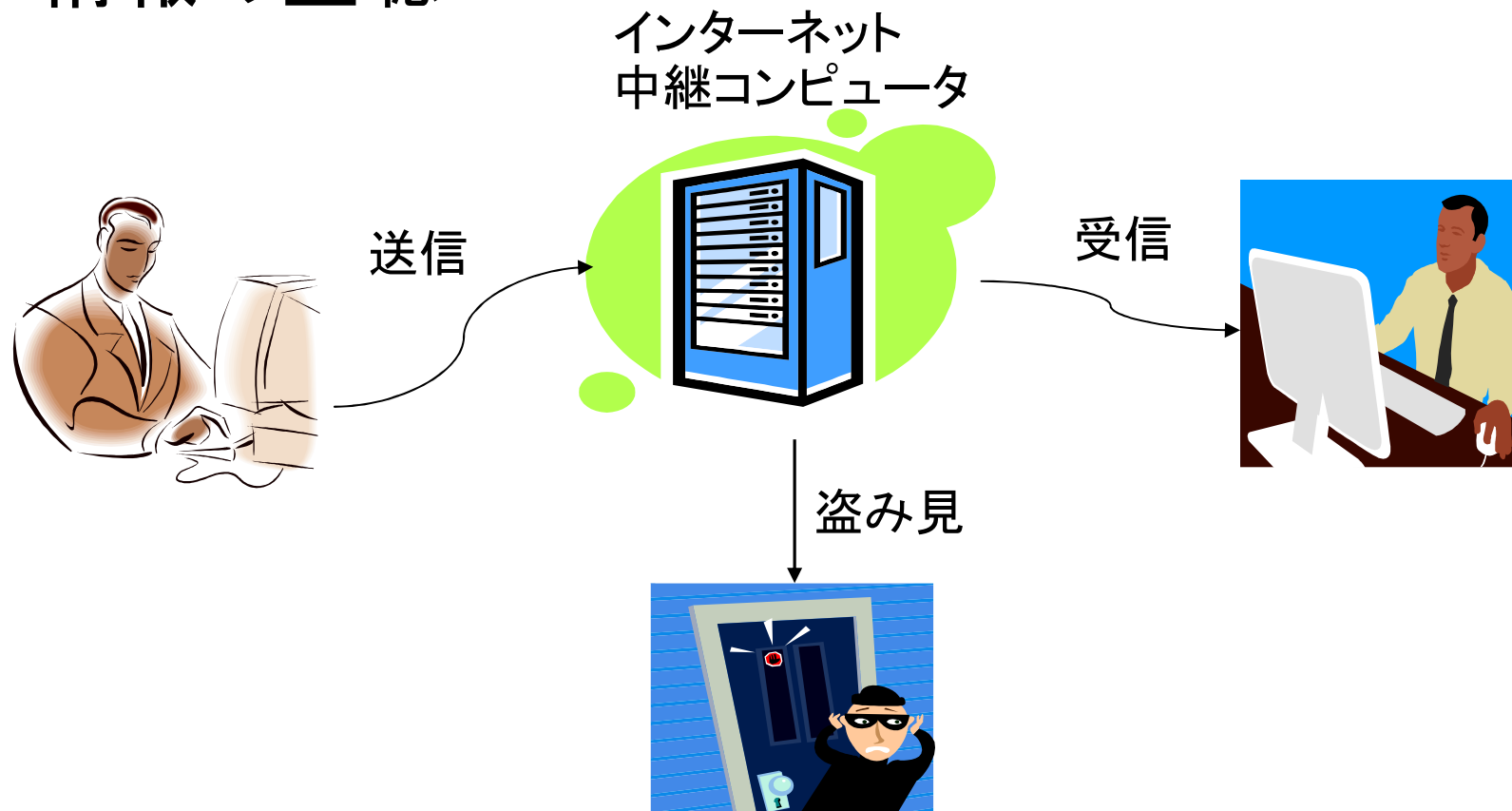
1.2 流れ解析の例(2/2)



角柱の周りの流れ解析:カルマン渦の発生原因

2. インターネットでの情報伝達

情報の盗聴



3. 情報を安全に送るには何が必要か

(1) 盗み見の防止は不可能

インターネットで情報を送る限りは、情報の盗み見を完全に防ぐことは不可能

(2) 読むことができないようにする

情報を「暗号化」して送信する。

送信者：情報を暗号化する。

受信者：復号化して情報に復元する。

4. 使用されている暗号を調べる

(1) 暗号化されたWeb

http : 暗号化されていないWeb

https : 暗号化されたWeb (**s**: security)

(2) 使用されている暗号を調べる

httpsのWebページを右クリック → プロパティ

(a) 接続: **SSL 3.0**、**RC4/128** ビット暗号 (高);
RSA / 1024 ビット交換

(b) 接続: **TLS 1.0**、**AES/128** ビット暗号 (高);
RSA / 1024 ビット交換

4.1 暗号化通信システム

(1) SSL (Security Session Layer)

SSL は、Netscape 社によって開発された通信路を暗号化する仕組み。

プライバシーに関わる情報やクレジットカード番号、企業秘密などを安全に送受信することができる。

RSA暗号で鍵を送付し、データ自体は共通鍵暗号で暗号化して送信する、ハイブリッド方式

(2) TLC (Transport Layer Security)

SSLを標準化団体IETFで標準化したもの

5. 共通鍵暗号と公開鍵暗号

(1) 共通鍵暗号

暗号化と復号化で**共通の鍵**を使用

利点：**高速**に処理できる

欠点：**鍵の送信が必要**である

(2) 公開鍵暗号

暗号化した鍵では**復号化**できない

利点：**鍵の送信が不要**(公開鍵で暗号化)

欠点：**共通鍵暗号**に比べ処理が**遅い**

5.1 共通鍵暗号

- (1) 紀元前50年にシーザーが使用
英字を指定数ずらす。(a,b,c,... → d,e,f,...)
- (2) 暗号化処理
ビットの転置、シフト、加算(桁上げなし)の組み合わせ。暗号化強度はビット数に依存。
- (3) 代表的な暗号
 - (a) DES (Data Encryption Standard) : 1977年に米国標準
 - (b) AES (Advanced Encryption Standard) : 2002年DESの後継
 - (c) RC4 : RSA暗号を開発したR.L.Rivestが開発した暗号

5.2 公開鍵暗号

(1) 1976年に公開鍵暗号を発表

共通鍵暗号の鍵の「配送問題」を解決する手段として開発。暗号化と復号化に別の鍵を用いる。

(2) 1978年にRSA暗号を発表

R.L.Rivest, A.Shamir, L.M.Adlemanの3名が開発したのでRSA暗号と言われる。

多数桁の因数分解の困難性を利用した暗号方式。
現在の公開鍵暗号は全てRSA/1024である。

6. RSAの仕組み

RSA暗号鍵の作成

- (1) 素数 p , q を選ぶ。
- (2) $n = p \times q$ 及び $f = (p-1) \times (q-1)$ を計算
- (3) 素数 e を選ぶ。
- (4) $d = 1/e \pmod{f}$ となる d を計算する。

(e,n) が公開暗号化鍵、 (d,n) が復号鍵となる。

6.1 暗号化と復号化

RSA暗号化

- (1) 文を n 以下の数 M に変換(公開方法)
- (2) $C=M^e \bmod(n)$ で暗号 C を作成

RSA復号化

- (1) $M=C^d \bmod(n)$ で元の数 M に復号
- (2) 数 M を文に変換(公開方法)

6.2 復号化の数学理論

$n=p \times q$ と M が互に素なら $M^{(p-1)(q-1)}=1 \pmod{n}$
なるオイラーの定理を使用

$$\begin{aligned} C^d \pmod{n} &= (M^e)^d \pmod{n} = M^{\alpha f+1} \pmod{n} \\ &= (M^{(p-1)(q-1)})^\alpha \times M \pmod{n} \\ &= M \pmod{n} = M \end{aligned}$$

$ed = 1 \pmod{f}$ で $ed = \alpha f + 1$ (α は整数)を利用

6.3 RSA暗号変換例

(1) 鍵とメッセージ

公開暗号鍵: $e=43291$ 、 $n=130733$

秘密復号鍵: $d=105691$

メッセージ: YES \Leftrightarrow $M = 16836$

(2) 暗号化

$$C = M^e = 16836^{43291} = 73724 \pmod{130733}$$

(3) 復号化

$$M = C^d = 73724^{105691} = 16836 \pmod{130733}$$

6.4 RSAの鍵の作成例

(1) 素数 p, q を選び n, f を計算

$p=239, q=547$: 選定

$n=p \times q=130733, f=(p-1) \times (q-1)=129948$

(2) 暗号鍵(素数, $e < f$)を選定

$e = 43291$, $(e, n) = (43291, 130733)$ を公開

(3) 復号鍵(秘密鍵, d)の計算

$d = 1/e = 1/43291 = 105691 \pmod{f}$

この計算はユークリッド互除法で計算する

7. RSA暗号の強度と暗号解読法

- (1) RSA暗号の強度(1024ビット(10進309桁))
RSA暗号の強度(解読の困難さ)は多数桁の因数分解の困難さに依存している。
- (2) 多数桁因数分解の世界記録
 - (a) 1996年 RSA-130(10進130桁)を因数分解
 - (b) 1999年 RSA-155(10進155桁)を因数分解
 - (c) 2005年 RSA-200(10進200桁)を因数分解
 - (d) 現在 RSA-704(10進212桁)がまだ未解読

7.1 因数分解の方法

(1) 篩(ふるい、Sieve)系解法

計算量は合成数の桁数に依存

RSA暗号の解読に都合が良い

MPQS、GNFSが代表的解法

(2) 楕円曲線法(Elliptic Curve Method,ECM)

計算量は小さい因数の桁数に依存

RSA暗号の解読には不向き

7.2 篩(ふるい)法

- (1) $A^2 - B^2 = (A - B)(A + B) \equiv 0 \pmod{N}$ の関係を使用し、 N を因数分解
- (2) $A_1^{l_1} \cdot A_2^{l_2} \cdots A_k^{l_k} \equiv B_1^{m_1} \cdot B_2^{m_2} \cdots B_j^{m_j} \pmod{N}$ なる関係を基底の数より多く集める
- (3) 0-1行列を計算し、両辺が平方になるもの(従属関係)のデータを探す
- (4) MPQS, GNFS等が代表的なふるい法

7.3 代表的なふるい法

- (1) **QS**(Quadratic Sieve、2次ふるい法)
MPQS(Multiple Polynomial QS、
複数次多項式2次ふるい法)が代表的解法
- (2) **GNFS**(General Number Field Sieve、
一般数体ふるい法)。現在、100桁程度
以上で最も高速な解法と言われている。
- (3) **MBPS**(Multiple Base Polynomial Sieve、
多重基底多項式ふるい法、**考案解法**)

7.4 QS(2次ふるい法)

1. MPQSの概要

QS: $f(x) = x^2 - n$: n は分解対象数

MPQS: $F(x) = (ax+b)^2 - n$, $f(x) = F(x)/a$

a, b の選び方により多数の $f(x)$ が得られる

2. MPQSの種類

(1) **MPQS** (Multiple Polynomial QS)

複数の a, b をある関係式より求める

(2) **SIQS** (Self-Initialization QS)

$b^2 - n \equiv 0 \pmod{a}$ なる関係から a, b を作成

7.5 GNFS(一般数体ふるい法)

(1) $f(x)=Ax^3+Bx^2+Cx+D$; d次多項式

$f(M)\equiv 0 \pmod{n}$, n ; 分解対象数

(2) a_k, b_k は下記を共に満たす整数

$$\prod (a_k + b_k \cdot M) = (\prod p_j^{s_j}) \equiv \alpha^2 \pmod{n}$$

$$\prod (a_k + b_k \cdot x) \equiv Q(x) \equiv q(x)^2 \pmod{f(x)}$$

$$q(M) \equiv \beta \pmod{n}, p_j \text{ は素数}$$

(3) $\alpha^2 \equiv \beta^2 \pmod{n} \rightarrow n$ を因数分解

7.6 2次多項式によるMBPS

(1) N: 分解対象の合成数 (DBPS:(1),(2),(3))

$$f(x) = Ax^2 + Bx + C, \quad f(M) \equiv 0 \pmod{N}$$

(2) 篩に利用する関数 ($A_1A_2=A$)

$$g(x) = (A_1x+a)(A_2x+b) - f(x) = G(sx+t)$$

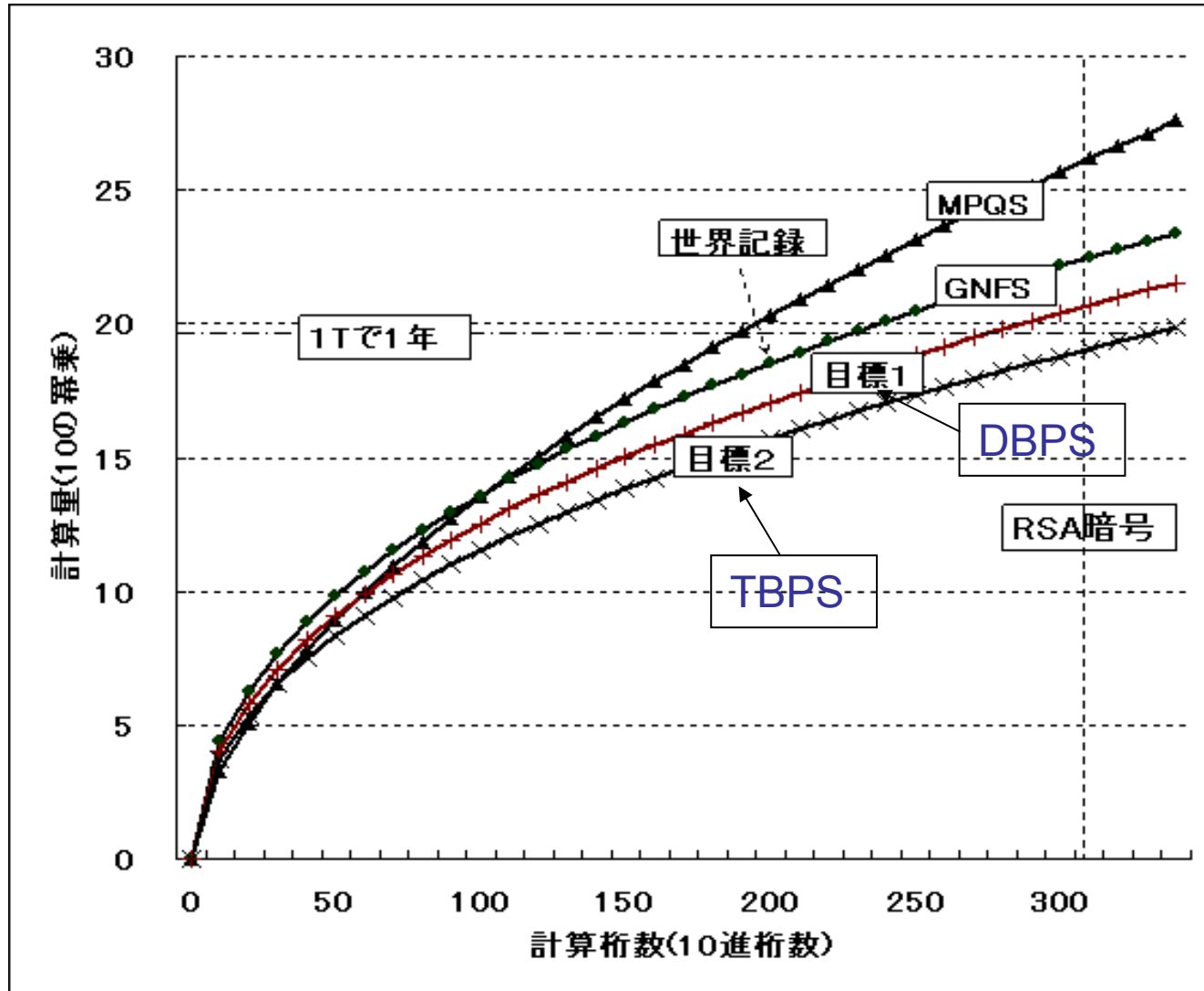
$$Gs = A_1b + A_2a - B, \quad Gt = ab - C$$

$$(A_1M+a)(A_2M+b) \equiv G(sM+t) \pmod{N}$$

(3) 素数基底と1次式でふるいを実施

(4) $f(\theta)$ でのGNFSと結合したふるい (TBPS)

7.7 因数分解の計算量推定



7.8 研究室の目標

(1) 2年後の目標

- (a) RSA-768(10進232桁)の解読
- (b) 計算法: DBPS(2重基底多項式ふるい法)
→ プログラム作成中

(2) 5年後の目標

- (a) RSA-1024(10進309桁)の解読
- (b) 計算法: TBPS(3重基底多項式ふるい法)
→ TBPSの問題点(自明解増大)の対策が必要