情報・システム工学概論

公開鍵暗号の数理（1回目）

高木　剛

東京大学工学部計数工学科

現代社会と暗号技術

昔の暗号

限られた人だけが使う特殊技術

現代の暗号

身近なもの

電子政府

個人認証、プライバシー保護

電子決済、仮想通貨

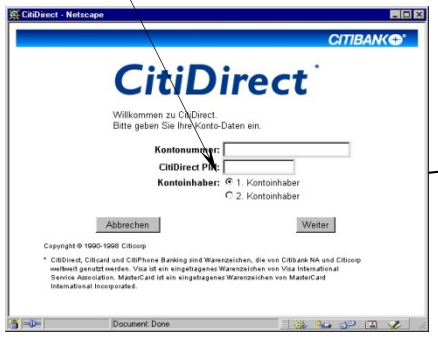
著作権保護、コピー防止

電気自動車、IoTセキュリティ

暗号は現代社会に無くてはならない技術

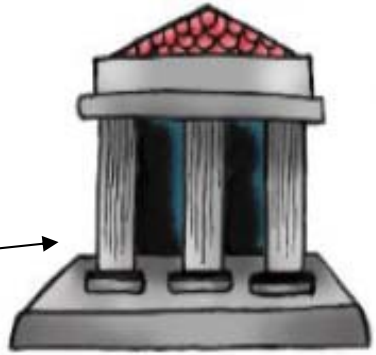
メールサーバへのログイン

パスワード (1205)



インターネット

パスワード (1205)

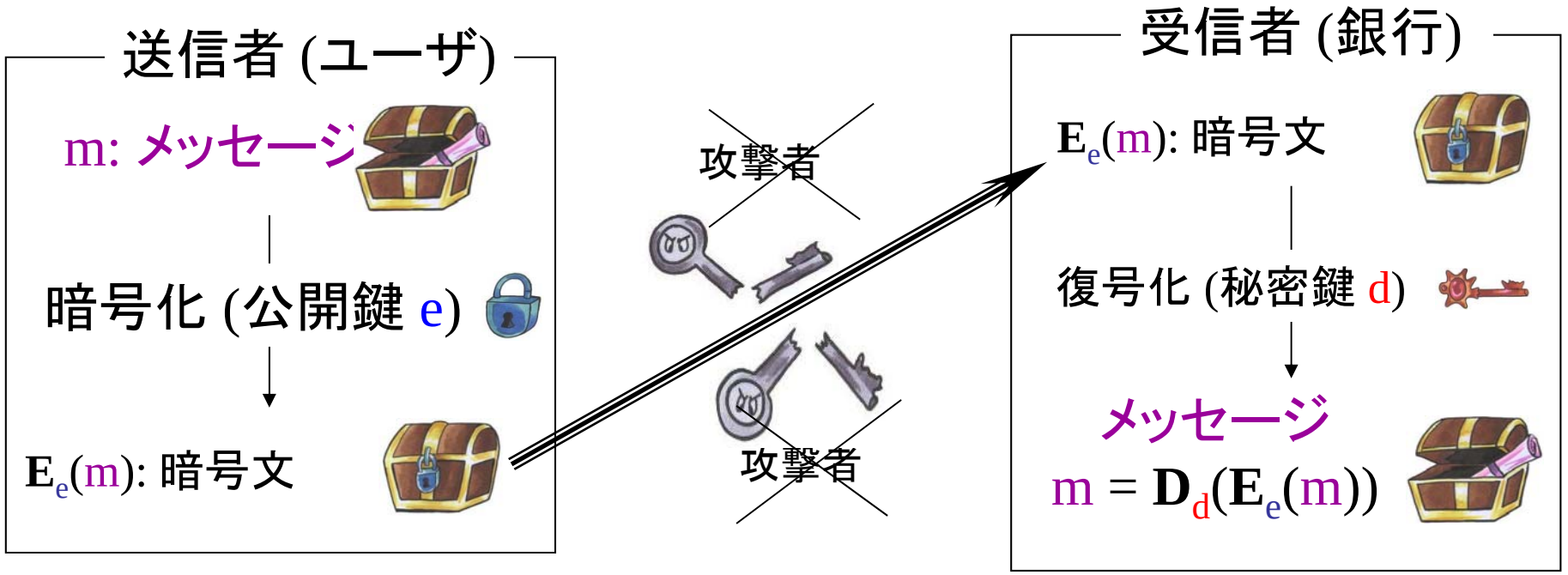


メールサーバ
u-tokyo.ac.jp

公開鍵暗号

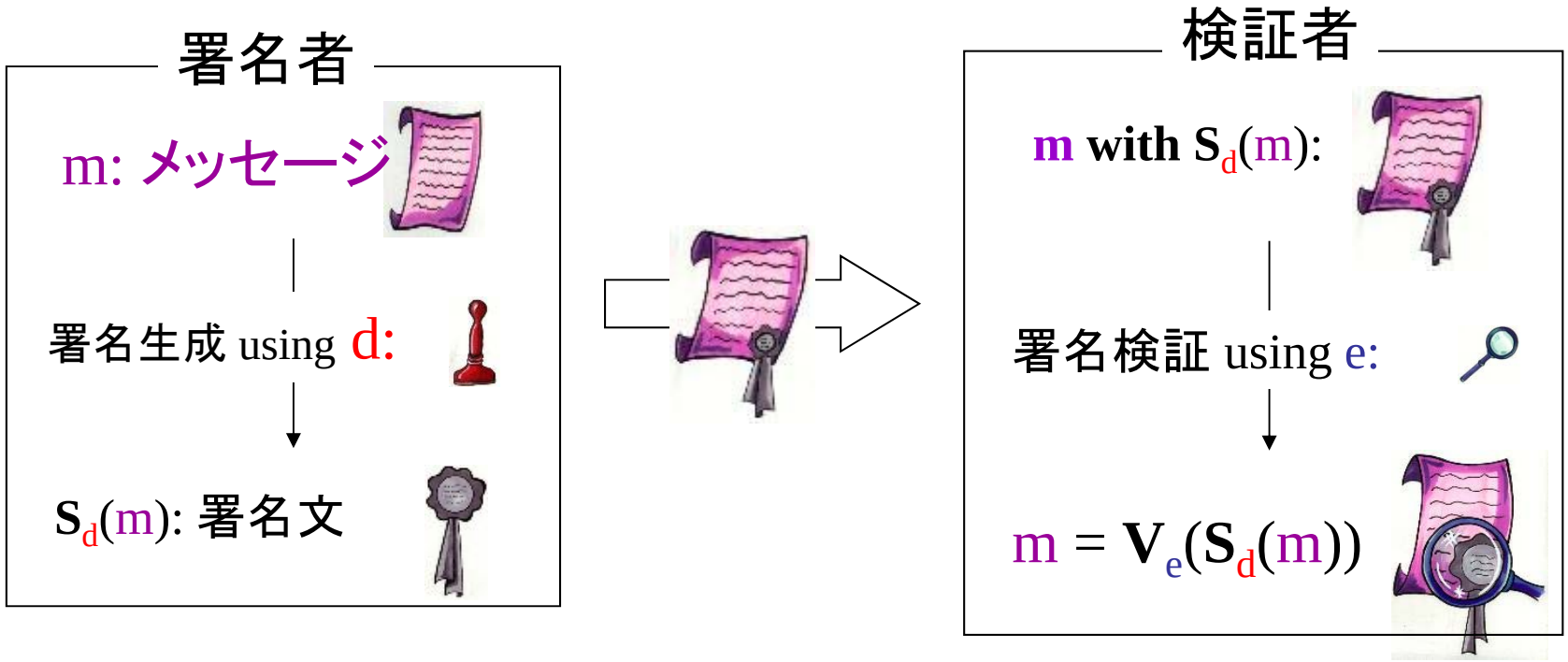
(e: 公開鍵  , d: 秘密鍵 ) 受信者が所有する (銀行)

E_e: 暗号化関数, D_d: 復号化関数, $m = D_d(E_e(m))$



ディジタル署名

署名者の(e: 公開鍵, d: 秘密鍵)
S_d: 署名生成関数, V_e: 署名検証関数, $m = V_e(S_d(m))$



x mod y

- 任意の整数x,yに対して、

  $x = q\,y + r,\ \ 0 \leqq r < q$

  を満たす整数q,rが一意的に存在する。

  r = x mod yと記述する。

- 例:　178 mod 35 = 3　　(∵ 178=5×35+3)

RSA暗号

- 1977年にRivest, Shamir, Adlemanが提案

[鍵生成]　素数p,qに対してn=pqとし、$ed = 1 \bmod (p-1)(q-1)$
を満たす整数e,dを生成。公開鍵 (e,n), 秘密鍵d。

[暗号化]　$Z_n = \{0,1,2,\ldots,n-1\}$, 平文$m \in Z_n$。
$c = m^e \bmod n$

[復号化]　$m = c^d \bmod n$

RSA暗号の原理

- フェルマーの小定理＋中国人剰余定理

$\gcd(m,n)=1 \rightarrow m^{(p-1)(q-1)} = 1 \bmod n$

- 鍵生成より$ed = 1 + k(p-1)(q-1)$を満たす整数kが存在する。

  復号化 $c^d \bmod n$

  $= (m^e)^d \bmod n \quad (\leftarrow c = m^e)$

  $= m^{1+k(p-1)(q-1)} \bmod n$

  $= m\,(m^{(p-1)(q-1)})^k \bmod n$

  $= m$

RSA暗号化の例

e=13, n=77 を公開鍵とする。

メッセージ m=20 を暗号化せよ。

暗号文 $c = m^e \bmod n = 20^{13} \bmod 77 = 69$

どうやって20^{13}を77で割った余りを高速に求めるか？

高速べき乗計算

$20^{13} = 20^{8+4+1} = 20^{2^3+2^2+1} = (((20^2) \times (20))^2)^2 \times 20$

$20 \rightarrow 20^2 \bmod 77 = 15$

$\rightarrow (20^2) \times (20) = 15 \times 20 \bmod 77 = 69$

$\rightarrow ((20^2) \times (20))^2 = 69^2 \bmod 77 = 64$

$\rightarrow (((20^2) \times (20))^2)^2 = 64^2 \bmod 77 = 15$

$\rightarrow (((20^2) \times (20))^2)^2 \times 20 = 15 \times 20 \bmod 77 = 69$

RSAの落と戸は？

秘密鍵 d に対して、以下で復号化できる。

$m = c^d \bmod n$

前頁の例では d=37 とする。

$69^{37} \bmod 77 = 20$ を確かめよ。

復号化検証

$69^{37} = 69^{32+4+1} = 69^{2^5+2^2+1} = (((69^2)^2)^2 \times (69))^2)^2 \times 69$

$69 \rightarrow 69^2 \bmod 77 = 64$

$\rightarrow (69^2)^2 = 64^2 \bmod 77 = 15$

$\rightarrow ((69^2)^2)^2 = 15^2 \bmod 77 = 71$

$\rightarrow (((69^2)^2)^2) \times (69) = 71 \times 69 \bmod 77 = 48$

$\rightarrow (((69^2)^2)^2) \times (69))^2 = 48^2 \bmod 77 = 71$

$\rightarrow (((69^2)^2)^2) \times (69))^2)^2 = 71^2 \bmod 77 = 36$

$\rightarrow (((69^2)^2)^2) \times (69))^2)^2 \times 69 = 36 \times 69 \bmod 77 = 20$

高速べき乗法

入力　　**c, n, d**

出力　　$\mathbf{c^d \bmod n}$

x= c;

For i=k-2 to 0

　x = x^2 mod n;

　if d[i]=1, then x=x*c mod n;

Return x

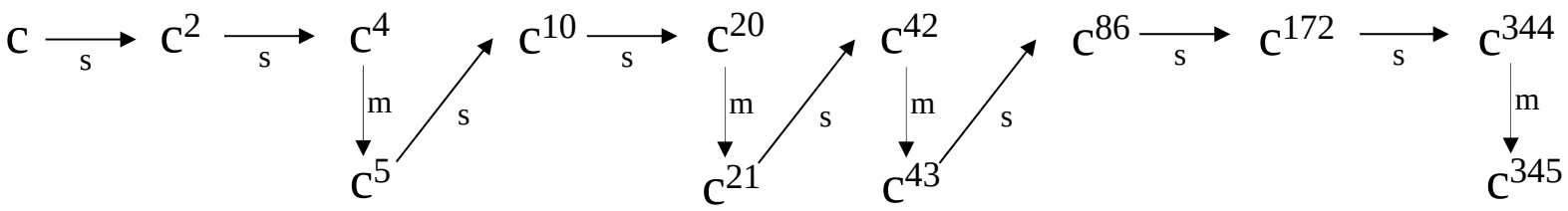
バイナリ法

$d = d[k-1]2^{k-1}+d[k-2]2^{k-2}+\ldots+d[1]2^1+d[0]2^0.$

$d[k-1]=1$

Example d = 345

345 = 1　0　1　0　1　1　0　0　1



問題

- n=247, c=2, d=216とするとき、

冪乗算 $2^{216} \bmod 247$ を計算せよ。

RSA暗号の安全性

- 公開鍵nが素因数分解されるとp,qが求まる。

- 公開鍵eから、関係式

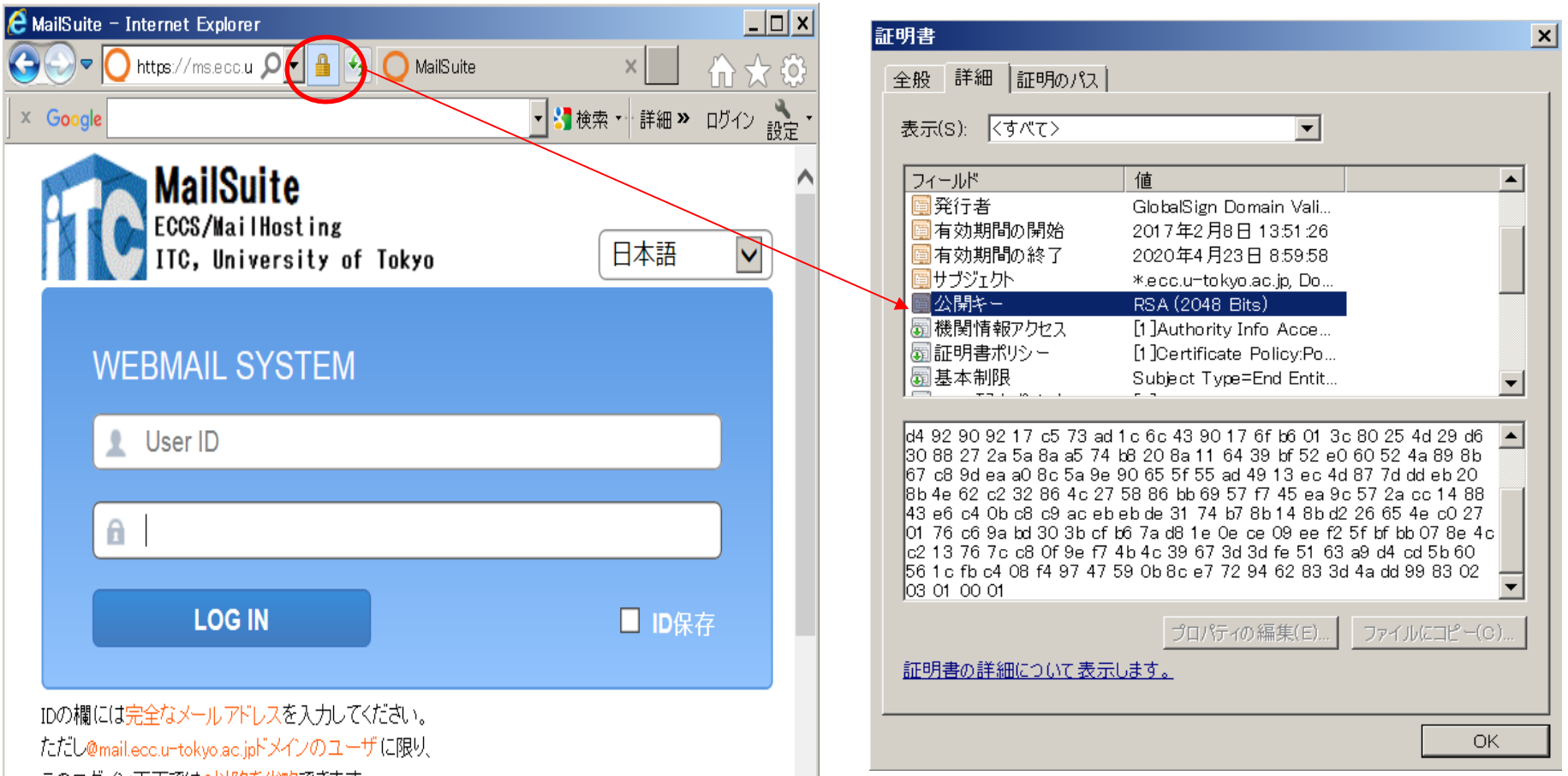
  $d = 1/e \bmod (p-1)(q-1)$

  により、秘密鍵dを求めることができる。

- RSA暗号の安全性は、素因数分解の困難性に根拠を置いている。

素因数分解

暗号鍵の例


MailSuite – Internet Explorer
https://ms.ecc.u
MailSuite
Google
検索
詳細 »
ログイン
設定
MailSuite
ECCS/MailHosting
ITC, University of Tokyo
日本語
WEBMAIL SYSTEM
User ID
LOG IN
ID保存
IDの欄には完全なメールアドレスを入力してください。
ただし@mail.ecc.u-tokyo.ac.jpドメインのユーザに限り、
このログイン画面では@以降を省略できます。
証明書
全般
詳細
証明のパス
表示(S):
<すべて>
フィールド
値
発行者
GlobalSign Domain Vali...
有効期間の開始
2017年2月8日 13:51:26
有効期間の終了
2020年4月23日 8:59:58
サブジェクト
*.ecc.u-tokyo.ac.jp, Do...
公開キー
RSA (2048 Bits)
機関情報アクセス
[1]Authority Info Acce...
証明書ポリシー
[1]Certificate Policy:Po...
基本制限
Subject Type=End Entit...
d4 92 90 92 17 c5 73 ad 1c 6c 43 90 17 6f b6 01 3c 80 25 4d 29 d6
30 88 27 2a 5a 8a a5 74 b8 20 8a 11 64 39 bf 52 e0 60 52 4a 89 8b
67 c8 9d ea a0 8c 5a 9e 90 65 5f 55 ad 49 13 ec 4d 87 7d dd eb 20
8b 4e 62 c2 32 86 4c 27 58 86 bb 69 57 f7 45 ea 9c 57 2a cc 14 88
43 e6 c4 0b c8 c9 ac eb eb de 31 74 b7 8b 14 8b d2 26 65 4e c0 27
01 76 c6 9a bd 30 3b cf b6 7a d8 1e 0e ce 09 ee f2 5f bf bb 07 8e 4c
c2 13 76 7c c8 0f 9e f7 4b 4c 39 67 3d 3d fe 51 63 a9 d4 cd 5b 60
56 1c fb c4 08 f4 97 47 59 0b 8c e7 72 94 62 83 3d 4a dd 99 83 02
03 01 00 01
プロパティの編集(E)...
ファイルにコピー(C)...
証明書の詳細について表示します。
OK

素因数分解問題

問題: 次の整数を素因数分解せよ。

- 15
- 187
- 1961
- 16637
- ・・・

素因数分解問題

問題: 次の整数を素因数分解せよ。

- $15 = 3 \times 5$
- $187 = 11 \times 17$
- $1961 = 37 \times 53$
- $16637 = 127 \times 131$
- ・・・

素因数分解の解読世界記録

- 2010年1月、231桁、約1500年×CPU（Opteron 2.2 GHz）

- 1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413
  =
  3347807169895689878604416984821269081770479498371376856891243138898288379387800228761471165253174308773781446799948 9
  ×
  3674604366679959042824463379962795263227915816434308764267603228381573966651127923337341714339681027009279873630891 7

RSAチャレンジ問題の解読記録推移

解読達成日	解読桁長	計算時間(パソコン1台換算)
1991年4月	100桁 (330ビット)	
1993年6月	120桁 (397ビット)	
1996年4月	130桁 (430ビット)	約7年 (1500 MIPS年)
1999年2月	140桁 (463ビット)	
1999年8月	155桁 (512ビット)	約36年 (8000 MIPS年)
2003年12月	174桁 (576ビット)	
2005年5月	200桁 (663ビット)	約55年 (Opteron 2.2 GHz)
2009年12月	231桁 (768ビット)	約1500年 (Opteron 2.2 GHz)
未解読	308桁 (1024ビット)	
未解読	616桁 (2048ビット)	

スーパーコンピュータ

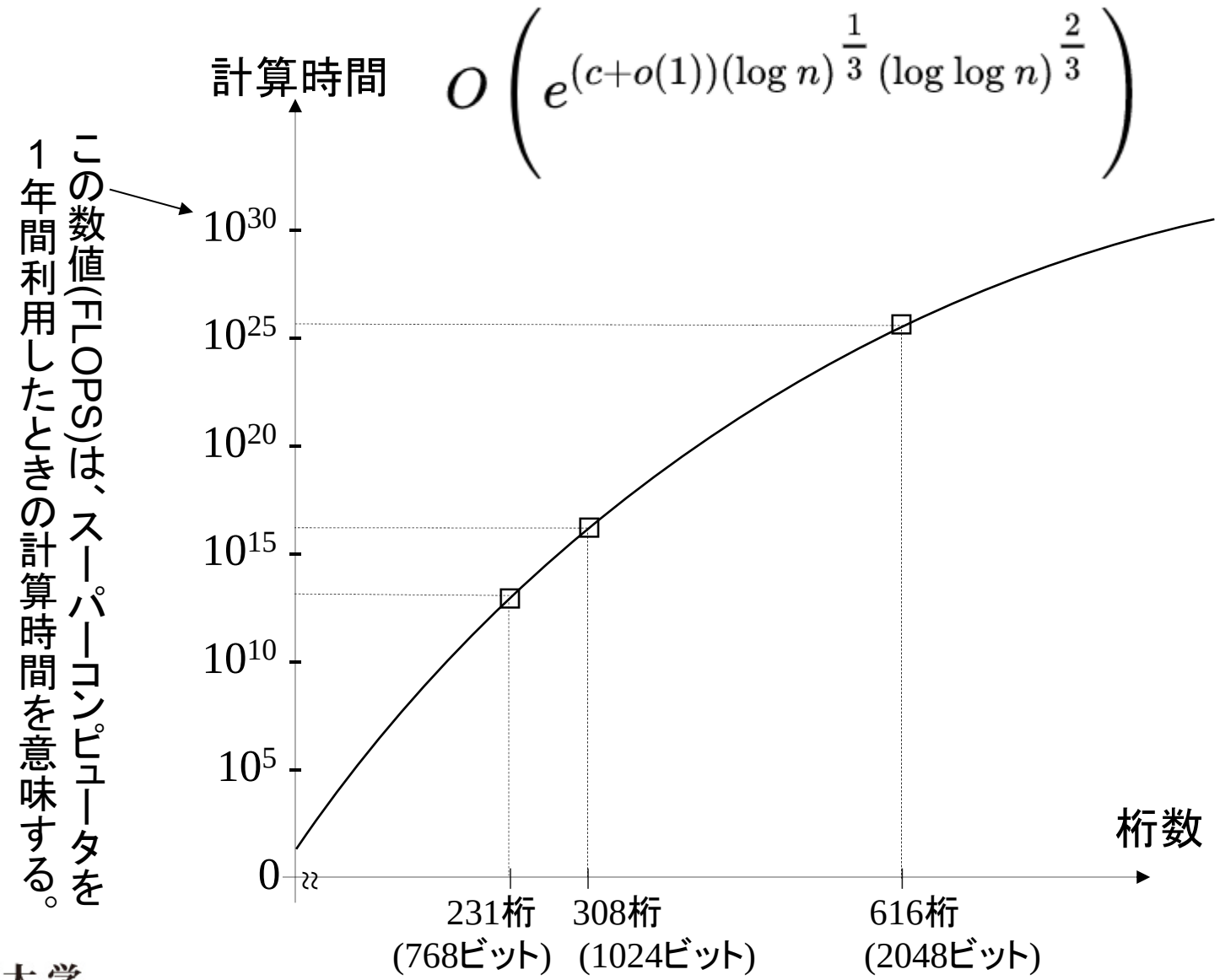
FLOPS (Floating point number Operations Per Second)
浮動小数点演算が1秒間に何回できるかの単位

スーパーコンピュータ「京」は、およそ1京(10^{16})FLOPS

エクサ	10^{18}	= 1,000,000,000,000,000,000	**百京**	← スーパーコンピュータ（2018年）
ペタ	10^{15}	= 1,000,000,000,000,000	**千兆**	
テラ	10^{12}	= 1,000,000,000,000	**一兆**	← 高性能なスマートフォン（2018年） スーパーコンピュータ（1993年）
ギガ	10^{9}	= 1,000,000,000	**十億**	
メガ	10^{6}	= 1,000,000	**百万**	
キロ	10^{3}	= 1,000	**千**	

安全な鍵長を選ぶには、ムーアの法則を考慮に入れて考察する必要がある。

数体篩法


計算時間
$O\left(e^{(c+o(1))(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}}\right)$
この数値(FLOPS)は、スーパーコンピュータを1年間利用したときの計算時間を意味する。
10^{30}
10^{25}
10^{20}
10^{15}
10^{10}
10^{5}
0
桁数
231桁 (768ビット)
308桁 (1024ビット)
616桁 (2048ビット)

素因数分解の困難性に関する計算量評価

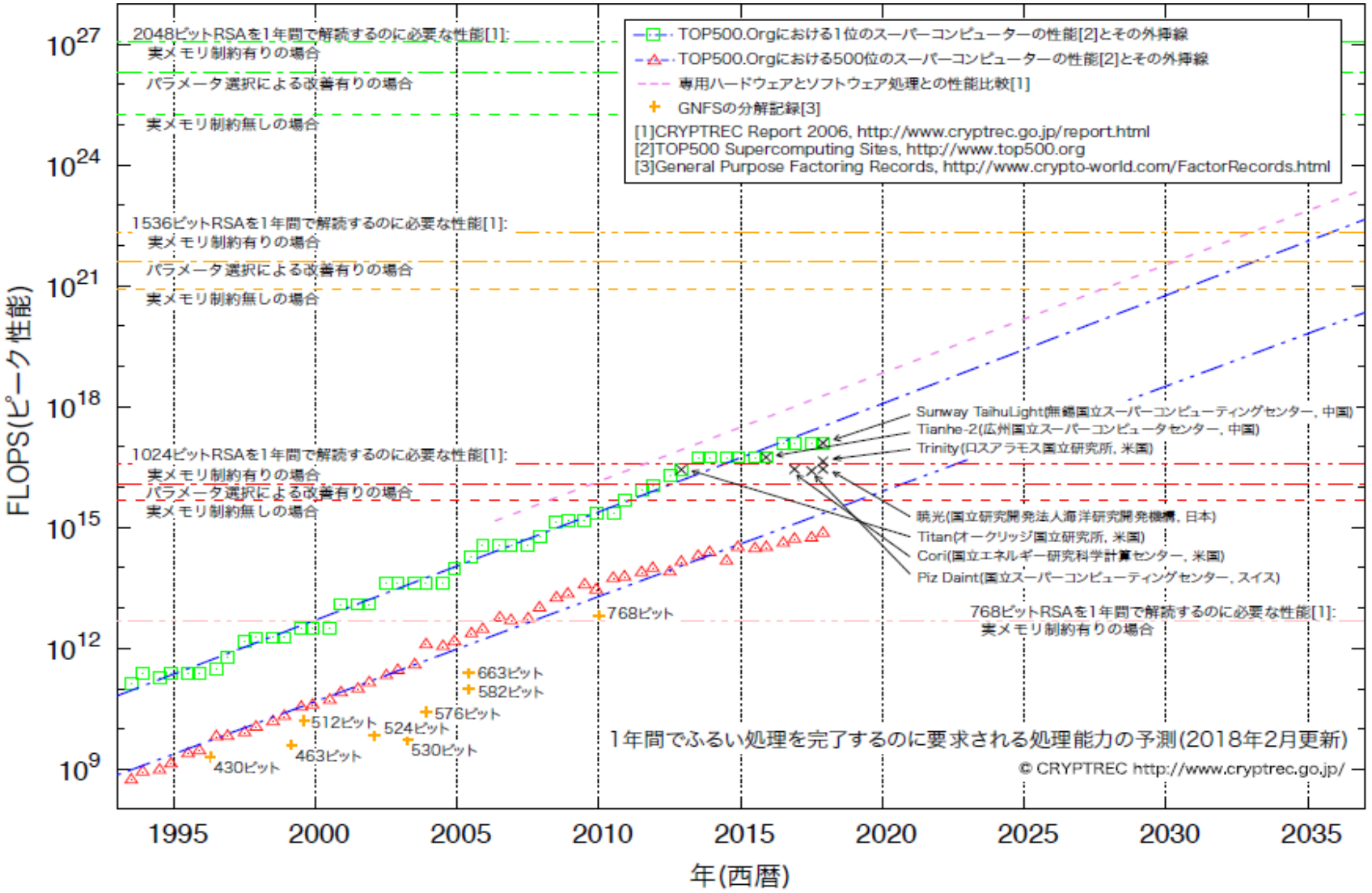
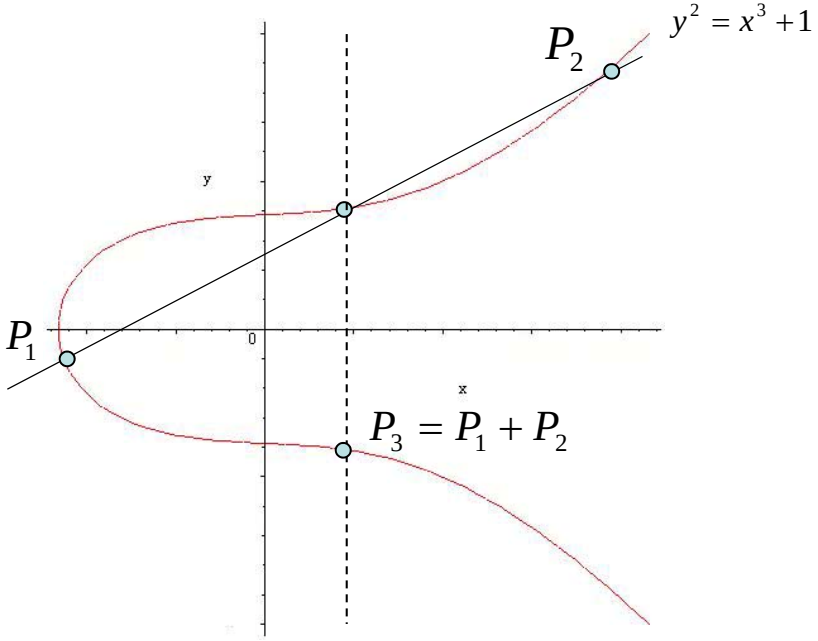


図１：素因数分解の困難性に関する計算量評価

(http://www.cryptrec.jp/)

楕円曲線暗号

- 1985年にMillerとKoblitzが独立に提案
- $E:=\{(x,y)\in(Z_p)^2 \mid y^2 = x^3+ax+b\}\cup\{\infty\}$ は群構造を持つ
- Weil-Hasse定理から#E~pを満たす
- #Eは素数位数ℓ



楕円曲線暗号

[共通パラメータ]　E：素数位数ℓの楕円曲線、生成元をPとし、
(E,P)をシステムの共通パラメータとして公開する。

[ユーザ鍵生成] 秘密鍵s$\in Z\ell$ ={0,1,2,…,ℓ−1} に対して、Q=sPとする。公開鍵Q, 秘密鍵s。

[暗号化]　楕円曲線E上の点を平文Mとして、共通パラメータP,
公開鍵Q、乱数r$\in Z\ell$ に対して、次を計算する。

$$(C_1,C_2) = (rP,rQ+M)$$

[復号化]　$M = C_2 - sC_1$
$= (rQ+M) - s(rP)$
$= srP + M - srP$

楕円曲線暗号の安全性

- 楕円離散対数問題: 入力 P,sPに対して、sを求める。
- DH問題: 入力 P,sP,rP に対して、srP を求める。

- 楕円曲線暗号の一方向性 ⇔ DH問題の困難性

- 数体篩法など準指数時間の解読法が知られていない。
- 最も高速な解読法はポーラードのρ法： 計算量 $O(\sqrt{p})$

⇨ 256ビットなどRSA暗号より短い鍵長で同レベルの安全性

公開鍵の例： 04 c5 fe b9 8d b2 28 9f 47 4a 05 93 2d 0b 85 79 cc 94 09 9a 79 cd 96 e1 10 ae d8 0b 89 b3
02 e2 30 a6 1c ae 53 ac 53 b9 6f 32 3a dd cf e8 01 96 aa 42 23 34 c7 a2 c9 9d 32 33 45 8c 03 c4 c0 71 23

Thank you, Questions?