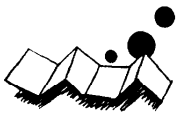
解　説



暗号アルゴリズムと計算量の理論†

嵩　忠雄††　藤原　融††

1. はじめに

暗号解読の計算上の困難さに関する問題を，主に公開鍵方式について概説する．現在の計算量の理論は，暗号の安全性問題に直接解答を与えないが，それらを考える枠組を提供し，また問題を位置付ける上で貢献できよう．本稿では，まず暗号問題に応用の可能性の大きい確率的アルゴリズムを中心として，計算の複雑さの問題を説明する．次に，現在，公開鍵方式は，整数の因数分解か離散的対数問題の困難さに基づく2つの系統にしぼられてきた観があるので，それら2つの問題がどの程度計算上困難な問題であるかを概説する．最後に，最近相ついで解読法が発表されたナップザック問題に由来する暗号系の解読にふれる．

2. 暗号方式の概略

2.1 鍵の生成，暗号化と復号化

ブロック暗号系では，次の3つの操作のために，**能率のよい**アルゴリズムが提供されている[4),5)]．

1) **鍵の生成**：無作為に復号化鍵 k_d を生成する．非対称鍵方式（公開鍵方式）では，k_d に対応する暗号化鍵 $k_e(\neq k_d)$ も生成する．k_e は公開鍵，k_d は受信者のみが知っている秘密鍵である．対称鍵方式（伝統的方式）では，$k_d=k_e$ である．k_e と k_d は，正当な送受信者のみが知っている秘密鍵である．

2) **暗号化**：与えられた暗号化鍵 k_e と平文 m から暗号文 c を作成する．

3) **復号化**：与えられた復号化鍵 k_d と暗号文 c から，元の平文 m を求める．

例 1：RSA 暗号方式[6)]

1) 鍵の生成：各受信者は秘密裡に異なる素数 p と q を選ぶ（例えば，10進100桁位の数を無作為に選び，例3で示すような素数判定アルゴリズムで判定し，素数が求まるまで繰り返す．安全上の考慮から，さらに $p-1$，$q-1$ が大きな素因数を含む必要があるので，これらの条件を満たす p，q が見つかるまで繰り返す．素数の分布定理から繰り返し回数の平均値が押えられる）．$n=pq$ とおく．復号化鍵 k_d を $(p-1)$ と $(q-1)$ の最小公倍数 L と互に素になるように選び，また暗号化鍵 k_e を，$k_ek_d\equiv 1 \pmod L$ を満たすように求める．平文と暗号文の集合は，$n-1$ 以下のすべての非負整数からなる．受信者は，n と k_e を公開ファイルに登録する．

2) 暗号化：$c\equiv m^{k_e} \pmod n$

3) 復号化：$m\equiv c^{k_d} \pmod n$

文献 6)に，これらを実行するための能率のよいアルゴリズムが与えられている．

対称鍵（暗号）方式では，秘密鍵を通信当事者に，いかに安全に配るかが問題であるが[5),7)]，次の方式が提案されている．

例 2：Diffie-Hellman の分配鍵方式[4)]

p を $p-1$ が大きな素因数をもつ，例えば，10進，200 桁位の素数とし，有限体 GF(p) の原始元の1つ g と共に公開されているとする．AがBと暗号通信を行う場合，Aは区間 $I=[1,\ p-1]$ から無作為に，1つの整数 x_A を選び（秘密にしておく），$y_A\equiv g^{x_A} \pmod p$，$y_A\in I$ である y_A をBへ送る．Bは区間 I から無作為に整数 x_B を選び（秘密にしておく），$y_B\equiv g^{x_B} \pmod p$，$y_B\in I$ である y_B を A に送る．A は

$$k\equiv y_B^{x_A}\equiv g^{x_Ax_B} \pmod p,\quad k\in I$$

を計算し，kを共通の秘密鍵とする．Bも同様に，

$$k\equiv y_A^{x_B}\equiv g^{x_Ax_B} \pmod p$$

を求め，鍵とする．

2.2 暗号方式の安全性

復号化鍵 k_d（対称鍵方式では，もちろん暗号化鍵でもある）を‘直接知らず’に，暗号文 c からもとの平文 m（またはそれに関する部分情報，例えば，mの最後のビット）を求めることを，暗号の解読という．解読に利用できる情報として，(1)平文集合，暗号文

† Cryptanalysis and Theory of Complexity by Tadao KASAMI Tohru FUJIWARA (Dept. of Information and Computer Sciences, Osaka University).

†† 大阪大学基礎工学部情報工学科

集合，鍵集合の定義，(2)鍵生成，暗号化，復号化のアルゴリズム，さらに(3)非対称鍵方式なら，復号化鍵 k_d に対応する公開の暗号化鍵が考えられる（もちろん，一般の知識も利用する）．その上に，

(i) いくつかの平文とそれに対する暗号文を知っているとき，known plaintext attack と呼ばれる．

(ii) 解読者が，解読上の必要から選んだいくつかの平文に対する暗号文を知ることができるとき，chosen plaintext attack と呼ばれる．

(iii) (ii)と同様に，解読者が選んだいくつかの暗号文に対する平文（意味の通らないようなものでもよい）を知ることができるとき，chosen ciphertext attack と呼ばれる．

なお，非対称鍵方式においては，暗号化鍵が公開されているから，解読者は任意に選んだ平文に対する暗号文を求めることができるので，(i)，(ii)はいつも可能である．

暗号の解読による利得の方が，解読のための計算費用を上回るなら，そのような暗号系の実用性は乏しい．無視できる程度の例外を除いて，暗号文の解読が過大な計算時間を要し，実際上不可能であるとき，**暗号系は安全**であるといわれる．

暗号方式では，鍵生成，暗号化，復号化はできるだけ容易で，暗号の解読はできるだけ難しいことが要求される．次章で，計算量の理論の立場から定式化を考える．なおディジタル署名については，7)を参照されたい．その安全性の問題は，基本的に暗号の安全性のそれと変わらない．

3. 計算量の理論から

3.1 問題の難易をどのように取り扱うか

例えば，次のような問題を考える．

Q_1：$2^{128}+1$ の素因数を1つ求めよ[8]．

Q_2：$2^{128}\leq x<2^{128}+2^{20}$ の範囲の任意の正整数 x が与えられたとき，x の素因数を1つ求めよ．

Q_3：任意の正整数 x が与えられたとき，x の素因数を1つ求めよ．

Q_1 は Q_2，Q_3 において，x が特定の値をとる場合であり，Q_2，Q_3 の個別問題と呼ばれ，Q_2，Q_3 は個別問題の集合と見なされる．個別問題を指定するパラメータ（入力ともいう）x を，2進数で表わし，その桁数を $|x|$ と書き，入力のサイズと呼ぶ．

Q_2 のように有限個の個別問題を含む問題について，それを解く'最良'のプログラムの標準計算機上での計算時間によって，その難易を表わそうという考えは，ごく自然と思われよう．しかし，仮に，可能な入力に対する計算時間の最大値，平均値とか中央値をプログラムの良さの尺度に選んだとすると，予め可能な全入力について(Q_2 では 2^{20} 個) 答を求めて表を作っておき，実行時には入力を見てその表を引くだけのプログラムが最良となろう．すなわち，入力サイズが限定されると，プログラム作成のための計算と実行時の計算とのトレードオフが可能となり，前者を無視することはできない．また，表作成のための計算とプログラマが'考える'ことを区別するのは難しい．

計算量の理論では，このような微妙な問題を避けるため，Q_1, Q_2 のような問題の難易を直接議論せず，Q_3 のように無限の個別問題を含む問題を対象とし，それを解くプログラムの所要計算時間が，入力サイズが大きくなるにつれて，漸近的にどのように増大するかによって問題の難易を定義する[9]．

3.2 確率的アルゴリズム

通常の計算機の基本演算及び

乱数発生操作：正整数 m が与えられたとき，0以上 $m-1$ 以下の整数を無作為に1つ選び出力する．

を用いた手続き（一般には，誤りとか出力せずに停止することを許す）を確率的アルゴリズム[1)~3)]（以下，アルゴリズムと略称する）という．乱数発生を用いていないことを強調する場合は，決定性アルゴリズムともいう．

問題に含まれる個別問題を指定するパラメータがとる値の集合をその問題の入力集合と呼ぶ．一般にすべての入力に対して，必ずしも正解を与えることは要求されていないとし，ある程度，回答しないこと，さらに誤った答も許されるとする．D を入力集合のある部分集合とし，D に属する入力に対しては，回答失敗率，誤り率を小さく押える必要があるが，D に属さない入力に対しては問題にしないとする．

問題 Q の入力集合の任意の元 x に対して，確率的アルゴリズム A が，

1) $h(|x|)$（ただし，h は非負整数を値とする単調関数）ステップ（ビット操作を単位とする）以内に，出力を与えて停止するか，出力を与えずに停止（回答の失敗を意味する）するかのいずれかであり，

2) D に属する同一の x について，初期状態から試行を繰り返すとしたとき，1回の試行で，失敗する確率が φ 以下 ($0\leq\varphi<1$)，また間違った答を出力する確率も ε 以下 ($0\leq\varepsilon<1$) であるとき，

Aは，問題Qを**h時間**に，入力部分集合Dに対して**失敗率 φ，誤り率 ε** で解くという．(φ, ε は一般に入力サイズの関数であってよい．) 特に h が多項式であるとき，多項式時間（確率的）アルゴリズムという．

例 3：素数判定問題「与えられた正整数 n が素数か否かを判定する」の Rabin の確率的アルゴリズム[2]．

1) 無作為に $1<i<n$ である整数 i を選ぶ．

2) 判定：i^{n-1} を n で割ると余りが1でないか，あるいは $n-1$ が 2^k で割切れ ($r=(n-1)/2^k$ とおく)，(i^r-1) と n が，1でも n でもない最大公約数を持つような正整数 k が存在するなら，'No' を出力し，停止．

3)，2) の条件が成立しないなら，m 回まで，1)，2) を繰り返す．m 回繰り返しても 2) の条件が一度も成立しないなら，'Yes' を出力し停止．

n が素数なら 2) の判定条件を満たす i は存在しないが，n が素数でないならば，2から $n-1$ までの整数の中の半数以上が 2) の条件を満たす[2]．したがって，

(i) n が素数なら，必ず 'Yes' を出力する．

(ii) n が合成数であるにも拘らず，誤って 'Yes' を出力する確率は 2^{-m} 以下である．'No' を出力すれば必ず，n は合成数である．

2) の判定は，$(\log n)^3$ のオーダのビット操作で可能である．なお，現在知られている最も効率的な決定性の素数判定アルゴリズムの計算時間は $(\log n)^{\alpha \log\log\log n}$ (α：定数) のオーダである[10]．

3.3 易しい問題と難しい問題

すべての入力に対して，任意に与えられた小さな正数より小さい失敗率，誤り率で，問題Qを解く多項式時間アルゴリズムが**知られているならば**，Qは**易しい**問題であるという．実用上は，さらに多項式の次数が低いことが必要である．上の条件を満たすアルゴリズムが**知られていないとき**，Q は **'難しい'** (引用符をつける) ということにし，また上の条件を満たすアルゴリズムが**存在しない**問題を，**難しい**問題という．一般に，与えられた条件を満たすアルゴリズムが存在しないことを証明するのは非常に困難である．非対称鍵暗号方式では，鍵の生成，暗号化と復号化が易しい問題であることが要求されているので，暗号の解読問題は，次の制約を満たすクラス (付録(2)の $(NP \cap co\,NP)^*$) に属する[11],[12]．

1) 入力 x (暗号文) に対する解 y (対応する平文) のサイズは $|x|$ の多項式オーダであり，

2) 解を見つけることは'難しい'が，与えられた y が x に対する解であるか否かを判定することは易しい問題である．

このクラスのどの問題についても，それに対する多項式時間の決定性アルゴリズムが存在しないことは証明されていない．このクラスが難しい問題を含むか否かは，計算量理論の重要な未解決問題の1つである．すなわち，非対称鍵暗号の解読問題は '難しい' 問題ではあるが，よほどの進展がない限り引用符がなくなる見込みはなく，個々の暗号系では，逆に易しい問題であることが示されるおそれもある．

難しさを保証する次善の策としてよく用いられるのは次のような議論である．問題Qと同程度に難しいと思われ，より標準的で研究の歴史も古い問題 Q_0 を選び，もしQが仮に易しい問題であれば，Q_0 も易しい問題になってしまうことを示す．そのために，任意の入力に対して，Qの解を (適当な失敗率，誤り率で) 1ステップで出力する仮想的アルゴリズム (**オラクル**と呼ばれる[9]) を想定し，それをサブルーチンとして利用すれば，Q_0 を解く多項式時間アルゴリズムが構成できることを示せばよい．これが示されるとき，Q_0 は Q に**多項式時間帰着**できるという．この場合，オラクルが呼ばれる回数は，入力のサイズの多項式オーダであるので，仮にQを解く多項式時間アルゴリズムがあるとして，それでオラクルを置換えても全体の計算時間は多項式のオーダである．すなわち，Qが易しいとしたら，Q_0 もそうなる．

3.4 暗号の安全性を議論する上での注意

(1) **2.2** で述べたように暗号の解読の問題は，単に'難しい'だけでなく，次に定義する均等に '難しい' ことが要請される．入力集合 I の部分集合 D においてどのサイズ n についても，Dの中のサイズ n の元の数と I の中のサイズ n の元の数との比が ρ 以上であるとき，Dを比率 ρ 以上の部分集合という．すべての $0<\rho\leq 1$ について，比率 ρ 以上のどのような入力部分集合の上に問題を制限しても，問題が '難しい' ならば，もとの問題は**均等に** '難しい' という．同様に，均等に難しいも定義する．

2.1 の Q_3 では，偶数の入力に対して，2が解となるから，比率 0.5 以上の1つの入力部分集合 (偶数入力) に対して，問題は極めて易しい．一方 **5.** で説明するように，入力全体に対しては '難しい' 問題である．

(2) 暗号解読者が利用できる情報としてどのようなものを前提とするかによって，解読の困難さがまったく異なる可能性がある．Lempel らは次のような特徴をもつ対称鍵方式の暗号例を示した[13]．

1) 暗号解読者が自分の選んだ平文mとそれに対する暗号文cを知っている(1つの平文に対する chosen plaintext attack) として，秘密鍵kを求める問題は，ナップザックの問題 (**7.** と付録(2)参照) そのものである．

2) 一方，解読者がn (秘密鍵のビット長) 個の平文，暗号文の対を知ることができれば(n組の known plaintext attack)，秘密鍵はn次元の連立1次方程式の解として表わされる．その係数行列は，暗号化の時に生成する乱数に依存して，確率 1/3 以上で正則となり，秘密鍵は容易に求まる．平文，暗号文の対の数を少しふやせば，正則となる確率が1に近づく．

(3) 次節の Williams の暗号系参照．

4. RSA 暗号系及びその変形の安全性

(1) RSA 暗号において，公開されている n, k_e (例1参照) から秘密鍵 k_d を求める問題とnの素因数分解とは，相互に多項式時間帰着されるので，同程度に'難しい'[6]．

(2) (1)から RSA 暗号の解読問題は，nの素因数分解問題に多項式時間帰着されるが，逆はまだ示されていない．しかし RSA 暗号系を若干変形した Williams の暗号系[14](nの素因数 p, q について，$p\equiv 3$, $q\equiv 7 \pmod 8$ の条件が付く)では次のことが示されている．任意に小さい正数ρについて，Dを暗号文集合の比率ρ以上の任意の部分集合とする．Dにおいて暗号を解読するオラクルを用いて，nをほとんど確実に因数分解する多項式時間アルゴリズムが構成できる．

すなわち，暗号の解読問題が均等に難しいことと，上述の制限の下での素因数分解問題が難しいこととは等価である．一方，上のアルゴリズムにおいて，オラクルは無作為に選んだ1つの暗号文に対応する平文を求めるためにのみ呼ばれる．オラクルを用いる代わりに，解読者が選んだ暗号文に対して，対応する平文を知ることができたら (chosen ciphertext attack)，多項式時間でnが因数分解され，解読が成功する．暗号の安全性を保証するための証明そのものが，chosen ciphertext attack の方法を示す皮肉な結果となっている．Rabin のディジタル署名法についても同様のことが示されている．暗号化にランダムの要素を導入することによって，この弱点を補うことができる[15]．

(3) 部分情報に関する安全性[16]

RSA 暗号の解読問題は，例えば次の問題のいずれにも多項式時間帰着できる．

1) Bを定数でないh変数 (hは定数) の論理関数とする．与えられた暗号文cに対して，元の平文mの最下位のh桁を m_1, $m_2, \cdots, m_h$ とするとき，$B(m_1, m_2, \cdots, m_h)$ の真偽を判定する．

2) 与えられた暗号文cに対して，もとの平文の最下位 m_1 を誤り率 $1/4-\varepsilon(\varepsilon>0)$ 以下で求める．

これらの結果は，RSA 暗号において，平文の各ビットが保護されていることを示すと共に，逆に暗号文の一部のビットをもらすおそれのあるプロトコルの中では，RSA 暗号が破られるかも知れないことを示している．

5. 整数の素因数分解の問題

与えられた正整数nが素数か否かの判定は，**例3**で示したように易しい問題であるが，nの素因数を求める問題 (**3.1** の Q_3) は，永年にわたる問題であるにも拘らず，能率のよい方法は知られていない．現在，Morrison-Brillhart の方法 10)，(Schroeppel の変形)，Pomerance の方法 17) などが知られているが，所要計算時間の平均値は次式のオーダである．

$$\exp(\alpha\sqrt{\log n \log\log n}),\ \alpha \text{ は正定数} \qquad (1)$$

nを2つ以上の異なる素数で割れる奇数と仮定する．次の条件を満たすnと素な整数 x, y が存在することが知られている．

$$x^2\equiv y^2,\ x\not\equiv y,\ x\not\equiv -y \pmod n \qquad (2)$$

このような x, y が見つかると，nの真の因数がnと $x+y$ の最大公約数として求まる．問題は x, y の探し方である．上述のいずれの方法でも，大略次のような方針を採用している．

vを正数とし，すべての素因数がv以下である整数を ***v*-smooth** と呼ぶことにする．v以下の素数を p_1, $p_2, \cdots, p_h$ とすると，v-smooth なzは，$p_1^{d_1}p_2^{d_2}\cdots p_h^{d_h}$ と表わされる．$(d_1, d_2, \cdots, d_h)$ をzの指数ベクトルと呼ぶ．正整数zをnで割った余りを (z) と書く．適当なvを選び，次のようにして，(x^2), (y^2)が v-smooth である x, y の中で，(2)を満たすものを探す．

1) 予定の回数 N に達しなければ 2)へ，達すれば停止 (失敗)．

2) $n-1$ 以下の正整数zを何らかの方法で生成し，(z^2) が v-smooth でなければ，1)へ，v-smooth なら 3)へ．

3) (z^2) の指数ベクトル (e_z と書く) が，すでに記憶されている指数ベクトルの組に対して，mod 2で考えて1次独立なら，z と e_z を記憶する．もし，1

次従属で，$e_z \equiv e_{z_1}+e_{z_2}+\cdots+e_{z_m} \pmod 2$ と表わされれば，

$x \leftarrow z z_1 z_2 \cdots,\ z_m,\ y \leftarrow p_1^{d_1} p_2^{d_2} \cdots p_h^{d_h}$

(ただし，$(d_1, d_2, \cdots, d_h)=(e_z+e_{z_1}+e_{z_2}+\cdots+e_{z_m})/2$)

とおき $(x^2 \equiv y^2 \pmod n)$ が成立する)，$x \equiv y$ または $-y \pmod n$ なら，1)へもどり，そうでなければ，n と $x+y$ の最大公約数 (n の真の約数) を出力し停止(成功)．

$v=\exp\{2lnn\,ln\,ln\,n)^{1/2}\}$，$N=v^2+1$ とおくと，2)で $n-1$ 以下の整数 z を無作為に生成するとき，計算時間の平均値が，$\exp\{3(2lnn\,ln\,lnn)^{1/2}\}$ のオーダ，失敗する確率が $N^{-1/2}$ のオーダである[18]．各方法の違いは，z の発生についてのヒューリスティックな工夫にあり，(1)式の α の値が異なる[17]．

文献 19)によれば，(i)Morrison-Brillhart の方法を IBM-370/158 で実行したところ，n が 10 進 54 桁の例で数日，この程度の大きさの n に対しては，計算時間は $n^{1/8}$ のオーダであり，(ii)Pomerance の方法を CRAY-1 で実行したとき，10 進 55 桁の難しい例で，6時間以下，この程度の大きさの n について，計算時間は $n^{1/10}$ のオーダであった．Pomerance は，10 進 100 桁の素因数分解が，数台のスーパコンピュータとか，特殊目的ハードウェアなどを用いることによって月単位の時間で実行できるのではないかと推定している．RSA 暗号系が提案された当初，10 進 100 桁の素因数分解に 100 年を要すると言われていたのに比べ，数年後にこのような推定が専門家から出ていることは興味をひく．

6. 離散的対数問題

例 2 で述べた Diffie-Hellman の分配鍵方式をはじめいくつかの方式は，次の離散的対数問題の '難しさ' に依存している．

離散的対数問題：q をある素数のベキ乗，q を有限体 GF(q) の原始元とする．零でない任意の GF(q) の元 u が与えられたとき，

$u=g^i,\ 0 \le i < q-1$

を満たす整数 i (ちょうど1つ存在する．Log(u) と略記する) を求めよ．

離散的対数を求める Merkle 及び Adleman の確率的アルゴリズム[20),21)]の概略を説明する．簡単のため，q を素数とする．

1) $q-1$ 以下の正整数のうち，離散的対数を求めるのが容易な整数の占める割合を ρ とする．与えられた整数 $0<u<q$ が，容易な整数なら (容易か否かの判定は易しいと仮定する)，対数 Log(u) を求める．もしそうでなければ，無作為に $q-2$ 以下の正整数 r を選び，$R \equiv u g^r \pmod q$，$0<R<q$ を求める．R が容易な整数なら，Log(R) を求めると，

$\mathrm{Log}(u) \equiv \mathrm{Log}(R) - r \pmod{(q-1)}$

Rが容易な整数でなければ，上記の乱数発生以降の操作を繰り返す．平均 $1/\rho$ 回で成功する．

2) 正数 v を適当に選び，v-smooth な正整数を 1)における容易な整数として採用し，v 以下の素数を $p_1, p_2, \cdots, p_h$ とする．u が v-smooth なら，その指数ベクトルを $(d_1, d_2, \cdots, d_h)$ とすると，

$$\mathrm{Log}(u) \equiv \sum_{j=1}^{h} d_j\ \mathrm{Log}(p_j) \pmod{(q-1)}$$

3) $\mathrm{Log}(p_j)\ (1 \le j \le h)$ を次のように求めることができる．区間 $[0,\ q-2]$ から無作為に整数 r を選び，$R \equiv g^r \pmod q$，$0<R<q-1$ が v-smooth なら，Log(R) と e_R を記憶し，mod $(q-1)$ で考えて，h 個の1次独立な指数ベクトルが得られるまで繰り返す．得られた h 個の離散的対数と指数ベクトルをそれぞれ，$r_i, (d_{i1}, d_{i2}, \cdots, d_{ih})\ (1 \le i \le h)$ とすると，

$$\sum_{j=1}^{h} d_{ij}\ \mathrm{Log}(p_j) \equiv r_i \pmod{(q-1)},\quad 1 \le i \le h$$

から，$\mathrm{Log}(p_j)\ (1 \le j \le h)$ が求まる．

以上の実行には，まず 3)で予め，$\mathrm{Log}(p_i)\ (1 \le i \le h)$を計算しておき，$u$が任意に与えられたとき，1)と2)で Log($u$) を求める．$v=\exp(\alpha\sqrt{\log q \log\log q})$ (α：定数) と選び，3)の計算を若干変更すると，全体の計算時間の平均は

$$\exp(\beta\sqrt{\log q \log\log q}),\ \beta\text{：定数} \qquad (3)$$

以下となる[21)]．

上記の方法は，一般の有限体の場合に拡張され，特に $q=2^h$ の場合*，一般の場合より著しく改善されることが示されている．GF(2^h) の元は，$h-1$ 次以下の GF(2)の上の多項式で表わされる．素因数の次数がすべて v 以下であるとき，v-smooth と呼ぶことにする．Blake, Fuji-Hara らは拡張ユークリッドアルゴリズムの利用，及び GF(2^h) では $(a+b)^2=a^2+b^2$ であることを利用して，(3)式の定数 β を小さくできることを示した[22)]．Coopersmith は，さらに改善して，平均計算時間が

$\exp(\gamma h^{1/3} \log^{2/3} h),\ \gamma$：定数

のオーダでおさえられることを示した[23)]．$h=127$ に

* 一般に，$q=p^n$ で，p が大きくないときにも適用される．

対し，次数12以下の既約多項式の Log 表が，IBM 3081 model K の CPU 時間で1時間足らずで作成された[28]．GF(2^{127}) を利用した Diffie-Hellman の分配鍵方式の開発計画が中止されたと伝えられる．qとして素数でなく，2のベキ乗を用いる理由は，暗号化，復号化の機構化の容易さであるが，安全性の上からは望ましくなく，同じレベルの安全を保証するためには，hを$\log_2 q$に比べて相当大きくとる必要があろう．

7. ナップザック問題に基づく非対称鍵暗号系の解読

Merkle-Hellman の非対称鍵暗号系[24]（MH 暗号系と略称）では，

1) 鍵の作成：秘密鍵と公開鍵を次のように作る．まず正整数 $b_1, \cdots, b_n$ を次のS条件を満たすように選ぶ．

$$S\text{条件}: b_i > \sum_{j=1}^{i-1} b_j, \quad 1 < i \leq n$$

次に，$\sum_{j=1}^{n} b_j$ より大きい整数 M_o とそれに素な正整数 W_o を選び，

$$a_i \equiv W_o b_i \pmod{M_o}, \quad 1 \leq a_i < M_o, \quad 1 \leq i \leq n \quad (4)$$

とおき，$a_1, a_2, \cdots, a_n$ を公開鍵とし，$b_1, b_2, \cdots, b_n$, W_o, M_o を秘密鍵とする．

2) 暗号化：平文はnビットで，平文 $(m_1, m_2, \cdots, m_n)$ に対する暗号文は整数 $\sum_{i=1}^{n} a_i m_i$ である．

3) 復号化：暗号文 $c = \sum_{i=1}^{n} a_i m_i$ に対し，$c' \equiv W_o^{-1} c \pmod{M_o}$, $0 \leq c < M_o$（ただし，$W_o^{-1} W_o \equiv 1 \pmod{M_o}$）を求めると，(4)式から，$c' = \sum_{i=1}^{n} b_i m_i$．$S$条件から，$c' > b_n$ なら，$m_n = 1$，そうでなければ，$m_n = 0$，$c' - b_n m_n > b_{n-1}$ なら，$m_{n-1} = 1$，そうでなければ，$m_{n-1} = 0$，以下同様に，$m_{n-2}, \cdots, m_1$ が求まる．

MH 暗号系の解読は，正整数の組 $a_1, a_2, \cdots, a_n$ と整数 c が与えられて，$c = \sum_{i=1}^{n} a_i m_i$（ただし，$m_i$ は0か1）を満たす $m_1, m_2, \cdots, m_n$ を求めることである（解の存在は保証されている）．$a_1, a_2, \cdots, a_n$ に制限がつかなければ，ナップザック問題（付録(2)）である．しかし，MH 暗号系では，$a_1, \cdots, a_n$ は，S条件という厳しい条件を満たす $b_1, \cdots, b_n$ から(4)式で求められたものであり，この弱点を利用して，1982年に Shamir は，解読法を示した[25]．ほとんどすべての公開鍵 $a_1, a_2, \cdots, a_n$ に対して，nの多項式時間内に，$Wa_1 \pmod{M}$, $Wa_2 \pmod{M}, \cdots, Wa_n \pmod{M}$ がS条件を満たすような W, M を与える決定性アルゴリズムを示した．前述の 3)において，W_o^{-1}, M_o, $b_1, \cdots, b_n$ の代りに，$W, M, Wa_1 \pmod{M}, \cdots, Wa_n \pmod{M}$ を使えば，もとの平文が求まる．

Shamir の方法では，各 a_i に対して，関数 $f_i(z) = a_i z - \lfloor a_i z \rfloor$, $0 \leq z < 1$ を考える（$\lfloor x \rfloor$ はxを超えない最大整数）．この関数は，(4)式の b_i から a_i を求める関数の逆関数を，両方の座標軸を M_o で割ることによって正規化して得られる関数である．$f_i(W_o^{-1}/M_o) = b_i/M_o$ であるから，$f_1(W_o^{-1}/M_o)$, $f_2(W_o^{-1}/M_o), \cdots, f_n(W_o^{-1}/M_o)$ は，S条件を満たす．$Z = W_o^{-1}/M_o$ の近傍に，関数 $f_1, f_2, \cdots, f_n$ の不連続点（関数値が1から0に変わる点）が集中することが示される．このような集中点を Lenstra の整数線形計画問題を解くアルゴリズムを利用して求める．

MH 暗号系の変形版である繰り返し MH 暗号系，Graham-Shamir の暗号系などについても解読法が提案されている[26]．

8. あ と が き

紙数の関係もあって，プロトコルの安全性の問題，擬似乱数系列，計算量を考慮した情報理論の再構成などを省略した．文献は直接参照したものに限ったので，文献 27)の文献リストを参照されたい．

有益なご助言をいただいた，本学谷口健一助教授に謝意を表します．

参 考 文 献

1) 五十嵐善英：確率的アルゴリズムの概観，情報処理，Vol. 21, No. 1, pp. 13-17(1980).
2) Rabin, M.O.: PROBABILISTIC ALGORITHMS, Algorithms and Complexity, J.F. Traub (ed.), pp. 21-39, Academic Press(1976).
3) Welsh, D.J. A.: RANDOMISED ALGORITHMS, Discrete Applied Mathematics, Vol. 5, No. 1, pp. 133-145 (1983).
4) Diffie, W. and Hellman, M.E.: New Directions in Cryptography, IEEE Trans. on IT, Vol. IT-22, No. 6, pp. 644-654 (1976).
5) Denning, D.E. R.: Cryptography and Data Security, Addison-Wesley Publishing Company (1982).
6) Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, CACM, Vol. 21, No. 2, pp. 120-126 (1978).
7) 小山謙二：ディジタル署名と暗号鍵管理，本特集.

8) Morrison, M. A. and Brillhart, J.: A Method of Factoring and the Factorization of F_7, MATHEMATICS OF COMPUTATION, Vol. 29, No. 129, pp. 183-205 (1975).
9) Garey, M. R. and Johnson, D. S.: Computers and Intractability--a Guide to the Theory of NP- Completeness, W. H. Freeman and Co. (1979).
10) ポメランス, C.: 素数を求めて, サイエンス, 1982年12月, pp. 102-112 (1982).
11) Brassard, G.: A Note on the Complexity of Cryptography, IEEE Trans. on IT, Vol. IT-25, No. 2, pp. 232-233 (1979).
12) 嵩, 山村: 公開鍵暗号系の安全性保証の難しさ, 京大数解研講究録, 381, pp. 52-72 (1980).
13) Lempel, A.: Cryptology in Transition, ACM Computing Surveys, Vol. 11, No. 4, pp. 285-303 (1979).
14) Williams, H. C.: A Modification of the RSA Public-Key Encryption Procedure, IEEE Trans. on IT, Vol. IT-26, No. 6, pp. 726-729 (1980).
15) Rivest, R. L. and Sherman, A. T.: RANDOMIZED ENCRYPTION TECHNIQUES, Advances in Cryptology, pp. 145-163, Plenum Press (1983).
16) Ben-Or, M., Chor, B. and Shamir, A.: On the Cryptographic Security of Single RSA Bits, Proc.of 15 th STOC, pp. 421-430 (1983).
17) Pomerance, C.: Analysis and Comparison of some Integer Factoring Algorithms, Computational Method in Number Theory, Lenstra Jr., H. W. and Tijdeman, R.(eds) pp. 89-139, Math. Centrum, Amsterdam (1983).
18) Dixon, J. D.: Asymptotically Fast Factorization of Integers, MATHEMATICS OF COMPUTATION, Vol. 36, No. 153, pp. 255-260 (1981).
19) Pomerance, C.: ON FACTORING LARGE NUMBERS OR HOW STRONG IS THE RSA CRYPTOSYSTEM?, ABSTRACTS OF PRESENTATIONS AT 1983 IEEE INFORMATION THEORY WORKSHOP on Multi-User Information Theory and Systems, pp. 27-28 (1983).
20) Hellman, M. E. and Reyneri, J. M.: FAST COMPUTATION OF DISCRETE LOGARITHMS IN GF(q), Advances in Cryptology, pp. 3-13, Plenum Press (1983).
21) Adleman, L.: A SUBEXPONENTIAL ALGORITHM FOR THE DISCRETE LOGARITHM PROBLEM WITH APPLICATIONS TO CRYPTOGRAPHY, Proc. of 20th FOCS, pp. 55-60 (1979).
22) Blake, I. F., Fuji-Hara, R., Mullin, R. C. and Vanstone, S. A.: Finite Field Techniques for Shift Registers with Application to Ranging Problems and Cryptography, Final Report, Project No. 106-16-02, Dept. of Communication Univ. of Waterloo, Canada (1983).
23) Coppersmith, D.: FAST EVALUATION OF LOGARITHMS IN FIELDS OF CHARACTERISTCI TWO, IBM Research Report RC 10187 (1983).
24) Merkle, R. C. and Hellman, M. E.: Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. on IT Vol. IT-24, No. 5, pp. 525-530 (1978).
25) Shamir, A.: A POLYNOMIAL TIME ALGORITHM FOR BREAKING THE BASIC MERKLE-HELLMAN CRYPTOSYSTEM. Proc. of 23rd FOCS, pp. 145-152 (1982).
26) Adleman, L. M.: On Breaking Generalized Knapsack Public Key Cryptosystems, Proc. of 15th STOC, pp. 402-412 (1983).
27) 嵩, 山村: 符号・暗号における最近の進歩, 電気四学会連合大会資料, pp. 5-125-5-128 (1983).

付録: 問題のクラス RP, NP, CONP

(1) 問題として, 各入力に対し正解が一意で, 真と偽のいずれかである問題 (判定問題と呼ばれる) を考える. 正解が真 (または偽) であるような入力全体の集合を D_T (または D_F) と書く. 入力部分集合 D_T (または D_F) に対して, 1より小さい定数以下の失敗率, 0の誤り率で解く, 多項式時間確率的アルゴリズムが存在する問題のクラスを *RP* (または *co RP*) と呼ぶ[3]. また, D_T (または D_F) に対して, (入力のサイズに依存してよいが) 1に等しくない失敗率, 0の誤り率で解く, 多項式時間確率的アルゴリズムが存在する問題のクラスを *NP* (または ***coNP***) と呼ぶ. 定義から, RP⊆NP, coRP⊆coNP が成立する. これらの式で真の不等号が成立すると予想されているが証明されていない. : **例3** のアルゴリズムにおいて, 出力 'Yes' を出す所を何も出力しないように変更すると, 上の coRP の条件を満たすから, 素数判定問題は coRP に属する. また NP に属することも示されている[9].

(2) クラス C の問題に多項式時間帰着されるすべての問題から成るクラスを C^* と書く. Q がクラス C に属し, C のすべての問題が Q に多項式時間帰着できるとき, Q を ***C*-完全**と呼ぶ. **7** で述べたナップザックの問題は NP*-完全である. 整数の素因数分解問題 Q_3, 離散的対数問題は $(NP \cap coNP)^*$ に属する. 例えば, RSA 暗号系は, 素数判定問題$\in NP \cap coRP$ の易しさと, $Q_3 \in (NP \cap coNP)^*$ の '難しさ' の '差' に依存している.

(昭和59年2月16日受付)