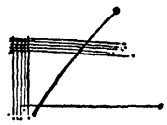
展　望



公衆暗号系の実現可能性と問題点†

土居範久††　広瀬　健†††
一松　信††††　西村和夫†††††

1. はじめに

本稿では解説「公衆暗号系」のあとを受けて，その実現可能性と問題点を中心に，公衆暗号系を展望する．

すなわち，公開鍵方式で用いられる算法の数学的基礎の概略を示すことにより，それらの算法の状況と新しい算法についての可能性を認識し，とくに Rivest-Shamir-Adleman による RSA 法や Merkle-Hellman による MH 法については，暗号化，復号，解読の手間について述べる．さらに，これにもとづいて公開鍵暗号系の実現性とその問題点を論じ，慣用の暗号系と公開鍵暗号系との実現上の問題を対比する．

以上の議論は，解説「公衆暗号系」で述べられた"常識"を仮定してはいるが，一応，独立して読めるように配慮したつもりである．

なお，暗号学の諸文献については，ここでは最後に必要最小限の引用をするにとどめたが，たとえば，著者達を含めたグループで調査報告した「市場創出型プロジェクト――"公開鍵方式暗号系"に関する調査研究報告書」（協同システム開発株式会社）などに詳しい．

2. 公開鍵暗号系における算法

ここでは，公開鍵暗号系で用いられる手続き，とくに RSA 法(Rivest-Shamir-Adleman の算法)[1]，MH 法 (Merkle-Hellman の算法)[2]，R 法 (Rabin の算法)[3]を中心とした展望を試みたい．

2.1 "一方通行関数"と"落し戸関数"

「解説：公衆暗号系」で述べたことであるが，公開鍵暗号方式では，次のような暗号化手続き E，復号手続き D の存在が必要である[4]．

（i） すべての平文 M に対して，$E(M), D(E(M))$ が存在して，
$$D(E(M))=M$$
が成立する．

（ii） E，D の手続きが比較的簡単である．

（iii） E から D を割り出すことが，実際上，不可能である．

そして，"署名"(**計数署名** (digital signature) という）を可能にするためには，さらに

（iv） すべての平文 M に対して，$D(M), E(D(M))$ が存在して，
$$E(D(M))=M$$
が成立する．

さて，（i）から明らかなように，D は E の逆変換である．（ii）によれば，E，D の手続きは容易に実行できるものでなくてはならない．暗号化および復号の手続きが複雑すぎては実用にならない，というわけである．ここで（iv）の条件が問題になる．暗号化の手続きが，復号のための手続きから割り出されるのは困る．E も E の逆変換 D も簡単に実行できるが，E から簡単な手続きとしての D が割りだされないようにしたい．

そこで，E の手続きは容易で，E の逆変換 E^{-1} の手続きは一般に莫大な手間を要する，しかし，ある"しかけ"を知っていれば，E^{-1} の手続きも容易に実行できる，というものを考える．つまり，"しかけ"を知った上での E^{-1} の手続きを D にしようというわけである．

関数 f の計算量は少なく，f^{-1} の計算量が莫大になるような関数を**一方通行関数** (one-way function) という．これは数学上の術語ではなく，暗号学におけ

---

† Feasibilities and Some Problems of the Public Cryptosystem by Norihisa DOI (Institute of Information Science, Keio University), Ken HIROSE (Dept. of Mathematics, School of Science and Engineering, Waseda University), Sin HITOTUMATU (Research Institute for Mathematical Sciences, Kyoto University) and Kazuo NISHIMURA (Dept. of Mathematics, Faculty of Engineering, Keio University).

†† 慶応義塾大学情報科学研究所
††† 早稲田大学理工学部数学科
†††† 京都大学数理解析研究所
††††† 慶応義塾大学工学部数理工学科

る方言である．逆関数 f^{-1} の意味も，$f(x)$ の値から x を対応させる手続きを表わす，といったほどの意味に了解されたい．

このような，f に比較すると f^{-1} の計算量がずっと大きくなる，という手続きの例は数多く存在する．たとえば n 個の数 $x_1, x_2, \cdots, x_n$ と，n^2 個の数 $a_{i1}, a_{i2}, \cdots, a_{in}$ $(i=1,2,\cdots,n)$ から，n 個の1次式

$$a_{i1}x_1+a_{i2}x_2+\cdots+a_{in}x_n$$

の値 b_i $(i=1,2,\cdots,n)$ を求めるには，n^2回の乗法と $n(n-1)$ 回の加法ですむが，この逆の手続き，つまり，n^2 個の数 $a_{i1}, a_{i2}, \cdots, a_{in}$ と n 個の数 $b_i(i=1,2,\cdots,n)$ から，

$$\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n \end{cases}$$

を満たす $x_1, x_2, \cdots, x_n$ を求めるには，$n^3/3$ 回程度の演算を必要とすることはよく知られている．

この場合は，f が $O(n^2)$，f^{-1} が $O(n^3)$ というわけで，大した差ではないが，f のオーダが n の多項式，f^{-1} のオーダが n の指数関数，ということになれば，n を適当に大きくすれば，f は計算可能，f^{-1} は実際上計算不可能ということになる．

このような一方通行関数 f の計算で簡単に暗号文を作り，f^{-1} の計算量の大きさを利用して解読を防止したいわけだが，このままでは復号もままならぬから，**落し戸** (trapdoor) と呼ばれる“しかけ”をもうける．

一方通行関数であって，ある情報(“しかけ”)を知っていさえすれば f^{-1} も簡単に計算できるような関数 f を，**落し戸関数** (trapdoor function) という．正規の受信者は，この“しかけ”，つまり，復号の鍵を知ることにより f^{-1} を簡単に計算し，復号することができる．

このような“落し戸関数”を提示したのが，RSA 法であり MH 法であり R 法だったのである．

2.2 RSA 法の算法について

素数でないことは分かっているが，その素因数は具体的に知られていない，という実例が多数ある．

素数の判定をする算法としては，その数の桁数の定数倍程度の計算量のものが知られている．これに対して，素因数分解を行うには，現在最も能率のよいといわれる算法でも，整数 x に対し

$$\exp((ln(x)\cdot ln(ln(x)))^{1/2})$$

程度の演算が必要になる．

“一方通行関数”といわれて，数学者が一番思いつきやすい例の一つがこの素因数分解であろう．具体的方法として最初に提示されたのが RSA 法であったのも当然な気がする．

RSA 法の算法は「解説」でも述べたが，以下の議論の都合上再録する．そしてここでは，RSA 法の算法が上記(i)～(iv)を満たすことを示そう[1]．

まず，平文は符号化されて自然数列になっているものとする．二つの素数 p, q をとり，$n=p\cdot q$ とおく．平文を表わす自然数列を適当に区切って，n より小さい整数 M の列とみなす．

$(p-1)(q-1)$ と互いに素な数 r,

$d\cdot r\equiv 1 \pmod{(p-1)(q-1)}$ なる数 d

をとる．

暗号化のための公開鍵は，n と r であり，復号のための秘密の鍵が d である．

暗号化手続き $E: M\mapsto C$ は，

$$C\equiv M^r \pmod n$$

なる暗号文 C を作ることである．

復号手続き $D: C\mapsto M$ は，

$$M\equiv C^d \pmod n$$

によって得られる．

(i)～(iv)を示すために必要な定義と定理を挙げる．

定義1 m を法としての既約剰余類の個数を $\varphi(m)$ と表わし，φ を Euler の関数という．

定理1 (1) p が素数ならば，$\varphi(p)=p-1$.

(2) $(x,y)=1$ ならば，$\varphi(x\cdot y)=\varphi(x)\cdot\varphi(y)$.

定理2 (Euler の定理)

$(a,m)=1$ なる任意の整数 a について，

$$a^{\varphi(m)}\equiv 1 \pmod m.$$

系 (Fermat の小定理)

素数 p と $(a,p)=1$ なる任意の整数 a について，

$$a^{p-1}\equiv 1 \pmod p.$$

さて，(i)と(iv)を示す：

定理1の(1)，(2)から，$n=p\cdot q$ に対し，

$$\varphi(n)=\varphi(p)\cdot\varphi(q)=(p-1)(q-1)$$

したがって，r, d の定義から，r と $\varphi(n)$ は互いに素で，

$$d\cdot r\equiv 1 \pmod{\varphi(n)} \qquad (*)$$

ところで，

$$D(E(M))\equiv(E(M))^d\equiv(M^r)^d\equiv M^{rd} \pmod n$$

$E(D(M))\equiv(D(M))^r\equiv(M^d)^r\equiv M^{dr} \pmod n$

であるから，

$M^{r\cdot d}\equiv M \pmod n$　　(＊＊)

を示せば，$0\leq M<n$ なるすべての M について

$D(E(M))=M$, $E(D(M))=M$

が成立し，(ⅰ)，(ⅳ)が示されたことになる．

そこで (＊) から (＊＊) を導こう．(＊)から，ある k に対して $d\cdot r-1=k\cdot\varphi(n)$ だから

$M^{d\cdot r}\equiv M^{k\cdot\varphi(n)+1} \pmod n$　　(♯)

である．次に，$(M,p)=1$ なる M に対しては，Fermat の小定理から，$M^{p-1}\equiv 1 \pmod p$ で，$\varphi(n)=(p-1)(q-1)$ であるから，$k'=k\cdot(q-1)$ とおけば，

$M^{k\cdot\varphi(n)+1}\equiv M^{k'\cdot(p-1)+1}\equiv M \pmod p$.

$(M,p)\neq 1$ なる M，すなわち $M\equiv 0 \pmod p$ なる M について上式が成立することは明らかゆえ，任意の M について

$M^{k\cdot\varphi(n)+1}\equiv M \pmod p$

が成立する．qについても同様にして

$M^{k\cdot\varphi(n)+1}\equiv M \pmod q$

であり，p, q は素数であるから

$M^{k\cdot\varphi(n)+1}\equiv M \pmod{p\cdot q}$.

ここで $n=p\cdot q$ と (♯) から

$M^{d\cdot r}\equiv M \pmod n$

つまり (＊＊) が成立する．

(ⅱ) の E, D が比較的簡単であることについては，次章を参照されたい．E は高々 $2\log_2 r$ 回，D も高高 $2\log_2 d$ 回の乗算で手続きが完了する．

最後に (ⅲ) の E から D が割り出せないこと，つまり，復号鍵 d の解読には莫大な計算量をともなうことについて述べる．

公開の暗号化鍵 n と r によって，解読者は，原理的には，次のようにして復号鍵 d を求めることができる：

(1) n を素因数分解し，$n=p\cdot q$ なる p,q を求める．

(2) $u\cdot r\equiv 1 \pmod{p-1}$, $v\cdot r\equiv 1 \pmod{q-1}$ なる u, v を求める (r が $(p-1)(q-1)$ と互いに素であることに注意)．

(3) (2)から，$u-v$ は $p-1$, $q-1$ の最大公約数の倍数になるから，

$u-v=k(p-1)+l(q-1)$

なる k, l が存在する．これを求める．

(4) $d=u-k(p-1)$　$(=v+l(q-1))$

とおけば，これが求める秘密の鍵 d である．

この手続きにもあるように，解読には (1) の素因数分解ないしは，それと同等な手続きが含まれる．(2)，(3)は p,q の桁数の対数程度の計算量だが，(1)は最初に述べたように莫大な計算量を要することになる．

ただし，素因数分解ないしはそれと同等の手続きを含むとはいえ，素因数分解のための新しい算法が発見されて，計算量が少なくなる，という事態が生じないとはいえない．これまでの歴史的状況からして，計算量が断然少なくなるような算法はないだろう，というだけのことである．

したがって，これは (ⅲ) の厳密な意味での証明ではない．なお，この算法の計算量に関わる話題は，次章以下で取扱う．

2.3 MH 法，R 法の算法について

実際上は計算不可能と思われる一群の問題に，NP 完全問題がある．MH 法の算法は NP 完全問題の一つである“ナップサック問題”を用いるものである[2]．この算法と，それが前節の (ⅰ)，(ⅱ)，(ⅲ) を満たすこと，また(ⅳ)を満たさないことについては「解説」で述べた．この算法は，公開鍵の量が多いこと，かなり大きな数が通信に使われることが欠点であるが，その反面，計算は加算，減算と大小の比較ぐらいなものですみ，早い変換が可能と思われる．

そのかわり，解読される可能性も RSA 法などより高い．変換を数回実行することが望ましいとも言われている．

この方法について興味深いのは，やはり NP 完全問題などとの関連といった面であろう．“素因数分解”のような恰好な材料が，ほかにすぐ見つかるとは思えないから，新しい方式，算法の発見は，やはり計算量の理論 (Computational complexity) の分野から得られる可能性が高いであろうと思われる．

最後に M. O. Rabin の算法，R 法に簡単にふれよう[3]．

平文は符号化されて自然数列になっているものとする．素数 p, q をとり，$n=p\cdot q$ とおく．平文を表わす自然数列を適当に区切って，n より小さい整数 x の列とみなす．

n と，ある定数 b をとって，これを暗号化のための公開鍵とする．素数 p, q が秘密の復号鍵である．

暗号化の手続き $E: x\mapsto c$ は，

$c\equiv x(x+b) \pmod n$

によって x の暗号文 c を作ることである．

復号の手続き $D: c \mapsto x$ は，方程式

$$x^2+bx-c\equiv 0 \pmod{n}$$

を解くことである．

復号のための鍵 p, q を知っていれば，

$$x^2+bx-c\equiv 0 \begin{pmatrix}\bmod p\\ \bmod q\end{pmatrix}$$

を解き，その解 u, v から，中国の剰余定理(Chinese remainder theorem)*を用いて x を求めることができる．

p, q を知らずに解読するとなれば，n を素因数分解することになり，RSA 法と同じく，そこで莫大な計算量を要することになる．

この方式のよいところは，暗号化手続きの簡単さであろう．しかし一つの暗号文について平文が普通には四つできる．意味のあるものは，普通には唯一つであろうが，それは n までの数のうちほぼ 3/4 は無意味であることを意味するから，誤りがあったときの復元に問題があろう．また任意の c に対し x が存在するとは限らないから，"署名" には工夫を要する．

さらに，この方式では，偶然重根になる場合にあたると，容易に解読される，という弱点をもっている．これは Rabin の論文にも明示されていないし，その確率はきわめて小さいが，一応注意を要する点であろう．

3. 暗号化と復号および解読の手間

前章で示したような，公開鍵暗号の各体系における暗号化や復号，解読に要する計算の手間について考えてみることにする．

3.1 RSA 法

まず RSA 法[1]について考えてみよう．RSA 法での暗号化は $C=M^r \pmod{n}$ であった．このべき乗の計算は，単純に考えれば r 回の乗算を要するようにみえるが，よく知られているように，r を2進数に展開することによって，ずっと少ない乗算で済ますことができる[5]．つまり $r=r_0+2r_1+2^2r_2+\cdots+2^kr_k$ $(r_i=0,1)$ と2進数で表現すれば

$$M^r=M^{r_0}\cdot(M^2)^{r_1}\cdot(M^{2^2})^{r_2}\cdots(M^{2^k})^{r_k}$$

であるので，r_i が1の因子だけをかけ合わせればよいことになる．手順としては M を順次2乗していき，必要なものをかけていけばよい（ただし，すべての計算は n を法として行う）．したがって，暗号化に要する乗算の回数は多くとも $2\log_2 r$ である．また，この演算数は M によらず，r によって先験的に定まるものである．

この暗号を実用に耐えるものにするには，後で示すように n を200桁程度の大きな数にしなければならない．ところが現在の計算機はこのような大きな整数の計算に便利なようにはできていないので，この暗号化には1秒程度の時間を要する．

RSA 法における復号の手続き $M=C^d \pmod{n}$ は，計算の手間の点では暗号化の手続きとほとんど同じである（$2\log_2 d$ 以下）．

ところが d を知らずに解読しようとすると，莫大な計算量を要する．というよりも，考案者たちは解読が事実上不可能だと思えるからこそ，この方法が暗号の体系に採用できるとしているのである．さて，RSA 法による暗号文を，公開されている鍵 n と r だけを用いて解読するには n を素因数分解すればよい．$n=p\cdot q$ と素因数分解することができれば，後はその n を公開した正当な受信者がしたのと同様な手順を踏めばよい．つまり，p と q とから $\varphi(n)=(p-1)(q-1)$ を計算し，これを法として r の逆数 d を求めれば，$M=C^d$ として暗号文の解読ができる．

このうち逆数の計算は，ユークリッドの互除法を応用して容易に行うことができる（文献 5）の演習問題 4.5.2.15）．簡単に説明すると，x_0 を法として x_1 の逆数を求めるには次のようにする．$i=2$ から始めて x_{i-2} を x_{i-1} で割った商を t_i，余りを x_i とし，これを繰り返す．このとき $x_i=x_{i-2}-t_ix_{i-1}$ である．この t_i を用いて，$b_0=0$, $b_1=1$ から $b_i=b_{i-2}-t_ib_{i-1}$ を計算していくと，$x_k=1$ となったときの b_k が求める逆数である $(x_1b_k=1 \pmod{x_0})$．$k<2\log_2 x_0$ であるので，逆数の計算量は $\varphi(n)$ の桁数程度で済む．また，前に示したように $M=C^d$ の計算量も d の桁数程度であるので，問題となるのは n の素因数分解である．

素因数分解は桁数が大きいとかなりむずかしい問題になる．現在のところ，R. Schroeppel による一番速い算法（未発表）でも $\exp((ln(n)\cdot ln(ln(n)))^{1/2})$ くらいの演算を要するようである[1]．この値は $n=10^{100}$ のとき 2.3×10^{15}, $n=10^{200}$ のとき 1.2×10^{23} になる．ひとつの演算に 10^{-6} 秒だけかかるとすると，それぞれ 74 年と 3.8×10^{9} 年になるが，RSA 法の考案者たちはこの点から n を200桁ぐらいの数にすることを

---

* $a_1, a_2, \cdots, a_k$ を任意の数とし，$m_1, m_2, \cdots, m_k$ を互いに素な組とする．このとき，$x\equiv a_i \pmod{m_i}$ $(i=1,2,\cdots,k)$ となるような数 x が存在する．証明は，たとえば，文献 5) を参照されたい．

推奨している．

RSA 法による暗号を解読するには，n を素因数分解することが必要なわけではない．$\varphi(n)$ を直接に求めてもよいし，べき乗の逆元 d を直接計算してもよい．このことについては RSA 法の考案者たちも気づいていたが，最近 Rabin が，どんなことをしても結局素因数分解をするのと同等であることを示したそうである（文献 6）による．Rabin は未発表）．したがって，RSA 法は素因数分解が困難である限り安全であろう．

3.2 MH 法

今度は MH 法[2]について考えてみる．MH 法での暗号化は加算をするだけで，非常に簡単である．平文 M を暗号化するには，その 2 進表現で 1 になっているビットに対応する b_i の総和をとればよい．したがって，その演算数は M の 2 進の桁数以下である．

復号をするには，まず暗号文 C に v をかける(mod m) が，これは 1 回だけでよい．それから秘密のナップサック a_i を順次に比較し，引いていく．この演算の回数は a_i の個数だけになる．暗号化，復号ともに RSA 法に比べて処理はだいぶ速い．その大きな差は，乗算と加減算の違いというよりも，いちいち法によって還元しなくてもよい点にある．

MH 法の暗号を解読するうまい方法は，現在のところ知られていない．もし全数検査をしようとすると，その数は 2^N (N は a_i の個数) になる．以上に示した RSA 法と MH 法の暗号化，復号および解読に要する演算数を表-1 にまとめておく．

MH 法の基になっているのはナップサック問題であるが，これは NP 完全問題である．一般に P 問題とは，その問題を解く手間が，問題のパラメタ n の多項式になるものをいう．それに対して NP 問題とは，計算の手間が n の多項式では抑えられないものをいう（しかし，解を確認するのは n の多項式の手間で済む）．さらに NP 完全問題とは，NP 問題の部分集合であり，このうちのどれかが P 問題であることが示されれば，全部の NP 問題が P 問題であることになってしまうということが証明されているものをいう．このようなことは実際に示せそうもないので，NP 完全問題は文字通り完全な NP 問題であると思われているわけである．したがってこの NP 完全問題のひとつであるナップサック問題を，一般的に効率よく解くような方法は，とても存在するとは思えない．

表-1 RSA 法と MH 法の手間

	RSA 法	MH 法
暗号化	$2\log_2 r$ 回以下の乗算	N 回以下の加算
復　号	$2\log_2 d$ 回以下の乗算	N 回以下の減算
解　読	$\exp((ln(n)\cdot ln(ln(n)))^{1/2})$	2^N 回以下の加算

4. 公開鍵暗号系の実現性とその問題点

上記のような公開鍵暗号系は，創案されてから 4 年ほどしかたっていず，まだ研究段階にあるが，その問題点についてはいろいろと議論されている．それらを列挙しておこう．

4.1 一方通行関数の強度

まず，どの方法についてもいえるのは，使用する一方通行関数の強度についてである．どの一方通行関数についても，秘密の鍵を知らずに逆変換をしようとすると，その作業は（現在のところ）困難を極める．しかしながら，どんな手段を尽しても，その計算量が将来に渡って画期的に減少することがないというような証明はない．たとえば素因数分解をする速い算法はまだ知られていないが，その存在を否定するような証明はまだない．

4.2 "NP 完全性" の意味

MH 法についていえば，解読のむずかしさを NP 完全性に委ねているが，この危険性については Lempel が指摘している[6]．まず，ナップサック問題は NP 完全問題であるが，そのナップサック (a_i) に落し戸 ($a_1+\cdots+a_{i-1}<a_i$ という特別な条件) を仕組んでもなお NP 完全であることは証明されていない．次に NP 完全問題とは，問題のパラメタ(たとえば暗号文)が最悪の場合について，手間が多項式で抑えられないものをいうのであるから，常にむずかしいとは限らないのである．極端にいえば，NP 問題による暗号文を解読しようとすると，たまに解けないことがあるが，大抵は簡単に解けるというようなこともあり得るわけである．さらに最悪の場合にも，どれだけむずかしいかはまだ明らかになっていない．Lempel は NP 完全問題に基づきながら簡単に解読できる暗号を例示して，この点に関して注意を促している[6]．

4.3 署名について

また MH 法は，そのままでは署名をすることができない．つまり値域内の任意の暗号文 C に対して，それに対応する原文 M が存在するとは限らないのである．この欠点をなくすためには，ナップサック (b_i) を変えて一方通行関数の値域の密度を上げればよいが，そうすると問題のむずかしさが減少し，通信の秘

密が保てなくなってしまう．これは線形な一方通行関数について一般的にいえることである．この点，RSA法は非線形であるために，署名が可能で，逆変換も存在するが，その計算には多大の手間を要する．

4.4 実用上の問題点

しかし，RSA法は暗号化や復号の手間と，解読に要する手間との差が十分に大きいとはいえない．そのために，どんな高速の計算機を使っても現在の技術では解読できないような，十分に長い桁数の法を鍵として用いると，その暗号化や復号にも多くの時間を要する．これは実用上の難点である．数百桁の整数を法のもとで瞬時に乗算する専用のプロセッサが必要となるだろう．

実用面を考えると鍵の長さや数も問題になる．長大桁の数をどうして記憶あるいは記録し管理したらよいだろうか．たとえば，RSA法の秘密の鍵は d だけであるが，この長い（数百）桁の数を記憶し，受信のたびに受信器に入力するのは大変である．Denningは ROM(受信器内部のチップや磁気カードなど）に記録しておくことを提案している[7]．MH法の場合はもっと状態が悪い．秘密の鍵は v とベクトル (a_i) であるが，a_i は長いもので百桁ぐらいの数であって，その総数も百から数百個に及ぶことが予想される．また公開の鍵についても，その記録や管理，呼出しの方法などに関して検討を要する．

そのような鍵を一般の加入者がどうやって決定したらよいかも未解決の問題である．鍵の決定は秘密を要するので，加入者がそれぞれ自分のところで行うしかないだろうが，そのプログラムはセンターのものを共用するとしたら，全加入者が納得するだろうか．また素数などの偏りを防ぐにはどうしたらよいだろうか．

以上のようなことを考えていくと，便利そうにみえる公開鍵暗号系も，その実用面には多くの解決しなければならない問題があることが分かる．また公開鍵暗号が社会に与える影響も調査しなければならないだろう．

4.5 RSA法，MH法，R法の比較

RSA法とMH法について，いままで述べてきたことをまとめて表-2に示しておく．表-2には第2章で説明したR法[3]も，比較の対象として加えておいた．表中の暗号化と復号の時間の項には，現在の高速な計算機上で，実用的に妥当な桁数の数（平文や暗号文）を対象としたときに，かかるとみなされる時間が記入してある．したがって，これらは正確な値ではなく，

表-2 RSA法，MH法，R法の比較

	RSA法	MH法	R法
暗号化の時間	数秒	1秒以下	数ミリ秒
復号の時間	数秒	1秒以下	数秒から数分
解読される可能性	きわめて小	ありうる	きわめて小（弱点あり）
N文字の平文からできる暗号文の長さ	2N桁	$\frac{5}{3}$N 桁以上	2N桁以上
平文と暗号文の対応	1対1	1対1（ただし任意の暗号文が平文にならない）	4対1
計数署名	容易	不可能	可能（工夫を要する）
公開鍵の個数	r, n の2個	数百個の b_i	b, n の2個
秘密の鍵の個数	d 1個	m, v の2個（と a_i）	p, q と定数を数個
実用化の問題点	暗号化と復号が遅い	公開鍵の数が多い	復号が遅い，重根になったとき解読可能

比較の目やすのためのものである．

MH法が解読される可能性について，少し述べておこう．MH法が基にしているナップサック問題は，形式的に線型計画問題になるが，この線型計画問題について，1979年にソ連のKhachianが，パラメタの数の多項式の手間で解く算法を発表した．この算法は線型計画問題の実用上十分な近似解を与えるものであり，近似の精度を上げると，それにつれて手間も増すというものである．ナップサック問題は，解が0か1かになる問題であり，精度は問題にならないので，MH法による暗号文は公開鍵の個数の多項式と全桁数の積程度の手間で解読できる可能性がある．もっとも，この数もかなり大きなものになりそうであり，さらにたいていの場合，上限に近い手間がかかるようなので，実際にはMH法が簡単に破られるということはなさそうだが，NP問題だからといって安心できないという一例にはなるだろう．

R法についても，前述したように，その原論文には明記されていない問題点がある．それは，R法

$$c=x(x+b) \pmod{n}$$

において，重根があると，容易に解読できるということである．この2次方程式が mod p と mod q のいずれでも重根になるのは，判別式が

$$d^2+c=0 \pmod{n}$$

のとき（$d=b/2$）であり，このときの平文は $x=-d \pmod{n}$ である．この対策としては，このような x が通信文に現れないような d（すなわち b）を採用しておけばよい．ところが，もっと危険なのは，d^2+c が mod p, mod q の一方のみで0になるときである．このときは d^2+c と n との最大公約数を計算することによって，n の素因数分解ができ，秘密の p,

q が分かってしまう．このようなことが起こる可能性は，ほぼ $1/p+1/q$ であり，p，q が大きければ無視できるほどである．しかし，一度重根の暗号文を解読されると，以後のすべての暗号文を復号されてしまうので，これは重大な欠陥である．

5. 慣用の暗号系と公開鍵暗号系

慣用の暗号系と公開鍵暗号系の重要な相違点は，鍵を用いる方法にある．DES を代表とする慣用の暗号系では暗号化および復号に同一の鍵を用いるのに対し，公開鍵暗号系では暗号化と復号に異なった鍵を用いる．

通信系，データベースなどの応用分野によって，実現する上で，それぞれに個有な問題点がある．さらに，通信系の場合にも，相互通信，郵便系のような一方向通信，計数署名など，それぞれに応じた解決策が検討され提案されている．

ここでは，特に，相互通信を取り上げ，慣用の暗号系と公開鍵暗号系の実現上の問題点を対比することにする．

相互通信で問題となることは，鍵の管理と配布および認証である．**認証** (authentication) とは，相互通信を行おうとしている加入者が互いに相手の身元を確認することである．

5.1 慣用の暗号系による鍵の配布と認証

慣用の暗号系の場合，鍵の管理と配布は，鍵も通信文を送るのと同じ通信網を使うという仮定に立つと，**鍵配布センタ** (key distribution center, KDC と略記する) が必要になる[8),9)]．この KDC を通信網内でどう位置づけるかによって，種々の方式が考えられるが，代表的な形態としては，集中管理，分散管理，階層管理の三つがある[9)]．集中管理は階層管理の退化した場合であり，完全な分散管理は階層管理の一変形とみなすことができる．

集中管理の場合の鍵の配布および認証のためのプロトコル例[8),9)]を**図-1** に示す．この図は，加入者 A が加入者 B と相互通信を開始したい場合である．K_a，K_b は，それぞれ A，B の秘密の鍵であり，自分と KDC しか知らない．K_c は，A と B との相互通信で使用するために KDC が新しく作り出した鍵である．$[i]^j$ は，平文 i を鍵 j で暗号化した暗号文を表わす．"*REQUEST* (要求)" は A が B に送信したい旨の要求で，恐らく，A および B の一意名である．id は，恐らく乱数で，"*REQUEST*" と id とによって，通信文が途中で改ざんされていないことおよび盗聴者が前に盗聴したものを再送したものではないことを確認できる．id' も恐らく乱数で，関数 f は，事前に A と B との間で取り決めた関数である（文献 8）では，$f: id'-1$)．**貯蔵法** (caching) を用いれば，この通信文の 1 および 2 は省略できる．

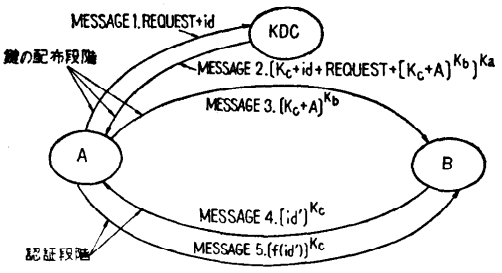


図-1 慣用の暗号系の鍵の配布と認証

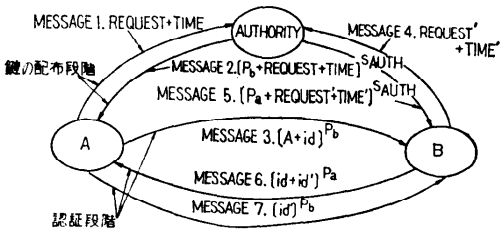


図-2 公開鍵暗号系の鍵の配布と認証

5.2 公開鍵暗号系による鍵の配布と認証

公開鍵暗号系の場合には，公開鍵を登録しておく自動"電話帳"を用意しておけば，特別な管理系はいらないようにみえる[4)]が，電話帳を安全に管理する必要があること，鍵の変更に伴って電話帳を正しく更新する必要があることなどから，やはり慣用の暗号系の場合の KDC に相当する鍵の管理系が必要になる．これを authentication server とか authority と呼ぶ[8),9)]．

公開鍵暗号系の場合の鍵の配布および認証のためのプロトコル例[8),9)]を**図-2** に示す．P_i は i の公開鍵を，S_i は i の秘密の鍵を表わす．$TIME$ (時刻) は改ざん，再送を防ぐための時刻印であり，id および id' は身元確認のためのものである．この場合にも，貯蔵法を用いることにより，通信文 1, 2, 4, 5 をはぶくことができる．

当初の予想に反し，公開鍵暗号系の方が慣用の暗号系よりも複雑になる．しかし，たとえば貯蔵法を用いれば，通信文の数を同数にすることができる．

5.3 その他の比較

慣用の暗号系と公開鍵暗号系とを全般的に対比してみると**表-3** のようになる．

表-3 慣用の暗号系と公開鍵暗号系の比較

	慣用の暗号系	公開鍵暗号系
秘密を保つ方法	鍵に依存 鍵を頻繁に変更	計算の複雑さに基づく一方通行関数に依存
解読の可能性	正しく使えば原理的には不可能（ただし，各種の奇襲方法がある）	原理的には可能だが実際上困難
署 名	特殊な方式を別途工夫しなければならない	方式によっては同時に可能
鍵の数	n 人相互なら $n(n-1)/2$ 個 すなわち n^2 に比例	n 人相互なら $2n$ すなわち人数自体に比例

6. おわりに

以上，公開鍵暗号系における算法，暗号系と復号および解読の手間，公開鍵暗号系の実現性と問題点，慣用の暗号系と公開鍵暗号系について述べたが，最後に今後の課題をいくつか述べておこう．

まず，慣用の暗号体系も公開鍵暗号体系も，それらが安全である根拠は，計算安全性にあることを再度認識しておく必要がある．したがって，たとえば，DESも1日で全数検査ができる機械がいつ頃実現できるかが重要な論争点の一つである．〔DESの可能な鍵の数は $2^{56} \simeq 10^{17}$ である．1μs当り一つの鍵を検査するチップを設計したとすると，10^{11} 秒すなわち約 10^6 日かかる．しかし，そのチップを 10^6 個用いて並列処理を行えば，所要時間はおよそ1日になる．〕また，慣用の体系や一方通行関数でさえも，計算上の安全性を証明できない．したがって，現在のところ公開鍵暗号の安全性を確立する方法がない．計算量の理論の研究を進め，そのような証明を定式化する必要がある．さらに，公開鍵方式のための算法の研究も今後の重要な課題の一つである．運用面では，鍵の管理に関する研究が，まだ十分とはいえない．

暗号に関しては，技術的な問題以外に法的な問題もあるが，計算機システムに暗号が導入されるのは必至であろうことが予想されるので，我が国でも，この方面の研究を十分進める必要がある．米国国家安全保障局 (National Security Agency) は，DESの鍵を56ビットにさせたが，それがDESの落し戸の一つではないかという疑いがかけられている．すなわち，もし米国以外の機関がDESを採用することがあれば，米国国家安全保障局は，それを解読できるのではないかという批判があることは，無視することができない警鐘であろう．

参考文献

1) Rivest, R., Shamir, A. and Adleman, L.: A Method for obtaining Digital Signatures and Public-key Cryptosystems, CACM, Vol. 21, No. 2, pp. 120-126 (1978).
2) Merkle, R. and Hellman, M.: Hiding Information and Receipts in Trapdoor Knapsacks, IEEE Trans. on Information Theory, IT-24 (1978).
3) Rabin, M. O.: Digitalized Signatures and Public-key Functions as Intractable as Factorization, Tech. Rep. MIT/LCS/TR-212, MIT Lab. Comput. Sci. (1979).
4) Diffie, W. and Hellman, M.: New Directions in Cryptography, IEEE Trans. on Information Theory, IT-22, pp. 644-654 (1976).
5) Knuth, D. E.: The Art of Computer Programming, Vol. 2, Addison-Wesley, Mass. (1969).
6) Lempel, A.: Cryptology in Transition, Computing Servey, Vol. 11, No. 4, pp. 285-303 (1979)〔西村和夫訳：暗号学の変遷，コンピュータサイエンス，bit 別冊，pp. 109-125 (1980)〕.
7) Denning, D. E.: Secure Personal Computing in an Insecure Network, CACM, Vol. 22, No. 8 (1979).
8) Needham, R. M. and Schroeder, M.D.: Using Encryption for Authentication in Large Networks of Computers, CACM, Vol. 21, No. 12, pp. 993-999 (1978).
9) Popek, G. J. and Kline, C.S.: Encryption and Secure Computer Networks, Computing Surveys, Vol. 11, No. 4, pp. 331-356 (1979)〔土居範久訳：暗号化と安全な計算機ネットワーク，コンピュータサイエンス，bit 別冊，pp. 149-173 (1980)〕.

（昭和55年10月6日受付）