第11章　暗号システム

11.1　暗号

本章では，整数論が暗号理論にどのように応用されるか，その概略を解説する．

暗号 (cryptography) あるいは**暗号システム** (cryptosystem) とは，第三者に通信内容を知られないように行う特殊な通信方法で、通信文を見ても特別な知識なしでは読めないように変換する表記法のことである． そのような変換を**暗号化** (encryption) という． 暗号化される前の文を**平文** (plaintext)，暗号化によって第三者に通信内容が知られないようにした文を**暗号文** (ciphertext) という． 平文は暗号化によって暗号文に変換されるが，逆に暗号文をから平文を復活させることを**復号化** (decryption) という． 暗号化，復号化のための手順で用いられるパラメータを**鍵** (key) という． 暗号化は復号化と対をなしていて，単純なシステムでは，暗号化のための手順の逆を行うことで復号化がなされる．

さて，整数論の応用という観点から，平文や暗号文はすべて自然数によって表されているものとする． すなわち，日本語や英語など普通の言語で書かれた文を適切な方法で自然数に変換したものを考える． もともとコンピュータ内部では，文字でも何でもすべてが自然数として表現されていると考えてよい． そこで，鍵も自然数として表し，平文，鍵，暗号文の間に整数論的な操作をほどこすことで暗号システムが構築される．

暗号化の鍵と復号化の鍵が同一である（あるいは片方からもう一方が容易に得られる）暗号を**共通鍵暗号** (common key cryptosystem) という． 古典的な暗号はすべて共通鍵暗号であり，20 世紀後半に**公開鍵暗号** (public key cryptosystem) が現れる以前は「暗号＝共通鍵暗号」であった．

11.2　ディフィー・ヘルマン鍵共有

一般に共通鍵暗号は，後で解説する公開鍵暗号に比べ単純に設計され，暗号化・復号化も速やかに実行でき大量データの処理が可能である． しかし，通信をしたい両者があらかじめ**鍵を共有**する必要があり，そこで安全性が損なわれる可能性がある． つまり，普通に考えれば，一方が作成した鍵をもう一方に伝えなくてはならず，その際，鍵を盗まれるおそれがある． このような危険を回避するひとつの方法として，1976 年，W. Diffie と M. E. Hellman は，鍵自体を伝えることなしに両者が同一の鍵を共有する方法を提唱した．これを**ディフィー・ヘルマン鍵共有** (Diffie-Hellman key exchange) という．

一般に，大きな数 b, m を固定し，a, x が関係式 $a \equiv b^x \pmod{m}$ をみたすとする． このとき，x から a は簡単に計算できるが，a から x を求めることは一般には困難である．

この困難な問題を**離散対数問題** (discrete logarithm problem) とよぶ．ここでいう離散対数とは，実数における対数の既約剰余類群 $(\boldsymbol{Z}/m\boldsymbol{Z})^\times$ における類似である．ディフィー・ヘルマン鍵共有の安全性は離散対数問題に依拠しており，実際の手順は以下のように説明される．

(1) 太郎と花子が鍵として「大きな数」を共有したいとする．
(2) 太郎と花子は，大きな素数 p と，法 p に関する位数が小さくない自然数 $g < p$ を用意して共有する．p, g は第三者に知られてもかまわない．
(3) 太郎は $p-1$ と互いに素な「大きな数」 x をランダムに選び，g^x を p で割った余り，すなわち
$$i \equiv g^x \pmod{p}, \quad 0 < i < p$$
によって定まる i を花子に伝える．太郎は x を秘密にする．
(4) 花子は $p-1$ と互いに素な「大きな数」 y をランダムに選び，
$$j \equiv g^y \pmod{p}, \quad 0 < j < p$$
によって定まる j を太郎に伝える．花子は y を秘密にする．
(5) 太郎は，花子から届いた j と，秘密の数 x を用いて
$$k_1 \equiv j^x \pmod{p}, \quad 0 < k_1 < p$$
なる k_1 を計算する．
(6) 花子は，太郎から届いた i と，秘密の数 y を用いて
$$k_2 \equiv i^y \pmod{p}, \quad 0 < k_2 < p$$
なる k_2 を計算する．
(7) 以上の方法で得られた k_1 と k_2 は等しいことが，
$$k_1 \equiv j^x \equiv g^{xy} \equiv i^y \equiv k_2 \pmod{p}, \qquad 0 < k_1, k_2 < p$$
よりわかる．これを $k\ (= k_1 = k_2)$ とする．
(8) k が「大きな数」ならば，これを共通の鍵として採用する．もし k が「小さな数」になってしまったら，もう一度はじめからやり直す．

以上の手順において，p, g, i, j は通信路に乗る（太郎と花子の間を行き来する）ので，悪意のある第三者に読み取られる可能性があるが，x, y と鍵 k は通信路に乗らないので読み取られる可能性はない．いま，p, g, i, j が第三者に知られているとすると，x, y から鍵 k も簡単に計算できてしまうが，i から x を求めることや，j から y を求めることは離散対数問題であり，非常に困難であると考えられる．すなわち，ディフィー・ヘルマン鍵共有の安全性は離散対数問題の困難さによるといえる．

11.3 RSA 公開鍵暗号

前節で述べたように，共通鍵暗号は，暗号化の鍵と復号化の鍵が同一，または，一方からもう一方が容易に導出できるものであった． これとは違い，暗号化の鍵がわかっても復号化の鍵を得ることが現実的に困難な場合，暗号化の鍵を公開しても暗号の安全性は保たれると考えられる． たとえば，インターネット上で多くのユーザからの情報（＝平文）を暗号で受け取りたい人Aは，まず，暗号化，復号化の鍵を 1 組だけ生成し，暗号化の鍵のみを公開する． どのユーザもその鍵を使ってそれぞれの平文を暗号化し，暗号文をAに送ることができる． Aは復号化の鍵を用いて，平文，すなわちユーザの情報を得るわけである．

このように，暗号化の鍵を公開しても安全性が保たれるようなシステムを一般に**公開鍵暗号** (public key cryptosystem) という． 暗号化の鍵を**公開鍵** (public key) とよび，復号化の鍵を**秘密鍵** (private key) とよぶ． 公開鍵暗号は，通信相手の各々に対して別々の鍵を用意する必要がなく鍵管理が容易であるなど共通鍵暗号に対する利点が多い反面，“なりすまし” を防ぐ必要や多くの処理時間を要するなどの欠点もある．

以下で解説する **RSA 暗号**は，1978 年に R. L. Rivest，A. Shamir，L. Adleman により開発された代表的な公開鍵暗号であり，現在広く普及している．

花子が太郎からメッセージを受け取りたいとき，RSA 暗号では次のような手順をふむ．

(1) 【準備】 花子は，2 つの異なる大きな素数 p, q を用意し，積 $N = pq$ を計算する． $\varphi(N) = (p-1)(q-1)$ と互いに素な大きな自然数 $d < \varphi(N)$ をランダムに選び，

$$ed \equiv 1 \pmod{\varphi(N)}, \quad 0 < e < \varphi(N)$$

をみたす e を計算する． (e, N) を公開鍵として太郎に知らせ，d を秘密鍵として秘密にする． p, q および $\varphi(N)$ の値も秘密にする．

(2) 【暗号化】 太郎は，メッセージを自然数で表現した平文 T を用意する． 必要ならば修正をほどこして，$T < N$ および $\gcd(T, N) = 1$ が成り立つようにしておく． この T から，公開鍵 (e, N) を用いて

$$C \equiv T^e \pmod{N}, \quad 0 < C < N$$

によって暗号文 C を作成し，花子に送る．

(3) 【復号化】 花子は，届いた暗号文 C から，秘密鍵 d を用いて

$$T' \equiv C^d \pmod{N}, \quad 0 < T' < N$$

を計算する． このとき $T' = T$ となっていて，太郎からのメッセージ T を復元できる． なぜなら，d, e のとり方から $ed = 1 + m\varphi(N)$ と書けるが，オイラーの定理より $T^{\varphi(N)} \equiv 1 \pmod{N}$ だから

$$T' \equiv C^d \equiv T^{ed} = T^{1+m\varphi(N)} = T(T^{\varphi(N)})^m \equiv T \pmod{N},$$

一方 $0 < T, T' < N$ であったから $T' = T$ を得る．

さて，RSA 暗号の安全性は，公開鍵 (e, N) から秘密鍵 d を導出することが困難であることよる． 以下に，暗号の安全性に係わる秘密鍵の管理についてまとめておく．

- 秘密鍵 d は，法 $\varphi(N)$ に関する e の逆元として，ユークリッドの互除法により効率よく計算できてしまうので，受信者にとって $\varphi(N)$ の秘匿は重要である．
- N の素因数 p, q が知られると，$\varphi(N) = (p-1)(q-1)$ と計算され，上述のとおり秘密鍵が知られてしまうので，p, q の秘匿も重要である．
- 逆に $\varphi(N)$ が知られると，

  $$\varphi(N) = (p-1)(q-1) = N - (p+q) + 1$$

  から $p+q$ がわかり，二次方程式

  $$x^2 - (p+q)x + N = 0$$

  を解くことで p, q が求まってしまう．したがって，N の素因数 p, q が知られることと $\varphi(N)$ が知られることは同等であり，これらの管理は同程度に重要である．

大きな整数の素因数分解は一般には難しく，N だけから p, q を求めることは現実には不可能であると考えられる． すなわち，素因数分解が困難であることが RSA 暗号の安全性の根拠となっているわけである．

11.4 ハイブリッド暗号システム

実際の RSA 暗号では，素数 p, q を選ぶとき，本当に素数であるかどうかの判定をする必要がある． 素因数分解に比べて素数判定は短時間でできること等を考慮して，現状では十進表示で数百桁の素数 p, q を選ぶことになる． したがって N, d, e も数百桁になり，数百桁の数百桁乗の（数百桁の数を法とした）計算を実行することになり，これには相当な時間がかかる． このようなことも含め様々な理由から，一般に公開鍵暗号は大量のデータを即時に暗号化・復号化するのには適さない． そこで，暗号化したい大量のデータは共通鍵暗号によって処理することとし，そこで使われる共通鍵は公開鍵暗号によって暗号化して配送する，という方法が考えられる． すなわち，共通鍵暗号のもつ効率性と公開鍵暗号のもつ鍵管理の安全性というそれぞれの特徴を組み合わせるわけである． 組み合せ方も上記のように単純なものから複雑なものまで様々なものが考えられる． このような暗号方式は**ハイブリッド暗号システム**とよばれ，広く実用化されている．

公開鍵暗号のような新しい暗号方式が提唱された後でも，以前に使われていた方式が捨てられるわけではなく，それらを共存させて「効率性と安全性」のバランスをとりながら新たな暗号システムが構築されている． それらの多くが整数論や代数学に依拠している．さらに,「効率性と安全性」の評価にも整数論的な深い考察が必要とされ，暗号理論は整数論の知識なしでは展開できないと言える．