暗号理論と代数

松本　眞

平成 18 年 12 月 9 日

目 次

基本的参考文献

Douglas R. Stinson “Cryptography: Theory and Practice” CRC Press, 1995.

1　古典的暗号

1.1　暗号とは

メッセージの送り手 (Alice) が受け手 (Bob) にメッセージを送る。が、通信路は敵 (Oscar) が傍受している。どうすれば、メッセージの内容を Oscar に知られずに送ることができるか？

- 戦争における必要性
- 現代における必要性：

  インターネットによる通信では、通信内容は第三者に傍受されている。

  電子メールの内容など、秘密情報の保護

  リモートログインにおける、パスワードの保護

1.2　情報の数値化（コーディング）

送りたい文章は文字列だが、それを数値の列に変えて送る
（デジタル計算機やデジタル通信は、すべての情報を 2 進数に変えて送る。

例 1.1. ASCII と呼ばれるコーディングだと A → 01000001, B → 01000010, ... （2 進数 8 桁）という具合に、数字とアルファベットを 8 ビットの数 (2 進数 8 桁、0 から 255) に対応づける。

文字を数値化するのをエンコード、逆の操作をデコードという。

例 1.2. シーザー暗号：(ローマ皇帝シーザーが使っていたという説あり)　アルファベット 26 文字を 0 から 25 までの数に対応させる。アリスは、送りたい文字列を数値化し、ある定数 K をそれぞれに足して送る。26 以上になったら、26 で割った余りをとる。
ボブは、受け取った数列から K を引く。

例 1.3. K=3 とする。

- エンコード iloveyou → 8, 11, 14, 21, 4, 24, 14, 20
- 暗号化 11, 14, 17, 24, 7, 1, 17, 23
- このままデコード loryhbrx

定式化 1.4. 暗号システムとは、

1. $\mathcal{P}$:可能な文の集合 (plaintext)
2. $\mathcal{C}$:可能な暗号化文の集合 (ciphertext)
3. $\mathcal{K}$:鍵の集合 (key)
4. $e : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$:暗号化関数 (encryption function)
5. $d : \mathcal{K} \times \mathcal{C} \to \mathcal{P}$:復号化関数 (decryption function)

の５つ組で、次の公理を満たすもの。

$$\forall x \in \mathcal{P}, \forall k \in \mathcal{K}, d(k, e(k, x)) = x.$$

アリスとボブは、手紙などの安全な手段であらかじめ鍵 k を共有しておく。アリスは送りたいメッセージ x に対し、その鍵 k を用いた暗号化文 $e(k, x)$ を送信する。ボブは受け取ったものに対して $d(k, -)$ を施すことにより、$d(k, e(k, x)) = x$ を復元できる。が、オスカーは k がわからないので復元できない。

例 1.5. $K = \mathcal{C} = \mathcal{P} = \mathbb{Z}/26 := \{0, 1, \ldots, 25\}$ で加減乗算を mod 26 で計算するものとし、

$$e(k, x) = x + k, \quad d(k, y) = y - k$$

としたものがシーザー暗号。

注意 1.6. $d(k, -)$ は、「元をもらって $-$ のところに入れて計算を行う」関数、すなわち

$$d(k, -) : \mathcal{C} \to \mathcal{P}, \quad y \mapsto d(k, y)$$

を表わす。

暗号の解読 (break the cipher)：オスカーが、暗号化文からもとの文を推測すること（１００%正しくなくてもよい)。

- $\mathcal{K}$ の元を全て調べつくせるようならば、オスカーは傍聴した $e(k, x)$ に対して全ての $k' \in \mathcal{K}$ について、いちいち $d(k', -)$ をほどこしてやり、結果が「意味のある文章」になったものを元の平文と推測すればよい。
- $\mathcal{P}$ の元を全て調べつくせるだけでも、(k を毎回取り替えない限りは）暗号システムとしては脆弱。

  例えば、$\mathcal{P}$ がアルファベット 26 文字の集合であるとする。$\mathcal{K}$ が膨大であっても、ひとたび $k \in \mathcal{K}$ を選ぶと $e(k, -) : \mathcal{P} \to \mathcal{C}$ はある一つの単射である。

英語のつづりでは、もっとも出現頻度が多いのは e である。したがって、暗号化文（$\mathcal{C}$ の元）がたくさん入手できたとき、出現頻度順のヒストグラムを作れば、どれが e を暗号化したものかがわかる可能性が高い。

また、「同じ文字が二個続く」のは限られている。success, balloon のように c,s,l,o は続くが、ii,aa は（ほとんど）ない。q のあとには必ず u がくる。これらのことを用いて、統計的手法で解読することができる。

一つの鍵 k を使う限り、同じ $\mathcal{P}$ の元は同じ $\mathcal{C}$ の元に暗号化されるので、送信するメッセージ中に同じ $\mathcal{P}$ の元が何度も現れるようだとやや危険。

そこで、$\mathcal{P}$ の元の数を 2^{128} から 2^{256} 以上にすることが推奨されている。

1.3 攻撃

インターネット通信など現代的環境では、さまざまな攻撃が考えられる。

まず前提として、暗号システムの情報はもれている、と考えた方がよい。すなわち、$\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d$ はすべて Oscar にも既知とする。暗号システムのプログラムは無料配布されていたり、有料で販売されていたりするので、Oscar はそれを手に入れられると考えたほうが良い。

では、何が Alice と Bob にのみ知られた情報であるのか、というと鍵 $k \in \mathcal{K}$ である。

暗号の攻撃でもっとも標準的なのは、Oscar が暗号化文をたくさん入手して、鍵 k を推測するというものである。

しかし、より Oscar にとって有利な状況として、「平文の一部が Oscar にもれている」という可能性がある。例えば、メールの冒頭部分はもれている可能性がある。既知平文攻撃 (known plaintext attack) とは、Oscar が平文とその暗号化文の組をたくさん集めることで、鍵 k を推測することである。

さらに Oscar にとって有利な状況は、「Oscar が選んだ任意の平文を、Alice に暗号化させて Oscar に送らせることができる」というものである。このような状況もありうる。選択平文攻撃 (chosen plaintext attack) という。

1.4 Block cipher

既知平文攻撃では、$\mathcal{P}$ の元と、対応する $\mathcal{C}$ の元の組が大量に漏れることを前提としている。$\mathcal{P}$ の元の個数が少ないと、全てが出尽くせば、暗号は解読されつくしたことになる。

そこで、$\mathcal{P} = \mathcal{C}$ を 256 ビット程度の整数の集合にし、

$$e(k, -) : \mathcal{P} \to \mathcal{C}$$

を「十分複雑な」写像にして、送りたい平文を 256 ビットごとに切り離して、各 256 ビットを $e(k,-)$ で変換して $\mathcal{C}$ の元の列とすることを、ブロック暗号という。この 256 ビット単位を「ブロック」という。

現在、世界標準とされているものは AES と呼ばれる方式である。

しかし、非常に冗長な平文を送った場合、例えば 2^{256} の可能性のうち 26 通りしか使わないなどの場合は、統計的攻撃さえ有効である。

1.5 Stream cipher

ブロック暗号では、各ブロックに対して同一の鍵 k を用いて暗号化を行う。それに対し、ストリーム暗号とは、鍵の列 $k_1, k_2, \ldots, k_n$ を用いて暗号化を行う。

平文を $\mathcal{P}$ の n 個の列とする。Alice と Bob は共有鍵 k を用いて、漸化式など決定性の方法で、$\mathcal{K}$ の元の列 $k_1, k_2, \ldots, k_n$ を生成する。そして、Alice は i 個めの $\mathcal{P}$ の元には、$e(k_i,-)$ を施して、それを Bob に送る。Bob は $d(k_i,-)$ を施して、もとの平文を得る。

SNOW2.0 という方式では、$\mathbb{F}_2$ 上の線形漸化式で得られた 16 ビット列に非線形な変換を行ったものを鍵の列とし、$e(k_i,-)$ は $-$ と k_i を $\mathbb{F}_2$ ベクトルとして足す、という方式を採用している。

（一般には、k_i は平文 $x_1, \ldots, x_{i-1}$ に依存して変更されてもよく、この場合は非同期式ストリーム暗号といい、上のものを同期式ストリーム暗号という。）

1.6 完全に安全な暗号

Shannon は情報量の概念を用いて、完全に安全な暗号の概念を作った。

ここでは、その中で簡単に説明できる部分のみを説明する。暗号化関数 $e : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ において、どの $x \in \mathcal{P}$ に対しても、$e(-,x) : \mathcal{K} \to \mathcal{C}$ が全単射であると仮定する。

送るメッセージの長さは $\mathcal{P}$ の元 n 個列と決まっている。このとき、Alice は前もって $\mathcal{K}^n$ の中から元を一様ランダムに選び、$(k_1, k_2, \ldots, k_n)$ として、それを安全な方法で Bob に送っておく。送りたいメッセージ $x_1, x_2, \ldots, x_n$ に対して、$e(k_1, x_1), e(k_2, x_2), \ldots, e(k_n, x_n)$ を暗号化文とする。この $k_1, \ldots, k_n$ は、一回つかったら捨てて忘れる。

一回しか鍵の列を使わないので、「一回限りの帖」(one-time pad) と呼ばれる。

このような暗号方式では、送りたいメッセージが何であっても、k_i を一様独立ランダムに選んでいるのだから、全単射性の条件から、暗号化文は一様独立ランダムに選ばれた $\mathcal{P}^n$ の元となる。したがって、これを傍受したものはどのように高い計算能力をもってしても、もとの平文に関するデータを得ることができない。

こういう方法は、軍事などのシリアスな場面では有効であるが、インターネット向きではない。メッセージと同じ長さの鍵を安全に送る方法が必要であり、そんなことができるならそこでメッセージを送ってしまえばよい。

このような悩みを解決するのが、公開鍵暗号システムである。

2 公開鍵暗号

今まで見てきた暗号システムでは、別の安全な通信手段で鍵を交換する必要があった。このような暗号システムを共有鍵暗号という。鍵さえ安全に交換できれば、あとは効率よく暗号化通信が行える。

公開鍵暗号システムは、安全な通信手段を使わずに、暗号システムを実現する方法の一つである。

2.1 Diffie-Hellman 鍵交換

まず、安全な通信路なしで鍵を交換するもっともシンプルなシステムである DH 鍵交換を説明する。

以下、数学的に厳密ではない議論をしばらくする。

われわれは、通信路の情報がだだ漏れであるという仮定をおいている。この時点では、Alice と Bob と Oscar は対称である。ここから、Alice と Bob のみが知り、Oscar は知らない情報 k を作り出さなくてはならない。そのためには、なんらかの情報の非対称性を作らなくてはならない。それは、Alice がある集合 S からランダムに $s_A \in S$ という元を選ぶことによってなされる。これにより、Alice のみが知り、Bob と Oscar は知らない情報 s_A を作り出すことができる。これだけでは Bob と Oscar は対称である、すなわち区別がない。そのため、Bob もあるランダムな元 s_B を選ぶことになる。これで、対称性はくずれる。

ここから、s_A、s_B に関する通信を行って、Alice と Bob が共有でき、かつ Oscar が知らない情報を作らなければならない。この際、s_A をそのまま Bob に送ってしまっては Oscar にばれて、Alice と Oscar が持つ情報が同じになってしまう。

したがって、Alice は Bob になんらかの関数 f を用いて s_A を $f(s_A)$ に変換して送ることになる。Bob は Alice に対して、s_B と $f(s_A)$ に依存した計算結果を送ることもできるが、とりあえず簡単のため対称的に、ある関数によって $g(s_B)$ を計算して Bob に送ることにする。この時点で 3 人が持っている情報は

- Alice は s_A, $g(s_B)$

- Bob は $f(s_A)$, s_B
- Oscar は $f(s_A)$, $g(s_B)$

という状態になる。f, g もあらかじめ Alice, Bob に配布されているものであるから、一般に公開されているものだと考えてよい。f または g の逆関数が計算できるものなら、$f(s_A)$ から s_A を Oscar が求めることができるので、Oscar が Alice の持つ情報を回復できることになり、目標は達成できない。

目標が達成されるには、「s_A と $g(s_B)$」からも、「$f(s_A)$ と s_B」からも計算ができるが、「$f(s_A)$ と $g(s_B)$」からは計算できない、そんなものがあれば、それを Alice と Bob が共有鍵として利用すればよい。

そのようなものとして初めて提唱され、いまでも有効な方法と思われているものが Diffie-Hellman の鍵交換法である。以下、これを説明する。

集合 S の集合 T への作用とは、

$$S \times T \to T, \quad , s, t \mapsto s * t$$

可換な作用とは、

$$s_1 * (s_2 * t) = s_2 * (s_1 * t)$$

が成り立つことである。S, T を十分大きな集合とし、可換な作用 $*$ を一つ決め、公開する。$t \in T$ を一つ決め、公開する。Alice はランダムに $s_A \in S$ を発生し、秘匿する。Bob はランダムに $s_B \in S$ を発生し、秘匿する。Alice は $s_A * t$ を Bob に送る。Bob は $s_B * t$ を Alice に送る。

Alice は $s_B * (s_A * t)$ を求める。Bob は $s_A * (s_B * t)$ を求める。これらは可換性より同一なので、これを共有鍵として用いる。

Oscar は、$s_A * t$, $s_B * t$ を入手できるが、そこから $s_A * s_B * t$ を計算しなくてはならない。

そこで、このようなシステムでは t と $s * t$ が既知であっても r が求めにくいという必要性がある。そのような二項演算はいくつか知られている。（本当に難しいことが保障されているものはない。NP=P 問題が未解決な限り、保障はない。）

そのような問題の一つとして、離散対数問題 (discrete log problem, DLP) が挙げられる。有限群 G に対し、$S = \mathbb{N}$, $T = G$,

$$\mathbb{N} \times G \to G, \quad (n, g) \mapsto g^n$$

である。すなわち、(g, g^n) から n を求める問題が DLP である。

ある種の有限群、$(\mathbb{Z}/p)^\times$, $\mathbb{F}_{p^n}^\times$, 有限体上の楕円曲線の有理点のなす群、などが難しいとされている。逆に言うと、多くの有限群で DLP は難しくない。

DLP を用いると、Alice は (g^{r_A}) を Bob に送り、Bob は (g^{r_B}) を Alice に送り、Alice は $(g^{r_A})^{r_B}$ を計算し、Bob は $(g^{r_B})^{r_A}$ ををを計算し、共有鍵として使用することができる。

「(g, g^{r_A}, g^{r_B}) から $g^{r_A r_B}$ を計算すること」を DH 問題といい、離散対数問題よりも弱いが、同値であるかいなかは未解決問題である。

これをそのまま使うことは、intruder in the middle 攻撃の存在により危険である。Oscar は、Alice と Bob の間に入って、Alice に対しては Bob になりすまし、Bob に対しては Alice になりすます、ということが可能だからである。

この問題を解決するには、Alice が Alice であること、Bob が Bob を証明する方法（本人認証）が不可欠であるが、それにはあとで触れる。

2.2 冪の計算

上のような応用では、群 G において、g^n の計算が必要である。積がある時間で計算できるとき、g^n は何回の積で計算可能か。n を二進展開する方法で、$O(\log n)$ の積で計算が可能である。

$n = 13 = (1101)_2$ のとき、上の桁からみて $1, 11, 110, 1101$ と 4 ステップにわける。(以下、2 進表記を使う。) $g, g^{11} = g^{10}g, g^{110} = (g^{11})^{10}, g^{1101} = (g^{110})^{10}g$ と計算していく。これを 2 進法 (binary method) という。

2.3 難しい離散対数問題とは

DH 鍵交換では、$G, g \in G$ は公開されていると考えられる。そして、g の冪しか使わないのだから、群として必要なのは G 全体ではなく g が生成する巡回群 $< g >$ しか使わない。よって、G は巡回群であるとしても一般性を失わない。仮に今、G が有限巡回群であるとする。すると同型

$$G \cong \mathbb{Z}/N$$

がある。(N は位数。)

ところで、$\mathbb{Z}/N$ においては、離散対数問題は容易にとける。より正確に言えば、$\mathbb{Z}/N$ において、乗算と、余りつき割り算という「群の演算に入っていない演算」が高速にできるなら、離散対数問題はそれらの $O((\log N))$ 回の演算によって求めることができる。

$\mathbb{Z}/N$ においては、離散対数問題は「$g, z \in \mathbb{Z}/N$ に対して $z = rg$ となる r を求めよ」という問題になり、

$$z \equiv rg \mod N$$

なる r を求めよ、すなわち

$$z = rg + bN$$

なる r と b を求めよ、という問題になる。これは、古くから知られている「与えられた整数 x, y, z に対し、

$$ax + by = z$$

となるような a, b を求めよ、という問題になる。

「特殊解 a^*, b^* を求める」ことができれば、一般解はやさしくもとまる。

$$a^*x + b^*y = z$$

から辺同士を引くと

$$(a - a^*)x + (b - b^*)y = 0$$

となり、$d := gcd(x, y)$, $x = x_0 d, y = y_0 d$ とおくことにより

$$(a - a^*)x_0 = (b^* - b)y_0$$

と同値になるが、x_0 と y_0 が互いに素であるから両辺は $x_0 y_0$ の倍数で、m 倍とすれば

$$a - a^* = my_0, b^* - b = mx_0 (m \in \mathbb{Z})$$

という形で一般解が求まるからである。

特殊解を求めるには、まず、$d|z$ でなければ解が存在しないことが分かる。$d|z$ であれば、

$$ax + by = d$$

なる a, b を求めて、それを z/d 倍すればよい。ので、このような a, b を求めればよい。

余りつき割り算を用いて、このような計算が可能であることはユークリッドの互除法の理論として知られている。

d は、$ax + by$ の形に表わされる最小の正整数である。そこで、まず $qx + y$ の形に表わされる最小の正整数を求める。これは、y を x で割った余り r に他ならない。次に、$x + qr$ の形で表わされる最小の正整数を求める。これは、x を r で割った余りに他ならない。この余りも x, y の整数係数一次結合で表わされる。その係数を得るのは容易である。

このように係数を計算しながら互除法を繰り返して、d に到達するまでやればよい。

互除法の反復の回数は $O(\log_\alpha N)$ であることが知られている。ここに、α は黄金比 $(1 + \sqrt{5})/2$ である。

$$O(\log_2 N) = O(\log_2(\alpha) \log_\alpha(N)) = O(\log_\alpha N)$$

であるから、余りつき除算や乗算の $O(\log N)$ 回演算で $\mathbb{Z}/N$ の DLP は求まる。

なお、同型

$$\phi : \mathbb{Z}/N \cong G, n \mapsto g^n$$

を通して考えたとき、右辺における DH 問題は、左辺においては「$1, r_A, r_B$ が与えられたとき、$r_A r_B$ を計算せよ」という問題である。すなわち、乗法を計算せよという問題である。ϕ の逆関数を求めよ、というのが DLP 問題であり、左辺における乗法を右辺の中で計算せよ、というのが DH 問題である。

2.4 DLP が解ける群

n 次対称群 S_n では、n が計算機で数え上げられる程度の大きさなら DLP が解ける。$g, y \in S_n$ が与えられたときに $y = g^r$ なる r を（あれば）求めるアルゴリズムが存在する。それには、g による $\{1, 2, \ldots, n\}$ への作用の軌道分解を行う。各軌道において、y の作用が g の何乗であるかを計算することができる。i 番目の軌道の長さを n_i とするならば、その軌道上に制限して

$$y = g^{r_i}$$

となる r_i を求めることができる。このとき、

$$r \equiv r_i \mod n_i$$

を全ての軌道について連立して、r を求めればよい。

2.5 公開鍵暗号システム

共有鍵暗号では、鍵は Alice と Bob の間で共有されるものであった。公開鍵暗号では、鍵はふたつ、「鍵をかける専用の鍵（暗号化鍵）k_e」と「鍵をはずす専用の鍵（復号化鍵）k_d」とを用意する。

公開鍵暗号では、モデルは

$$e : \mathcal{K}_e \times \mathcal{P} \to \mathcal{C}$$

$$d : \mathcal{K}_d \times \mathcal{C} \to \mathcal{P}$$

および、性質

$$e(k_e, -) \circ d(k_d, -) = \mathrm{id}_{\mathcal{P}}$$

を持つ鍵のペア

$$(k_e, k_d) \in \mathcal{K}_e \times \mathcal{K}_d$$

をランダムに一つ選び出す方法からなる。

Alice は、このような k_e と k_d を秘密裏につくり、k_e を一般に公開し、k_d を秘匿しておく。この状況で、Bob から Alice に向けて情報を暗号化して送ることができる。Bob は $e(k_e, -)$ を計算できるからである。（ちなみに、Oscar もできる。）Alice は、$d(k_d, -)$ を使って複号化することができる。ここで、暗号の安全性のために仮定されていることは、「k_e から k_d を推測することが困難である」ということである。掛けるための鍵は公開されているが、そこからはずすための鍵は推測が困難であるという仮定である。

より正確に言えば、k_e から $d(k_d, -)$ を計算することが困難である、という仮定である。

Alice から Bob に暗号通信をするためには、同様に Bob が k'_e と k'_d を作って k'_e を一般に公開しておく。

2.6 ElGamal 公開鍵暗号系

G を群とし、$g \in G$ を固定する。$\mathcal{P} := G$, $\mathcal{C} := G \times G$, $\mathcal{K}_e = G$, $\mathcal{K}_d := \mathbb{Z}$ とする。

Alice が作る公開鍵と秘匿鍵のペアは $k_e := g^a$ と $k_d := a$ である。

Bob 側が行う計算は、

$$e(k_e, x) = (g^r, xk_e^r) \quad (= (g^r, xg^{ar}))$$

である。ここに、r は Bob が選ぶ秘匿された整数乱数である。（したがって、e はこの場合厳密に言えば関数ではなく、確率的にしか値が決定されない。）

Alice 側が行う計算は、

$$d(k_d, (y, z)) = zy^{-k_d} \quad (= (zy^{-a}))$$

である。

2.7 RSA 公開鍵暗号

RSA 公開鍵暗号系は、1977 年世界で最初に実現された公開鍵暗号系である。これも、群に基づく暗号系である。

Alice はある有限群 G を決める。その位数を m とする。Alice は乱数 a を発生し、$ab \equiv 1 \mod m$ となる b を互除法により求めておく。$\mathcal{P} = \mathcal{C} = G$ であり、鍵の集合は $\mathcal{K}_e = \mathcal{K}_d = \mathbb{Z}/\phi(n)$ である。Alice が公開するのは G と $k_e := a$ である。秘密鍵は $k_d := b$ である。

暗号化関数は

$$e(x, a) := x^a$$

で与えられる。復号化関数も同じ

$$d(y, b) := y^b$$

で与えられる。暗号化して複合化すると元に戻ることは、

$$(x^a)^b = x^{ab} = x^1$$

より従う。ここで、$ab \equiv 1 \mod m$ を用いた。

問題は、この方法で公開鍵暗号を実現するには、少なくとも m を秘匿しないとならないという点にある。m が Oscar の手に渡れば、a から b が互除法で求まってしまうからである。

群 G とそこでの冪の計算方法は、公開してしまうのにも関わらず、その位数は知られない、などという都合のいい状況を作れるであろうか。それが可能だというのが RSA 暗号の背景にある。

ある大きな整数 n に対して

$$G := (\mathbb{Z}/n)^\times$$

を考えてみよう。この群の位数は $\phi(n)$ と書かれる（いわゆる、オイラー関数である）。$\phi(n)$ は、n の素因数分解がわかれば求めることができる。が、素因数分解がわからないときに $\phi(n)$ を求めることは困難であると信じられている。

特に、Alice が非常に大きな異なる素数 p, q を探し、$n = pq$ としたとしよう。このとき、

$$\phi(n) = (p-1)(q-1)$$

となる。n を公開したとしても、$n = pq$ なる素因数分解がわからない限り、$\phi(n)$ を計算することは困難であると信じられている。というのは、もしそれができたとすると、$p + q = n - \phi(n) + 1$ となり、$pq = n$ も公開されているから、二次方程式の二根として p, q を求めるのは容易であるからである。

一般に、与えられた整数を素因数分解するのは、その整数に大きな素因数 (たとえば 2^{500} 程度) が二つ以上ある限り困難であると信じられている。

2.8 素数の探索

RSA を実現するには、Alice が private な、誰にも知られていない素数を二つ探索することが必要である。

Oscar にとって、推測が容易な素数は使うべきでない。例えば、「2^{500} 以上の最小の素数」とか「$2^n - 1$ の形の素数」(Mersenne Prime、メルセンヌ素数と呼ばれる) などは推測されやすい。

実用的には、「2^{500} から $2^{501} - 1$ までの整数をランダムにとり、それが素数であるかどうかを判定し、みつかるまで繰り返す」ということが行われている。

2.9 素数判定

N が素数であることの判定には、$\sqrt{N}$ までの自然数で割り切れないことを確かめればよい。が、2^{500} のような大きな数に対しては、これは不可能である。

そのため、より高速（で、かつある意味不完全）な素数判定法が利用されている。

Fermat 判定法

定理 2.1. (Fermat の小定理)
n が素数ならば、$a \in (\mathbb{Z}/n)^\times$ に対して

$$a^{n-1} = 1 \mod n.$$

証明は、群 $(\mathbb{Z}/n)^\times$ の位数が $n-1$ になることより従う。

Fermat テストは、

1. $a \in \mathbb{Z}/n$, $a \neq 0$ をランダムに選ぶ
2. $a^{n-1} \mod n$ を計算する
3. 結果が 1 でなければ「n は合成数」という結論を返す。結果が 1 ならば「n が素数か合成数かわかりません」という結論を返す。

ものである。上で「わからない」と判断された n を「a に関する擬素数」という。これを何度も繰り返して、一度も「合成数」と判断されなかった n は素数である可能性が高い。

定義 2.2. 全ての $a \in (\mathbb{Z}/n) - \{0\}$ に対して、Fermat テストを通過するような非素数 n を Carmichael 数という。無限に存在することが知られている。

10000 以下の Carmichael 数は 561, 1105, 1729, 2465, 2821, 6601, 8911 の 7 つである。

Carmichael 数の存在により、Fermat テストで棄却できない合成数があることが知られる。このような問題点を改善するのが Miller-Rabin テストである。

Miller-Rabin テスト

命題 2.3. n を素数 ≥ 3 とし、$a \in (\mathbb{Z}/n) - \{0\}$ とする。$n-1$ を 2 で割り切れるだけ割り続けて、$n-1 = 2^r d$, d は奇数、と因数分解しておく。すると、$a^d = 1$ であるか、もしくは $a^{2^k d} = -1$ となる $0 \leq k < r$ が存在する。

証明. $a^{2^r d} = 1$ であるから、$a^{2^k d} \neq 1$ となる最大の $0 \leq k < r$ があるか、もしくはそれが存在しない（すなわち $a^d = 1$ となる）。

存在したならば、最大性より $(a^{2^k d})^2 = 1$ である。$x^2 - 1 = 0$ の解は、整域内には高々二個しかなく、それらは $-1, 1$ に限る。$n \geq 3$ より、これらは相異なる。k のとりかたから $a^{2^k d} = -1$ となる。 □

Miller Rabin テスト：$n \geq 3$、奇数、と仮定する。

1. $n - 1 = 2^r d$, d:奇数、を求める。
2. $a \in \mathbb{Z}/n$, $a \neq 0$ をランダムに選ぶ
3. a^d, a^{2d}, a^{4d}, $\ldots a^{2^{r-1}d}$ をこの順に計算し、最初に 1 となるか、どこかで -1 になることを確かめる。ならなければ、「n は合成数」と判定する。
4. 上で合成数と判定されなければ、「n が素数か合成数かわかりません」という結論を返す。

n が合成数であるとき、上のような a であって「合成数である」という判定を与えるものを「証拠」(witness) という。

Carmichael 数においては、Fermat テストに関する「証拠」は存在しなかった。Miller-Rabin テストが Fermat テストより優れている点は、次の定理にある。

証明が煩雑なので、とりあえずは定理の意味するところだけ把握すると良い。

定理 2.4. n を奇数の合成数とする。$(\mathbb{Z}/n)^\times$ の元で、Miller-Rabin テストにおける「証拠」とならないようなものは全体の高々 1/4 以下しか存在しない。

定理の言っていることは、次の点で重要である。すなわち、n が合成数であるとき、$a \in (\mathbb{Z}/n)^\times$ をランダムに選べば、Miller-Rabin テストにより確率 3/4 以上で n を合成数であると決定することができる。

逆にいうと、合成数であると見抜けない確率は 1/4 以下である。したがって、合成数 n に対して例えば 100 回 Miller-Rabin テストを行い合格する確率は $4^{-100} = 2^{-200} \sim 10^{-30}$ 以下である。言い換えると、100 回の Miller-Rabin テストに合格した n が合成数である確率は極めて低く、実用的には「素数である」と考えても差し支えない。

定理 2.4 の証明には、初等整数論や初等群論の理論が使われる。学部 3 年くらいまでの代数学の、ちょうど良い応用である。

補題 2.5. p を奇素数とする。$(\mathbb{Z}/p^e)^\times$ は、位数 $(p-1)p^{e-1}$ の巡回群である。

補題 2.6. （第 3 準同型定理）
$G > H > N$ を、群とその二つの正規部分群とする。このとき、H/N は自然に G/N の正規部分群とみなすことができ、

$$G/H \cong (G/N)/(H/N)$$

なる自然な群同型が存在する。

これらの補題の証明は、標準的な代数学の教科書を参照のこと。

証明. (定理 2.4 の)
$n = p_1^{e_1} \cdots p_s^{e_s}$ と、素因数分解する。すると、中国式剰余定理から

$$(\mathbb{Z}/n)^\times \cong (\mathbb{Z}/(p_1^{e_1}))^\times \times \cdots (\mathbb{Z}/(p_s^{e_s}))^\times$$

となる。この対応を $a \mapsto (a_1, \ldots, a_s)$ であらわそう。
$a \in (\mathbb{Z}/n)^\times$ であって、証拠とならないものを考える。$a^d = 1$ もしくは、ある k によって $a^{2^k d} = -1$ が成立するが、前者であれば $-a$ が $(-a)^d = -1$ を満たすから、$k \geq 0$ に対し後者を満たすような a が存在する。このような a が存在する $0 \leq k \leq r-1$ のうちで最大なものをとり、K とする。

$$L := \{a \in (\mathbb{Z}/n)^\times | a^{2^K d} = \pm 1\}$$

と置くと、これは部分群で、K の最大性から

$$\{a : \text{証拠でない}\} \subset L$$

となる。なぜなら、証拠でない a は $a^d = 1$ もしくは $a^{2^k d} = -1$ $(0 \leq \exists k < r)$ を満たすからである。
したがって、$[(\mathbb{Z}/n)^\times : L] \geq 4$ を示せば定理の証明は終わる。
いま、上記の中国式剰余定理による同一視を行って、

$$L' := \{a = (a_1, \ldots, a_s) | a_i^{2^K d} = \pm 1 (\forall i)\} < (\mathbb{Z}/n)^\times$$

なる部分群を考える。$L' > L$ である。いま、

$$f := (-)^{2^K d} : (\mathbb{Z}/n)^\times \to (\mathbb{Z}/n)^\times$$

なる群順同型を考えたとき、中国式剰余定理を介して

$$L = f^{-1}(\{(1, 1, \ldots, 1), (-1, -1, \ldots, -1)\}),$$

$$L' = f^{-1}(\{(\pm 1, \pm 1, \ldots, \pm 1)\}),$$

である。この ± 1 の符号はばらばらに自由に選んでよいものとする。

ここで、K のとりかたから、$a^{2^K d} = -1$ となるような a は存在する、すなわち

$$f : L \to \{(1, 1, \ldots, 1), (-1, -1, \ldots, -1)\}$$

が全射であることに注意しよう。この a を上のように分解して $a = (a_1, \ldots, a_s)$ とおこう。成分 a_i のうち好きなものを 1 に取り替えたものを a' とおけば、

$$a'^{2^K d} = (\pm 1, \ldots, \pm 1)$$

の符号のパターンを任意に選ぶことができる。したがって、

$$f : L' \to (\{(\pm 1, \pm 1, \ldots, \pm 1)\})$$

は全射である。

準同型定理 2.6 より、

$$L'/L = (L/\ker f)/(L'/\ker f) = \{(\pm 1, \pm 1, \ldots, \pm 1)\})/(\{(1, 1, \ldots, 1), (-1, -1, \ldots, -1)\}).$$

である。もし $s \geq 3$ ならば、この指数 2^{s-1} は 4 以上であり、証明は終わる。

もし $s = 1$ ならば、$n = p^e$ となる。このとき、補題 2.5 より $(\mathbb{Z}/(p^e))^\times$ は位数 $(p-1)p^{e-1}$ の巡回群である。L の元の位数は $2^{K+1}d | p^e - 1$ の約数であるから、$p-1$ の約数でなくてはならず、そのようなものは全体の $1/p^{e-1}$ しかない。これが $1/4$ に達しないのは $p = 3, e = 2$ のときのみであるが、$(\mathbb{Z}/9)^\times = \{1, 2, 4, 5, 7, 8\}$ には証拠が 2,4,5,7 の 4 つあり、$1/4$ 以上の割合で存在する。

もし $s = 2$ ならば、$n = p_1^{e_1} p_2^{e_2}$ である。$[L' : L] = 2$ だから、$[(\mathbb{Z}/n)^\times : L'] \geq 2$ を言えばよい。これが成立しないとすると $(\mathbb{Z}/n)^\times = L'$ であり、L' の元は $2^K d$ すると各成分が ± 1 となるから、さらに自乗すれば $a^{2^{K+1}d} = 1$ となる。$2^{K+1}d | n - 1$ より、特に n は Carmichael 数である。証明は、次の補題により完結する。 □

補題 2.7. 奇数 n が Carmichael 数であれば、それは三つ以上の相異なる素因数を持ち、それらの重複度は 1 である。

証明. n を素因数分解して

$$\mathbb{Z}/n \cong \mathbb{Z}/(p_1^{e_1}) \times \cdots \mathbb{Z}/(p_s^{e_s})$$

とおく。各成分の乗法群は $(p_i - 1)p_i^{e_i - 1}$ 次巡回群である。仮定より、$n-1$ 乗すると何でも 1 になる。ということは $(p_i - 1)p^{e_i - 1} | n - 1$ である。右辺は p_i で割り切れないから、$e_i = 1$ でなくてはならない。$n = pq$ と二つの異なる素数の積であるとすると、$p - 1 | pq - 1$ であるが、 $\bmod (p-1)$ で見ると $pq - 1 \equiv q - 1$ より $p - 1 | q - 1$。役割を取り替えて $q - 1 | p - 1$。よって $p = q$、これは矛盾。 □

2.10 署名

Alice が、自分の書いた文章 $x \in \mathcal{P}$ に対して、それが確かに自分の書いたものであるという証拠＝署名を施したいとする。どうすれば良いか。

公開鍵暗号のときと同様に、どこか信頼できるホームページなどで、Alice を identify する情報が公開されている必要性がある。

定義 2.8. 署名系 (signature scheme) とは、

$\mathcal{P}$：平文の集合 $\mathcal{A}$：署名の集合 $\mathcal{K}_s$:署名用鍵の集合 $\mathcal{K}_v$:署名検証用鍵の集合

$s : \mathcal{K}_s \times \mathcal{P} \to \mathcal{A}$:署名関数

$v : \mathcal{K}_v \times \mathcal{P} \times \mathcal{A} \to \{T, F\}$:検証関数

と、次の公理をみたす $k_s \in \mathcal{K}_s, k_v \in \mathcal{K}_v$ の組をランダムに生成する手段からなる。

$$v(k_v, x, a) = T \Leftrightarrow a = s(k_s, x)$$

Alice は、k_s と k_v の組を生成し、k_v を公開して k_s は秘匿しておく。文 x に署名をつけたいとき、Alice は x に対する署名として $s(k_s, x)$ を計算し、組 $(x, s(k_s, x))$ をもって「Alice の署名付き文」とする。

Bob（に限らず任意の人）は、(x, a) を受け取ったとき、Alice の公開している検証鍵 k_v を用いて $v(k_v, x, a)$ を計算し、T(True) であることを確認することで a が Alice の署名であることを確認する。

以上により実現できていることは「Oscar は、勝手な文章に Alice の署名をつけることはできない」ということ（だけ）である（仮定は、k_s が k_v などの公開されている情報から推測できない、というものである）。

つまり、上で実現されていることは、「『Alice の署名は k_s の情報を持つ人（＝ Alice 自身）が行った』と誰にでも確認できる」だけである。

（たとえば、Oscar は、Alice の文 x を傍受して、勝手に自分の署名をつけて流通させることができる。）

署名の典型的な使用例は、公開鍵暗号と組み合わせて用いるものである。通信している相手が確かに Alice であるかどうかを、Bob は確認することができる。

Alice は、自分の書名付きメッセージ (x, a) を、Bob の公開された暗号化関数により暗号化して $e(k_{e,B}, (x, a))$ して Bob に送る。Bob はそれを復号して (x, a) を得て、それから Alice の公開している k_v を用いて「Alice が署名していること」すなわち、Alice 自身からのメッセージであることを確認する。

これで実現されているのは、Bob が「Alice 自身がかつて署名したメッセージである」と確認できるということだけである。Oscar が $e(k_{e,B}, (x, a))$ を傍受しておいて、何度も Bob に送りつけるというような攻撃は可能である。Bob が銀行で、Alice の文章が「振込み依頼」である、というような状況では依然と

して問題である。これをさけるには、Alice 側がメッセージの中に「何番目のメッセージか」あるいは「メッセージを送った時刻」を明示し、Bob 側がそれを記録しておくことで、繰り返しの利用を防ぐ必要性がある。

RSA は署名方式の一例を与える。$n = pq$ とし、$\mathcal{P} = \mathcal{A} = \mathbb{Z}/n$ とする。$k_s, k_v \in \mathbb{Z}/\phi(n)$, $k_s k_v = 1$ としておいて、

$$s(k_s, x) := x^{k_s}, \quad v(k_v, x, a) := 「a^{k_v} = x \text{ となるか否か}」$$

とすればよい。

公開鍵暗号として RSA が用いられている状況では、Alice は署名用の鍵 k_s を、暗号解読用の秘匿鍵 $k_{d,A}$ と共用し、署名検証用の公開鍵 k_v を、暗号化用公開鍵 $k_{e,A}$ と共通にして公開することができる。

この場合、Alice は x への署名 a は $a = x^{k_{d,A}}$ で計算する。Bob の暗号化 (公開) 鍵 $k_{e,B}$ で暗号化して Bob に送り、Bob は復号化 (秘匿) 鍵でそれを復号して (x, a) を得て x と $a^{k_{e,A}}$ の一致をみて、それが Alice の署名つきであることを確認する。