

教養原論 数理の世界
数学とコンピュータ 2011
RSA 暗号

野呂 正行

神戸大学大学院理学研究科数学専攻

November 18, 2010

暗号とは

- 暗号化

ある規則で, 文字を別の文字に変換

- 復号

暗号文をもとの文 (平文; ひらぶん) に戻す

- 単純なもの: 表を使う

例: アルファベットを単に何文字かずらす (シーザー暗号)

⇒ 文字の出現頻度を調べると解読できる

参考: 英文字の出現頻度 (COD; 数字は%)

E(11), A(8.5), R(7.6), I(7.5), O(7.2), T(7.0), N(6.7),
S(5.7), L(5.5), C(4.5), U(3.6), D(3.4), P(3.2), M(3.0),
H(3.0), G(2.5), B(2.1), F(1.8), Y(1.8), W(1.3), K(1.1),
V(1.0), X(0.3), Z(0.3), J(0.2), Q(0.2)

暗号方式

暗号は、一般にアルゴリズムにより暗号化, 復号
— 秘密保持のため, 鍵が必要

- 共通鍵暗号 (例: DES, AES)

暗号化, 復号に同じ鍵を使う

鍵は, 通信者 A, B が何らかの方法で共有する

⇒ 鍵の共有方法が問題

- 公開鍵暗号 (例: RSA, 楕円曲線暗号)

A から B にデータを送る場合, B が公開している鍵で暗号化してから送る

B は, 自分が持つ秘密鍵で復号する

⇒ 一時的に使う秘密鍵の送付に使える

例 : https://...

A が web shop B で物を買いたいとする
(B の URL は信用できるとする)

問題点 : インターネット上では常に盗聴の危険あり

- 初対面の相手と安全に通信するにはどうするか?
データは暗号化する必要がある
⇒ 自分と相手だけが知る鍵が必要
- 初対面で、どうやって 鍵を共有するか?
A が共通鍵を作り, B に渡せばよい
- 共通鍵を相手に安全に渡すにはどうすればよいか
B が公開鍵をつくり, 公開し, A は共通鍵を, その公開鍵で暗号化して渡す

公開鍵暗号の要件

- 公開鍵から秘密鍵がバレない
暗号化方法は当然公開されている (復号方法も)
公開鍵と秘密鍵は当然密接に関係している
⇒ 公開鍵から秘密鍵がバレないような方式が必要
- 暗号文から平文が簡単に復元できない
鍵がなくても平文が復元できるような意味がない
例：一文字ずつ変換したのでは, 頻度解析などで解読できる
⇒ 複数文字 (文字列) を変換できなければいけない

RSA 暗号

Rivest, Shamir, Adleman により考案された (1977)

前述の問題点を解消できる暗号方式の一つ

- 公開鍵 n は素数 p, q を 2 つかけたもの ($n = pq$)
例 : 1024 ビット RSA 暗号の公開鍵は, 150 桁程度の素数を 2 つかけたもの
 $0 \leq x \leq n - 1$ なる整数 x を, 同じ範囲の整数に変換, 逆変換できる.
- 文字列の変換
各文字を数に変換してつなげた大きい整数として変換
アルファベット (ASCII) : 1 バイト
漢字, かな : JIS なら 2 バイト, UTF-8 なら 3 バイト
以上

余談：文字コードについて

- ASCII

(いわゆる半角) アルファベット, 数字, 記号を 1 バイト (0-127) で表す.

- JIS コード (ISO-2022-JP)

漢字を ASCII 2 バイトで表すが, 通常の ASCII と混在するために, 表の切り替え (エスケープシーケンス) が必要.

- EUC-JP

JIS 2 バイトの各バイトの最上位ビットを 1 にしたもの. ASCII コードは最上位ビットが 0 なので, 切り替えなしに混在可能.

- UTF-8 (Macintosh で使用)

世界中のすべての文字を表現する表 (Unicode). ASCII は 1 バイト, その他の文字を 2 から 6 バイトで表す. 漢字は 3 バイト以上必要.

剰余計算の基本

- 合同式以下「 $A - B$ が C で割りきれる」を「 $A \equiv B \pmod{C}$ 」と書く. $\Rightarrow A \pmod{C} = B$ をより一般化

例: $10 \equiv 6 \pmod{4}$, $3 \equiv -2 \pmod{5}$

$B \pmod{C} = A$ なら $A \equiv B \pmod{C}$ である.

- フェルマー (Fermat) の小定理

a を整数, p を素数とすると $a^p \equiv a \pmod{p}$.

さらに, a が p で割りきれなければ $a^{p-1} \equiv 1 \pmod{p}$.

- 余りの計算は, 部分的に行ってよい

$$a \pm b \equiv (a \pmod{n}) \pm (b \pmod{n}) \pmod{n}$$

$$ab \equiv (a \pmod{n})(b \pmod{n}) \pmod{n}$$

$$a^k \equiv (a \pmod{n})^k \pmod{n}$$

RSA 暗号 : 公開鍵 (n, e)

- ① $e = 65537$ がよく用いられる.
- ② 大きい素数 p, q を用意する.
 p, q は同じくらいの大きさで, $p \neq q$
実用上 p, q は 10^{150} くらいで, $p - 1, q - 1$ が
 $e = 65537$ で割り切れないものを選ぶ.
- ③ $n = pq$ に対し, (n, e) が公開鍵

RSA 暗号 : 秘密鍵 d

e は $n' = (p-1)(q-1)$ を割り切らない素数なので,
 $GCD(e, n') = 1$

$\Rightarrow de \equiv 1 \pmod{n'}$ を満たす $d > 0$ がある

この $d > 0$ を秘密鍵とする.

- ① F_e での n' の逆数を k とする

$$kn' \equiv 1 \pmod{e}$$

- ② $d = \frac{1-kn'}{e}$ とおく

$$de = 1 - kn' \text{ なので } de \equiv 1 \pmod{n'}$$

- ③ d にいくつか n' を足せば $d > 0$

$$(d + mn')e \equiv de \pmod{n'} \text{ なので,}$$

$$(d + mn') \equiv 1 \pmod{n'}$$

RSA 暗号 : 安全性

- 鍵の安全性

d を作るには $n' = (p - 1)(q - 1)$ が必要

⇒ p, q が必要

⇒ n の素因数分解が必要

⇒ $n \simeq 10^{300}$ では事実上無理

(最近では ⇒ $n \simeq 10^{600}$ が推奨されている)

すなわち, 「数学者, 計算機科学者, コンピュータの現在の能力では素因数分解ができない」ことに基づく暗号方式が RSA 暗号

量子コンピュータが実用化されると危うい?

- e の選び方

e はある程度大きい素数ならなんでもよい

RSA 暗号 : 暗号化 ($E = A^e \bmod n$)

① データを数字の列に変換

例えば, UTF-8 の 3 バイト文字 (漢字など) の列なら

$$\boxed{a_1} \quad \boxed{a_2} \quad \cdots \quad \boxed{a_l} \quad 0 \leq a_i \leq 2^{24} - 1 \text{ (3 バイト)}$$

② 1. で作った数の列を, n 未満の数に分割する.

$$\boxed{a_1} \quad \cdots \quad \boxed{a_m} \quad \boxed{a_{m+1}} \quad \cdots \quad \boxed{a_{2m}} \quad \cdots$$

3 バイトが m 文字分 \Rightarrow 2 進 2^{24m} 桁

$\Rightarrow 2^{24m} \leq n$ となるように m をとればよい.

$$A_1 = \boxed{a_1} \quad \cdots \quad \boxed{a_m}, \quad A_2 = \boxed{a_{m+1}} \quad \cdots \quad \boxed{a_{2m}}, \quad \cdots \text{ とおく.}$$

すなわち,

$$A_1 = a_1 \cdot 2^{24(m-1)} + \cdots + a_{m-1} \cdot 2^{24} + a_m, \dots$$

と見なす. このとき $0 \leq A_i \leq n - 1$ である.

③ 各 A_i を暗号化 : $E_i \leftarrow A_i^e \bmod n$

RSA 暗号 : 復号 ($D = E^d \bmod n$)

① 各 E_i に対し $D_i \leftarrow E_i^d \bmod n$
実は, $D_i = A_i$ となっている.

② 数字の列を元に戻す

作った D_i を, 3 バイトずつに区切り, 文字に戻す

注意: E_i から A_i を求めるのは困難.

e 乗して $\bmod n$ したら E_i になる数は何か? ($\bmod n$ での e 乗根)

$\Rightarrow 0, 1, \dots, n-1$ を e 乗して $\bmod n$ すれば求まるが, n が巨大だと無理

簡単な例

$n = 1223460961843067$, $e = 65537$, $d = 115276424115473$

暗号文 : 139187734536180 を復号してみる.

$$139187734536180^d \equiv 253530592481414 \pmod{n}$$

$$253530592481414 = (\text{e695b0e79086})_{16}$$

実は $(\text{e695b0})_{16}$ は「数」, $(\text{e79086})_{16}$ は「理」 (UTF-8 で)

実際, 「数理」を暗号化してみると

$$253530592481414^e \equiv 139187734536180 \pmod{n}$$

と, 上で与えた暗号文になることがわかる.

数学的な証明が必要な部分

$E = (A^e \bmod n)$, $D = (E^d \bmod n)$, $0 \leq A, D \leq n - 1$ ならば
 $D = A$

これは $A^{ed} \equiv A \pmod n$ を示せばよい

基本的事実

- ab が p で割り切れるなら a が p で割り切れる。
これが p が「素」であることの基本的性質
- a が $n(= pq)$ で割り切れる $\Leftrightarrow a$ が p でも q でも割り切れる
- p が素数なら $a^p \equiv a \pmod p$ (フェルマーの小定理)

$A^e \bmod n, E^d \bmod n$ の計算は大変そう

⇒ 繰り返し 2 乗法 (repeated squaring)

剰余計算の基本原理 $(ab) \bmod n = (a \bmod n)(b \bmod n) \bmod n$

と指数法則 $a^{u+v} = a^u \cdot a^v$

を繰り返し使う.

- ① $k = (k_m k_{m-1} \cdots k_0)_2$ と 2 進で書く
- ② $a_i = a^{2^i} \bmod n$ ($i = 0, 1, \dots, m$) を計算する
 $a^{2^{i+1}} = a^{2^i + 2^i} = a^{2^i} \cdot a^{2^i}$ より $a_{i+1} = a_i^2 \bmod n$
- ③ $k_i = 1$ である i に対する a_i を全部かけて $\bmod n$ する
例: $k = 53 = 2^5 + 2^4 + 2^2 + 2^0 = (110101)_2$ なら,
 $a^k = a^{2^5} \cdot a^{2^4} \cdot a^{2^2} \cdot a^{2^0}$ より
 $a^k \bmod n = (a_5 \cdot a_4 \cdot a_2 \cdot a_0) \bmod n$

例: $12^{100} \bmod 15$ の計算

$$100 = 2^6 + 2^5 + 2^2 = (1100100)_2$$

$$a_0 = 12 \bmod 15 = 12$$

$$a_1 = 12^2 \bmod 15 = 9$$

$$a_2 = 9^2 \bmod 15 = 6$$

$$a_3 = 6^2 \bmod 15 = 6$$

$$a_4 = 6^2 \bmod 15 = 6$$

$$a_5 = 6^2 \bmod 15 = 6$$

$$a_6 = 6^2 \bmod 15 = 6$$

$$8^{100} \bmod 15 = (6 \cdot 6 \cdot 6) \bmod 15$$

$$= ((6 \cdot 6) \bmod 15 \cdot 6) \bmod 15 = (6 \cdot 6) \bmod 15 = 6$$

素数 p, q の作り方

- 整数をランダムに発生させて, 素数かどうかチェック
- 素数チェック
確定的ではないが, 通ればほとんど間違いなく素数, というチェック方法がある
(実用上, 「ほとんど間違いなく素数」で問題ない, とされる)
- 整数をランダムに発生させる
より乱数発生アルゴリズムがいろいろある

フェルマーの小定理の証明

a は p で割り切れないとする. $x \bmod p$ を \bar{x} と書く.

1. $1 \leq i, j \leq p-1$ のとき, $i \neq j$ なら $\bar{ai} \neq \bar{aj}$

もし $\bar{ai} = \bar{aj}$ なら $a(i-j)$ が p で割り切れる. a が p で割り切れないので $i-j$ が p で割り切れる. これは矛盾.

2. $\{\overline{a \cdot 1}, \dots, \overline{a \cdot (p-1)}\} = \{1, \dots, p-1\}$ (集合として等しい)

1. より $\overline{a \cdot i}$ はすべて異なり, 0 と異なるので OK.

3. $a^{p-1} \cdot (p-1)! \equiv (p-1)! \pmod{p}$

2. の両辺の要素をそれぞれかけた.

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

3. より $(a^{p-1} - 1)(p-1)!$ が p で割り切れる. $(p-1)!$ は p で割り切れないので $a^{p-1} - 1$ の方が p で割り切れる.

$$A^{de} \equiv A \pmod{n}$$

$de \equiv 1 \pmod{n'}$ より $de - kn' = 1$ を満たす k がある.

$A^{de} = A^{1+kn'}$ より $A^{1+kn'} \equiv A \pmod{n}$ をいえばよい.

$n = pq$ より次をいえばよい:

$$A^{1+kn'} \equiv A \pmod{p}, \quad A^{1+kn'} \equiv A \pmod{q}$$

① $A \equiv 0 \pmod{p}$ のとき

$$A^{1+kn'} \equiv 0 \pmod{p} \text{ より } A^{1+kn'} \equiv A \pmod{p}$$

② $A \equiv 0 \pmod{p}$ でないとき (A が p で割り切れないとき)

$$A^{1+kn'} = A \cdot A^{k(p-1)(q-1)} = A \cdot (A^{p-1})^{k(q-1)} \text{ で,}$$

A が p で割り切れないから

$$A^{p-1} \equiv 1 \pmod{p} \text{ (フェルマーの小定理).}$$

$$\text{よって, } A^{1+kn'} \equiv A \pmod{p}.$$

(\pmod{q} も同様)