暗号入門

かがみ

2001 年 10 月 26 日

目次

1 暗号の種類

この節では「暗号」という言葉を常識的な意味で使用します。

1.1 暗号の方式

暗号の方式は、通常次の 3 つの種類があります。

- アルゴリズム秘匿方式
- 共通鍵暗号 (対称暗号)
- 公開鍵暗号 (非対称暗号)

1.2 アルゴリズム秘匿方式

アルゴリズム秘匿方式は、その名の通りで、暗号化の手順を秘密にする方法です。次の利点と欠点があります。

- 利点 — ばれないと以外と強力。
- 欠点 — きっとばれてしまう。
- 本当の欠点 — 暗号は、「これだけの強度がある」という根拠が理論的に証明できなければ使い物にならないというのが現在の認識。 アルゴリズムを秘匿しては客観的な強度を判定できない。

実質的には例えば私が「P 君の悪口」をファイルに書いて、それを圧縮して各バイトを 3bit ローテーションすれば、きっとばれません。こんなファイルを手間かけて読もうとはだれも思わないからです。圧縮するのは頻度類推を避けるためです。

だけど、本気で読もうとすればすぐにばれてしまいます。

2 共通鍵方式

ここでいやな顔をする人が多いと思いますが、ちょっと記号を導入します。

- 暗号前の文 — D。 これは普通のテキストと思って下さい。
- 暗号後の文 — E。 これはぐちゃぐちゃで内容が分からないテキストと思って下さい。
- 暗号の鍵 — K。 これは通常のパスワード程度の文字列と思って下さい。

共通鍵方式の暗号は次の特徴を持ちます。

- 基本的にアルゴリズムは公開されている。
- 鍵 K を使用して暗号化する。キーにより同じ D から異なる E が生成される。
- K を知らないと E から D を知るのは非常に難しい。K を知っていれば—少なくともコンピュータを使えば— E から D を復元するのは容易。

DES が一番有名です。

利点と欠点は次の通りです。

- 利点 — 鍵を知っていれば暗号化、復号化が高速に行える場合が多い。 また、ビット置換アルゴリズムが多いのでハードウエア化しやすい。
- 利点 — 少なくとも DES 等は、十分鍵を長くすれば「鍵を知らないで」解読するのが難しいと信じられている。
- 欠点 — 暗号化する側、復号化する側が共通の鍵 (K) を知っている必要がある。 じゃあ、その鍵をどうやって送るのかが最大の問題になる。この問題に関しては、公開鍵方式の場所で考えます。

3 公開鍵方式

公開鍵方式の暗号は—本当は意味が違うんですが—暗号化する人 (送信者) と復号化する人 (受信者) が別の鍵を使います。また、ひんしゅくものですが、もう一度記号を導入します。

- 暗号前の文 — D。
- 暗号後の文 — E。
- 暗号の秘密鍵 — K。

- 暗号の公開鍵 — P。

送信者、受信者は次の情報を「知っています」

- 送信者
  - P
- 受信者
  - K

次の手順で通信が行われます。

- 通常受信者は P を一般に公開します。
- 送信者は P を公開した受信者に対して D(平分) を P で暗号化して E(暗号文) として送信します。
- 受信者は E を受信し K を利用して D に変換します。 K を知っていると簡単な計算で D が分かる様な仕組みになっているのです。逆に K を知らないと、D を復元するのは非常に難しいと考えられる方法を使用します。

公開キー方式の利点はなんでしょうか。例えば誰かがインターネット上でお店を開いて、クレジットカード番号を入力してもらう状況を考えてみます。 クレジットカードの番号はインターネット上で暗号化され、他人に理解できない様にする必要があると仮定します。
まず、共通鍵暗号を使用した場合の状況を考えます。

- 全部のお客さんに個別の共通鍵を与える必要あり。
- 共通鍵は郵送、 FAX、電話 等でやりとりする必要がある。郵送、FAX、電話でさえも、最近は国家の介入が心配である。
- そもそも、鍵を郵送等で送るくらいなら、最初からクレジットカード番号を送ればいいんじゃないか。

次に、公開鍵方式を使用した場合の状況を考えます。

- お店は公開鍵 P をインターネット上に公開する。
- 客はクレジットカード番号を P で暗号化して送信する。
- お店は秘密鍵 K(これがばれたら大変) でクレジットカード番号を復元する。

というわけで、非常に運用が良くなります。実際にはお店の様なシステムで P、K を固定するのは危険ですので、 P、K を動的に生成して通信を行います。

4 RSA

いままでは、いわゆる「お話し」であったので、本節では、最も有名な公開鍵暗号方式である RSA について少々真面目な話をします。

4.1 基本記号と概念

4.2 計算量

ここでは計算量の真面目な話は行いません。ただ、次の事実は認識して下さい。ここで現れる整数は数 100 桁程度と思って下さい。

- 整数 n の素因数分解は難しいと信じられている。
- 整数 n、m の最大公約数は簡単に求まる。
- 整数 p が素数かどうかを判定するのは難しいらしいが、「確率論的方法」を使用すれば、実際上問題が生じないレベルで高速に判定可能。

「難しい」とか「簡単」「高速」という意味は厳密に定義すると面倒なのでここでは常識的な意味と思って下さい。

4.2.1 剰余演算

今後登場する記号は基本的に自然数を表しているとします。また「自然数」は 0 を含むとします。

n、m、p を自然数とします。$p \geq 2$ の場合

$$n \bmod p \tag{1}$$

を n を p で割った余りとします。 文脈上 p で割った余りを考えていることが明白な場合、単に

$$n \tag{2}$$

と書く場合もあります。

m を p で割った余りと n を p で割った余りが等しい場合

$$n = m \bmod p \tag{3}$$

と記述します。文脈上 p で割った余りを考えていることが明白な場合、単に

$$n = m \tag{4}$$

と書く場合もあります。

mod は次の著しい性質を持ちます。

$$\begin{gathered} a = b \bmod p、c = d \bmod p \quad \text{が成立するとき} \\ a + c = b + d \bmod p \\ ac = bd \bmod p \end{gathered} \tag{5}$$

4.2.2 $px + qy = 1$

一般に n、m を自然数として、その最大公約数を d とします。 このとき

$$nx + my = d \quad \text{を満たす整数} x, y \text{が存在する} \tag{6}$$

という事実が成立します。特に p と q を互いに素な自然数とすると

$$px + qy = 1 \quad \text{を満たす整数} x, y \text{が存在する} \tag{7}$$

また次の事実が成立します。

$$px + qy = 1 \quad \text{を満たす整数} x, y \text{は具体的、高速に計算できる} \tag{8}$$

4.2.3 Euler の関数 φ

Euler(オイラーと読む) の関数 φ(ファイと読む) は次の様に定義されます。

$$\varphi(n) \text{は} n \text{と互いに素な} n \text{より小さい自然数の数} \quad n \geq 1 \tag{9}$$

次の性質が成り立つことが知られています。

$$\begin{array}{cc} p \text{が素数の場合} & \varphi(p) = p - 1 \\ n, m \text{が互いに素な場合} & \varphi(nm) = \varphi(n)\varphi(m) \\ \text{特に} p, q \text{が相異なる素数の場合} & \varphi(pq) = (p-1)(q-1) \end{array} \tag{10}$$

4.2.4 Euler の定理

今回使用する Euler の定理は—Euler の定理というのは山ほど存在するのです—次の定理です。

$$n, p \text{が互いに素の場合} \quad n^{\varphi(p)} = 1 \bmod p \tag{11}$$

これで準備完了です。

4.3 RSA の仕組み

今後任意の文は「自然数」で表現可能とします。例えば文を適当なコードでのコンピューターのビット列と見なしてそれを巨大な整数と思えば良いのです。

公開暗号では受信者が秘密鍵と公開鍵を作り公開鍵を公開します。

秘密鍵は次の手順で作成します。

- 巨大な相異なる素数 (現在の技術では数 100 桁) p, q を選ぶ。
- これを絶対にばれない様に管理する。

公開鍵は次の手順で作成、公開します。

- $(p-1)(q-1)$ と互いに素な整数 e を選択する。
- pq, e を公開する。

送信者は平文 D から暗号文 E を次の計算式で作成します。 送信者は pq, e の値は知っているので

$$E = D^e \bmod pq \tag{12}$$

ただし、平文 D は p,q いずれよりも小さいとします。長文の場合 D を p,q いずれよりも小さい部分に分けて暗号化するのでこの仮定は問題ありません。p, q が素数であるので D と pq は互いに素です。

さて E を受け取った受信者はどうするかといいますと、次の手順で E から D を知ることが可能です。受信者は e はもちろんのこと p, q の値も知っています。

- 次の解をもとめる。

$$em + (p-1)(q-1)n = 1 \quad \text{を満たす整数} m \geq 0, n \leq 0 \tag{13}$$

これは、もともと e を $(p-1)(q-1)$ と互いに素な様に選んであるので可能です。 また計算も高速にできます。

- 次の計算を行う。

$$E^m \bmod pq \tag{14}$$

なんと最後の計算値は D となります。なぜかといいますと。

$$E^m = D^{em} = D^{1-(p-1)(q-1)n} = D(D^{-n})^{(p-1)(q-1)} \bmod pq \tag{15}$$

ところが

$$\varphi(pq) = (p-1)(q-1) \tag{16}$$

であり pq と D は互いに素なので pq と D^{-n} も互いに素。従って Euler の定理により

$$(D^{-n})^{(p-1)(q-1)} = (D^{-n})^{\varphi(pq)} = 1 \bmod pq \tag{17}$$

従って

$$E^m = D \bmod pq \tag{18}$$

4.4 RSA の簡単な考察

役に立つ暗号であるからには次の条件が満たされる必要があります。

- 鍵を知っている場合簡単に暗号化、復号化が容易に行える。
- 鍵を知らない場合暗号化、復号化が実質上不可能。

まず心配になるのは、一般に D, n, m が非常に大きい (数 100 とか数 1000 桁) の場合

$$D^n \bmod m \tag{19}$$

が高速に計算できるかということです。真面目に D を n 回かけ算していたらお話になりません。m での余りを毎回もとめて数値の増大を防いでもだめです。なんったって n は、例えば「桁数」が 500 なんんですから。これでは宇宙が終わるまで待っても答えはでません。

これはみんなが知っていると良いと思うのですが。べき乗を計算する場合次のアルゴリズムを使用します。7ページに C っぽく書いた例を載せますが (int はいくら大きくても良いことにします) n はどんどん半分になるので 500 桁の n に対して $\log_2 10^{500}$ で 1500 回程度のループで計算完了です。

これらの事実により、送信者も受信者も「鍵の知識」があれば高速な計算が可能であることがわかります。

暗号がばれないかどうかに関しては pq — これは公開されている — を知っている場合に p, q を知ることが簡単かどうかにかかっています。誰も証明した人はいませんが、次の事実が信じられています。

一般の素因数分解問題は計算困難

これが RSA 暗号の強度のよりどころとなっています。有用な暗号で、強度が本当に「証明」されたものを私は知りません。 ある仮定のもとでの強度が証明されているだけの様です。

```
#define odd(n) ((n)&0x01)  /* n が奇数かどうか */
/*
 * D の n 乗を求める
 */
int power(int D, int n)
{
    int pow;

    pow=1;
    while (n!=0) {
        if (odd(n)) {  /* n が奇数 */
            pow*=D;    /* D 倍して */
            n--;       /* 1 回分減らす */
        }
        pow=pow*pow;   /* pow を 2 乗 */
        n/=2;          /* n は一気に半分になる */
   }
   return(pow);
}
```