第 2 章

暗号と数学

76 回生 Chito

2.1　はじめに

はじめまして, 76 回生の Chito です。いつの間にか高校生になっていました。

コロナ禍の中オンラインサービスを利用することが多くなった人もいると思いますが, 今回は, 私たちが利用するオンラインサービスの安全に利用するために使われている暗号化, 特に最も有名であろう「RSA 暗号」と, 好き嫌いの分かれる「数学」との関連性について話そうと思います。

2.2　数学に守られた世界

今, 私たちはネット上の様々なサービスを利用しています。例えば「LINE」や「Twitter」「Facebook」など, 様々なサービスがあります。

そして, こうしたサービスのセキュリティを維持するために, 数学が大いに役立っています。数学が好きな方もそうでない方もぜひ読んでみてください。

2.3　暗号化

暗号とは, ある情報を特定の決まった人しか読めないように一定の手順に基づいて無意味な文字や符号の列に置き換えたもので, 情報の伝送や記録, 保存の際, 第三者に盗み見られたり改竄されないようにするために作成されます。

暗号化とは

暗号化とは, データの内容を他人には分からなくするための方法で, 暗号化通信[*1]や電子署名[*2]などに利用されています。

用語一覧

平文

暗号化する前のもの

復号

暗号化されたデータを元の状態に戻すこと

[*1] 通信内容を暗号化して第三者による盗聴や改竄をされにくくした通信方式
URL が「https://~」から始まるものは暗号化通信が使われている

[*2] 電磁的記録（電子文書）に付与する, 電子的な徴証であり, 紙文書における印章やサイン（署名）に相当する役割をはたすもの

鍵

データを暗号化や復号する時に使うもの

暗号化と復号

ある暗号化するシステムがあり, そのシステムに元のデータ (**平文**) と**鍵**を入力することで, 出力として暗号化されたデータを得ることを暗号化と言います。

わかりやすく言うと, 元のデータ (平文) と鍵を入れると, それらから暗号文を作って表示する「不思議な箱」があり, その箱に元のデータ (平文) と鍵を入れて暗号化されたデータを取得する行為のことです。

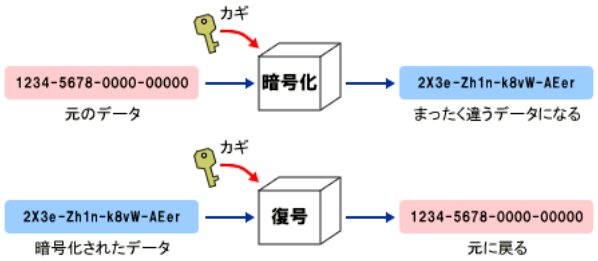

図 2.1: 暗号化と復号 (総務省 HP)

元のデータ (平文) が同じでも鍵が異なると異なった暗号文を出力します。

逆に, 暗号化されたデータから鍵を用いて元のデータ (平文) を得る操作のことを**復号**と言います。

暗号化する時も復号する時も**鍵**は非常に大事な役割を果たし, もし鍵が第三者の手に渡ってしまうと, 暗号が破られ, 元のデータを盗み見されるかもしれません。

その為, **鍵**を第三者の手に渡らないように厳重に管理しなければなりません。

もし, **鍵**が第三者の手に渡ってしまうと, 暗号が破られるかもしれず, そうすると重要な個人情報が盗まれる可能性があります。

また, 同じパスワードを使い回していたりすると, 他のサービスも悪用されたり, 銀行口座から預金を不正に引き出されるなど甚大な被害を被るかもしれません。

共通鍵暗号と公開鍵暗号

共通鍵暗号

共通鍵暗号とは暗号化と復号で同じ鍵を使用する暗号化の仕組みのことです。

共通鍵暗号は公開暗号と比べて, 処理が高速である一方, 暗号化の方法が既に知られている場合, 鍵さえ分かってしまえば誰でも復号でき, 平文を見ることができてしまうので, 鍵の受け渡しには十分注意する必要があります。

また, 暗号化する側と復号する側が同じ鍵を持つ必要があり, データのやり取りをする人数が増えるほど, 鍵が漏洩する確率が高まります。この問題を回避するためには, データの受け渡しの相手ごとに鍵を変えれば良いのですが, その場合やり取りする人数が増えるほど, 必要となる鍵の数は増えます。

n 人の間でデータのやり取りをする場合, データのやり取りをするペアごとに鍵が必要ですから, 全体で必要となる鍵の種類は n 人から 2 人を選ぶ組み合わせ, すなわち ${}_nC_2 = \frac{n(n-1)}{2}$ 種類となります。

たとえば, 4 人の間でデータをやり取りする場合は 6 種類, 5 人の間でデータをやり取りする場合は 15 種類の鍵が必要となります。

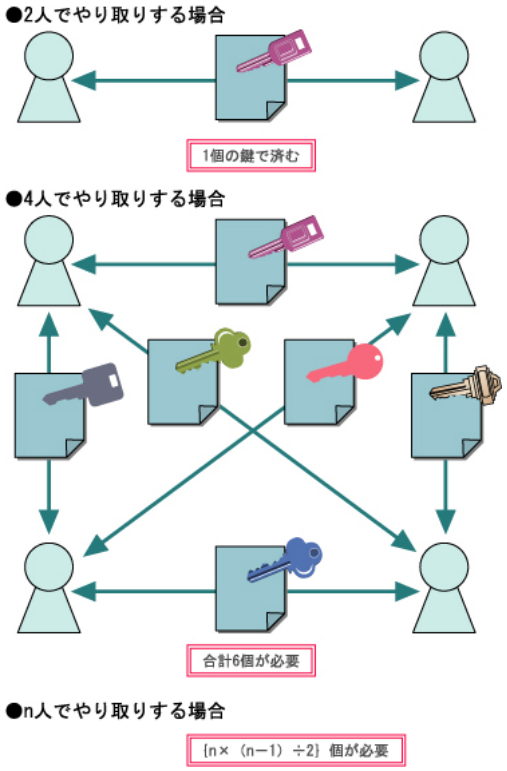

図 2.2: 共通鍵暗号の組み合わせ

公開鍵暗号

公開鍵暗号とは, 暗号化と復号で異なる鍵を用いる暗号方式のことです。

共通鍵暗号と違い, 鍵 (公開鍵) の配送を極秘に行う必要がなくなりました。一方, 異なる 2 種類の鍵を使って暗号化や復号を行うため, 複雑なアルゴリズムが必要となり, 共通鍵暗号と比べると処理にかかる時間が長くなります。

2.4 RSA 暗号の破り方

RSA 暗号

RSA 暗号は, 1977 年に発明された公開鍵暗号方式の一つで, 「フェルマーの小定理」に基づいて設計されています。

発明者である Ronald Linn **R**ivest, Adi **S**hamir, Leonard Max **A**dleman の頭文字をつなげてこのように呼ばれるようになりました。

それでは, 実際に RSA 暗号を破っていきたいと思います。

(注意:暗号の破り方がわかっても実際にあるサービスで試すと, 捕まる恐れがあるので十分お気をつけ下さい。)

用語一覧と表記上の注意

1. $a \times b$ は ab と表記する
2. a を n で割った余りと b を n で割ったまりが同じであることを $a \equiv b \pmod{p}$ と表記する
3. a と b の最大公約数を $lcm(a,b)$,a と b の最小公倍数を $gcm(a,b)$ と表記する
4. a を b で割った余りを $a \mod b$ と表記する

RSA 暗号の仕組み

1. 二つの異なる素数 p と q を用意する
2. $n = pq$ とおく (公開鍵その 1)
3. $\lambda(n) = lcm(p-1, q-1)$ とおく[*3]
4. $1 < e < \lambda(n)$ かつ e と $\lambda(n)$ が互いに素 ($gcd(e, \lambda(n)) = 1$) となるような正の整数 e(公開鍵その 2) を見つける
   $65537(= 2^{16} + 1)$ が使われることがよくある。
5. $de \equiv 1 \pmod{\lambda(n)}$ となる整数 d(秘密鍵) を見つける。

暗号化する前の文 (平文) を M, 暗号化した後の文 (暗号文) を C とおく

暗号化

$C = M^e \mod n$(M^e を n で割った余り)

復号

$M = C^d \mod n$

フェルマーの小定理

p が素数で a が p の倍数でない正の整数のとき,
$a^{p-1} \equiv 1 \pmod{p}$
が成り立つ

証明

$M^{ed} \equiv 1 \pmod{n}$ を証明すればよい。
$M^{ed} \equiv 1 \pmod{p}$...$(*)$ を証明する。
M が p の倍数のとき, 両辺は p の倍数であるから,(*) は成り立つ
M が p の倍数でないとき,$ed \equiv 1 \pmod{\lambda)(n)}$ から $ed-1$ はλ(n) の倍数で $\lambda(n) = lcm(p-1, q-1)$ であるから, $ed-1$ は $p-1$ の倍数である。よって,$ed - 1 = (p-1)N$ となる整数 N が存在し,$ed = 1 + (p-1)N$ と表される。
よって,$M^{ed} = M^{1+(p-1)N} = M\dot{M}^{(p-1)N} = M\dot{(}M^{p-1})^N$
このとき M は p の倍数でないから, **フェルマーの小定理**より $M^{p-1} \equiv 1 \pmod{p}$
よって $M\dot{(}M^{p-1})^N \equiv M \equiv M\dot{1}^N = M$
同様に,$M^{ed} \equiv M \pmod{q}$
よって $M^{ed} - M$ は p の倍数かつ q の倍数であり p, q は互いに素であるから,$M^{ed} - M$ は $pq = n$ の倍数である
従って,$M^{ed} \equiv M \pmod{n}$ である

例

$p = 11, q = 13$, 平文 $M = 8$ とする。

[*3] $\lambda(n) = \phi(n)$ とする方法もある。$\phi(x)$ はオイラー関数と呼ばれている。

1. $p = 11, q = 13$
2. $n = pq = 11 * 13 = 143$
3. $\lambda(n) = lcm(p-1, q-1) = lcm(10, 12) = 60$
4. $e = 17$ とすると,$1 < e < \lambda(n) = 60$ かつ e と $\lambda(n)$ が互いに素であるから条件を満たす。
5. $d = 53$ のとき $de = 17e \equiv 1 \pmod{\lambda(n)}$ を満たす。・・・(d の見つけ方については, 後の「ユークリッドの互除法と一次不定方程式の整数解」のところで詳しく触れます。)
6. このとき暗号文を C とおくと,C は $M^e = 8^{17}$ を n で割った余りで,$C = 112$。また,M は $C^d = 112^{53}$ を $n = 143$ で割った余りであるから,$M = 8$ となりちゃんと復号されたことが確認できる。

ユークリッドの互除法と一次不定方程式の整数解

ユークリッドの互除法

ユークリッドの互除法とは 2 つの整数の最大公約数を求めるアルゴリズムの一つで, 下記のような操作を繰り返すことで,2 つの整数の最大公約数を求めることができます。

2 つの自然数 $a, b(a \leqq b)$ について,a を b で割った余りを r とすると, a と b との最大公約数は,b と r との最大公約数に等しいという性質が成り立つ。この性質を利用して,b を r で割った余り (r' とおく), 割る数 r をその余り (r') で割った余り....., と剰余を求める計算を余りが 0 になるまで繰り返すと, 余りが 0 になった時の割る数が a と b との最大公約数となる。

非常に有名で, 中学受験算数でも出てくることがよくあるかもしれません。また, 下のような図で表されることもよくあります。

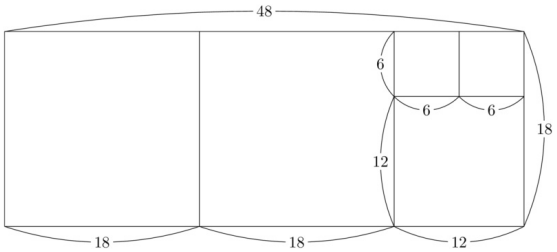

図 2.3: ユークリッドの互除法

■例 81249 と 73073 の最大公約数を求める

$$
\begin{aligned}
81249 &= 73073 * 1 + 8176 \\
73073 &= 8176 * 8 + 7665 \\
8176 &= 7665 * 1 + 511 \\
7665 &= 511 * 15 + 0
\end{aligned}
$$

よって,81249 と 73073 の最大公約数は 511

一次不定方程式の整数解の求め方

前の例の「$de \equiv 1 \pmod{\lambda(n)}$ を満たす d」はどのようにして見つけたのだろうか。前の例では $e = 17, \lambda(n) = 60$ であったから $17d \equiv 1 \pmod{60}$。よって,$17d - 1 = 60x$ すなわち,$17d - 60x = 1$ となる整数 x が存在する。ここで,17 と 60 の最大公約数をユークリッドの互除法を使って求めると下のようになります。

$$60 = 17 \times 3 + 9 \tag{2.1}$$
$$17 = 9 \times 1 + 8 \tag{2.2}$$
$$9 = 8 \times 1 + 1 \tag{2.3}$$
$$8 = 1 \times 8 \tag{2.4}$$

(2.3) から,$9 - 8 \times 1 = 1 \cdots (*)$

(2.2) から,$8 = 17 - 9 \times 1$ であるから, これを上の式 $(*)$ に代入して $9 - (17 - 9 \times 1) \times 1 = 9 \times 2 - 17 = 1 \cdots (**)$

(2.1) から,$9 = 60 - 17 \times 3$ であるから, これを $(**)$ の式に代入して, $(60 - 17 \times 3) \times 2 - 17 = 60 \times 2 - 17 \times 7 = 17 \times (-7) - 60 \times (-2) = 1 \cdots$

このようにしてユークリッドの互除法の手順を逆にたどることで一不定方程式の整数解を得ることができます。また, 次のようにすることで一般解を求めることができます。

$17d - 60x = 1$

$17 \times (-7) - 60 \times (-2) = 1$

辺々引いて,$17(d + 7) - 60(x + 2) = 0$ すなわち,$17(d + 7) = 60(x + 2)$, よって,$17(d + 7)$ は 60 の倍数で 60 と 17 は互いに素より $d + 7$ は 60 の倍数であるから,$d + 7 = 60k$ となる整数 k が存在し,$d = 60k - 7$ と表される。

$k = 1$ を代入することで例のように $d = 53$ を得ることができます。

RSA 暗号の安全性

この暗号方式は, 現在のコンピュータでは巨大な数の素因数分解を常識的な時間内で行うことは不可能であることを利用して, 安全性を確保しています。逆に言えば, もし比較的短い時間で巨大な数の素因数分解を行うことができるのであれば, この暗号方式の安全性が崩れることになります。そして, 現在開発途中の,「量子コンピューター」は素因数分解を高速で行うことが得意なので, 今後, 量子コンピューターが世界にどのような影響を与えていくのか注目していきたいと思います。

RSA 暗号の破り方

では, 解読が難しいとされている RSA 暗号をどのようにして解読するのでしょうか？ 具体的な方法を 2 つ紹介します。

その 1

公開鍵 n を頑張って素因数分解して,p, q を求める。

暗号文 M と p, q が分かるとと上記の手順で平文を求めることができる。

その 2

Common Modulus Attack

公開鍵 $(n, e_1), (n, e_2)$ の組と平文 M をそれぞれの公開鍵で暗号化した暗号文 C_1, C_2 があり, e_1, e_2 の最大公約数が 1 のとき与えられたデータから平文 M を求めることができる。

■手順 $e_1 s_1 + e_2 s_2 = 1$ となる正の整数 s_1, s_2 を見つけたとすると $c_1^{s_1} c_2^{s_2}$ を n で割った余りで M が求められる。

■証明 $c_1 \equiv M^{e_1} \pmod{n}$

$c_2 \equiv M^{e_2} \pmod{n}$

ここで,$e_1 s_1 + e_2 s_2 = 1$ となるような整数 s_1, s_2 を見つけたとすると

$$
\begin{aligned}
c_1^{s_1} c_2^{s_2} &\equiv (M^{e_1})^{s_1} (M^{e_2})^{s_2} \pmod{n} \\
&\equiv M^{e_1 s_1} M^{e_2 s_2} \pmod{n} \\
&\equiv M^{e_1 s_1 + e_2 s_2} \pmod{n} \\
&= M^1 \\
&= M
\end{aligned}
$$

2.5 最後に

余りの計算で暇つぶし

夏休み、憧れの N 先輩と一緒に彼女の親戚の家がある長野に向かうことになった主人公の K
これは新幹線の中での話.....
K「N 先輩、誕生日はいつですか？」
N 先輩「1992 年 7 月 19 日だよ」
K「1992 年 7 月 19 日だから、え〜と、日曜日ですね!合ってますか？」
N 先輩「う〜ん、わかんない」

誕生日の曜日を求める「ツェラーの公式」というものがあり, その公式は今回使った「余りによる演算 (mod)」が使われています。

上の会話の元ネタが知りたい方は金曜ロードショーでそのうちやると思います。そのアニメでは RSA 暗号も出てくるので RSA 暗号の仕組みを知っていると, より楽しむことができると思います。

では最後に,

「**お願いしまぁーーーーーーーす!!!(Enter キー, カッターン)**」

2.6 参考文献及び参考サイト

- https://qiita.com/tcb78/items/3eaa4a222bd544012db5
- https://en.wikipedia.org/wiki/RSA_(cryptosystem)
- https://manabitimes.jp/math/680
- https://manabitimes.jp/math/1146
- https://elliptic-shiho.hatenablog.com/entry/2015/12/14/043745