

# CTF 入門 CRC32 で 誤りを 検知する

@kurenaif

## 理論編

わからなくても大丈夫！

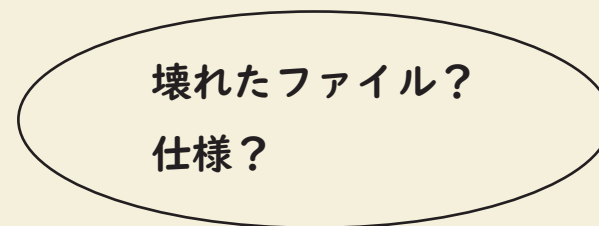
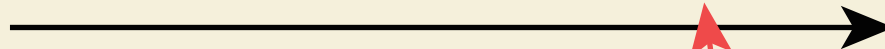
CRC32 って？

## Cyclic Redundancy Check

「誤りを検知する」検査方法



ファイル送信



壊れたファイル



何らかの外乱でファイルが破壊  
(ネット調子悪いとか)

## なぜ CRC32 を解説するの？

符号理論は勉強してないので、体系的に説明できない… すまない…

- 拡大体
- $GF(2^n)$  な多項式
- 「体」であることから得られる様々な恩恵
- 「線形性」が成り立つことからの様々な遊び

暗号で広く使われるこれらを使って、比較的簡単に応用ができる分野だから

## どこで CRC32 は使われているの？

### png ファイルとかに内蔵されている

```
IHDR Interlace: 0
IHDR Compression algorithm is Deflate
IHDR Filter method is type zero (None, Sub, Up, Average, Paeth)
IHDR Interlacing is disabled
Chunk CRC: -869110134
Chunk: Data Length 309 (max 2147483647), Type 1346585449 [iCCP]
Ancillary, public, PNG 1.2 compliant, unsafe to copy
Unknown chunk type
Chunk CRC: -960355186
Chunk: Data Length 65536 (max 2147483647), Type 1413563465 [IDAT]
Critical, public, PNG 1.2 compliant, unsafe to copy
IDAT contains image data
Chunk CRC: 1441794358
Chunk: Data Length 65536 (max 2147483647), Type 1413563465 [IDAT]
Critical, public, PNG 1.2 compliant, unsafe to copy
IDAT contains image data
Chunk CRC: -1856956801
Chunk: Data Length 65536 (max 2147483647), Type 1413563465 [IDAT]
Critical, public, PNG 1.2 compliant, unsafe to copy
IDAT contains image data
Chunk CRC: 126825411
Chunk: Data Length 19446 (max 2147483647), Type 1413563465 [IDAT]
Critical, public, PNG 1.2 compliant, unsafe to copy
IDAT contains image data
Chunk CRC: -425976308
Chunk: Data Length 0 (max 2147483647), Type 1145980233 [IEND]
Critical, public, PNG 1.2 compliant, unsafe to copy
IEND contains no data
Chunk CRC: -1371381630
```

今日の講義では

png ファイルからバイナリデータを  
抽出して、

この CRC を求めるところまでを  
目標にしています！

## CRC32 の理論を説明する道のり

- ・ 体とは？
- ・ 有限体とは？
- ・ 多項式で有限体を作ろう！
- ・ 多項式の体をコンピュータで扱ってみよう
- ・ CRC32 への応用

「体」とは？

直感的には、「足し算」、「引き算」、「掛け算」、「割り算」  
ができるもの。

ただし、いくつかの制約がある。

「引き算」と「割り算」は使わない。

「体」では、「足し算」と「掛け算」のみ行う。

「引き算」と「割り算」は、「逆元」を使って表現する

$$1 - 2 = 1 + (-2)$$

2 と足して 0 になる数

$$4 \div 2 = 4 \times \frac{1}{2}$$

2 とかけて 1 になる数

## 使える数（集合）の制限。

「体」では、まず使う数を制限する。

- ・ 整数
- ・ 有理数
- ・ 複素数
- ・ ベクトル
- ・ 行列
- etc...

使う範囲は自分で決めていい。

僕は「複素数のベクトル」にする！

私は「整数」だけ！

使って良い数たち全部のことを

「集合」という

制限したら、計算前も、計算後もとにかくその数以外は使用不可。

具体例は次のページで



使える数（集合）の制限。

整数だけの範囲で考えてみる。

$$1 + 2 = 3$$

$$1 \times 2 = 2$$

$$1 - 2 = 1 + (-2) = -1$$

$$4 \div 2 = 4 \times \frac{1}{2} = 2$$

分数は整数ではない！

アウト！

使える数（集合）の制限。

3次元ベクトルだけの範囲で考えてみる。

$$(1, 2, 3) + (2, 3, 4) = (3, 5, 7)$$

$$(1, 2, 3) - (2, 3, 4) = (-1, -1, -1)$$

$$(1, 2, 3) \cdot (2, 3, 4) = 20$$

掛け算を内積と割り当てると、20は3次元ベクトルじゃないので  
アウト！

（それぞれの要素の掛け算やわり算であれば、分数を認めればOKですね）

使える数（集合）の制限。

0 を含む正の有理数だけの範囲で考えてみる。

$$1 + 2 = 3$$

負の数は正の数ではない！

$$5 - 2 = 5 + (-2) = 3$$

アウト！

$$1 \times 2 = 2$$

$$4 \div 2 = 4 \times \frac{1}{2} = 2$$

使える数（集合）の制限。

有理数だけの範囲で考えてみる。

$$1 + 2 = 3$$

$$5 - 2 = 5 + (-2) = 3$$

$$1 \times 2 = 2$$

$$4 \div 2 = 4 \times \frac{1}{2} = 2$$

すべて有理数で表されているので ◎

0で割るみたいな異常なものを除いて、

選んだ範囲すべてでこのようにならない。

## 体とは

1. まず使える数の範囲を決める（「集合」を決める）
2. 足し算と掛け算を何にするか考える（ベクトルの内積は NG）
3. 引き算と割り算は、足し算と掛け算に直す。
4. どんなときでも、使えるのは集合の中の数だけ。  
（0 で割るみたいな例外を除いて、どんな計算をしても  
集合の中の数字に収まらないといけない。）  
（集合の中の数一個のことを「元」という）

厳密な定義は教科書とかでしっかり学んでね

## 有限体とは？

整数：無限個存在する。

有理数：無限個存在する。

複素数：無限個存在する。

0 を含む正の整数を 5 で割った値：5 個しか存在しない

0 を含む正の整数を  $N$  で割った値： $N$  個しか存在しない。

この有限個しかないやつらで、もし  
「体」を作ることができれば  
有限体になる。

引き算と割り算を考える。

引き算は、足して0になるような数に変換してから、  
足し算をする。

$$2 - 3 = 2 + \overset{\text{変換}}{\longrightarrow} (-3) = 1$$

これは3と足すと、0になる数。

割り算は、かけて1になるような数に変換してから、  
掛け算をする。

$$2 \div 3 = 2 \times \overset{\text{変換}}{\longrightarrow} \frac{1}{3} = \frac{2}{3}$$

これは3とかけると、1になる数。

正の整数 mod 5 は「体」にできるのか？

足し算と掛け算はできそう。

正の数だから負の数が存在しない。→ 引き算ができない。

整数だから分数は存在しない。→ 割り算ができない。

- ・ 引き算は、足して0になるような数に変換してから、足し算をする。
- ・ 割り算は、かけて1になるような数に変換してから、掛け算をする。

この方針で、引き算と割り算を考え直してみよう。



## 正の整数 mod 5 は「体」にできるのか？

余り 5 の世界での引き算

変換



例えば...

1 と足して 0 になるような数は 4

$$1 - 1 = 1 + 4 = 0 \pmod{5}$$

$$2 - 2 = 2 + 3 = 0 \pmod{5}$$

$$3 - 3 = 3 + 2 = 0 \pmod{5}$$

$$4 - 2 = 4 + 1 = 0 \pmod{5}$$

mod 5 の中のすべての数たちに対して、負の数的な数字が見つかった！

しかも全部 mod5 の中に入ってる！

## 正の整数 mod 5 は「体」にできるのか？

余り 5 の世界での引き算

例えば...

変換

2 とかけて 1 になるような数は 3

$$1 \div 1 = 1 \times 1 = 1 \pmod{5}$$

$$2 \div 2 = 2 \times 3 = 1 \pmod{5}$$

$$3 \div 3 = 3 \times 2 = 1 \pmod{5}$$

$$4 \div 4 = 4 \times 1 = 1 \pmod{5}$$

mod 5 の中のすべての数たちに対して、逆数的な数字が見つかった！

しかも全部 mod5 の中に入ってる！

正の整数 mod 5 は「体」にできるのか？

使える範囲の数だけを使って、  
「足し算」、「引き算」、「掛け算」、「割り算」  
ができたので、これは体！

しかも使える数が5つだけの有限個なので、  
有限体！

mod 素数 の世界であれば、有限体は作れる！

正の整数 mod 4 は「体」にできるのか？

2 と掛け算して、1 になる数字は存在しない。

$$2 * 1 = 2 \bmod 4$$

$$2 * 2 = 4 \bmod 4$$

$$2 * 3 = 2 \bmod 4$$

2 では割り算できないから、これは「体」ではない。

Q. 多項式って何？

A. こんなやつ↓

$$1 \times x^2 - 2 \times x + 1$$

x の  $n$  乗と、なにか**係数**がひっついてるやつ

係数で使える値を束縛する

$$x^2 - 1 = 0$$

この式解けますか？

係数で使える値を束縛する

$$x^2 - 1 = 0$$

$$(x + 1)(x - 1) = 0$$

$$x = 1, -1$$

係数で使える値を束縛する

$$x^2 + 1 = 0$$

この式解けますか？



## 係数で使える値を束縛する

複素数が使えなかったら  
この式変形はできない。

$$x^2 + 1 = 0$$

$$(x + i)(x - i) = 0$$

$$x = i, -i$$

複素数を使っていいテスト以外ではバツになりそう

使える係数の範囲を絞る。

明確に、複素数を使っていけませんと明記すると  
答えが変わる。

## 多項式の「足し算」、「掛け算」、「引き算」

多項式も、整数と同じように、足し算や掛け算、引き算ができる。

### 多項式の足し算

$$(x^2 + 2x + 1) + (x^3 + 2x^2 + 3) = x^3 + 2x^2 + 2x + 4$$

### 多項式の引き算

$$(x^2 + 2x + 1) - (x^3 + 2x^2 + 3) = -x^3 - x^2 - 2$$

### 多項式の掛け算

$$(x + 1)(x + 2) = x^2 + 3x + 2$$

## 多項式の「足し算」、「掛け算」、「引き算」

できないときもある

多項式も、整数と同じように、足し算や掛け算、引き算が~~できる~~。

### 多項式の足し算

$$(x^2 + 2x + 1) + (x^3 + 2x^2 + 3) = x^3 + 2x^2 + 2x + 4$$

### 多項式の引き算

もし係数に負の数が使えなければ  
ここで計算できなくなる

$$(x^2 + 2x + 1) - (x^3 + 2x^2 + 3) = -x^3 - x^2 - 2$$

### 多項式の掛け算

もし係数で掛け算ができなければ  
ここで計算ができなくなる

$$(x + 1)(x + 2) = x^2 + 3x + 2$$

## 多項式の「割り算」

(x+1) とかけて 1 になる多項式を  
求めなければならない。

$$\frac{1}{x+1} = a + bx + cx^2 + dx^3 + \dots$$

単項式の割り算を、なんとかしてこの形に落とし込まなければならない。

難しそう…

## 多項式の割り算の「あまり」

正の整数は引き算や割り算はできなかったが、  
「あまり」を導入することで「有限体」にできた。

実は、多項式でも「あまり」を使うことで、「体」に  
することができる。

## 多項式の割り算の「あまり」

$$\begin{array}{r} x + 2 \\ x^2 + x + 1 \overline{) x^3 + 3x^2 + 5x + 5} \\ \underline{x^3 + x^2 + x} \phantom{+ 5} \\ 2x^2 + 4x + 5 \\ \underline{2x^2 + 2x + 2} \\ 2x + 3 \end{array}$$

これが

$x^3 + 3x^2 + 5x + 5$  を

$x^2 + x + 1$  で割ったあまり。

## 多項式の世界の「体」

正の整数の世界では、「素数の」mod を取ることによって、「有限体」にすることができた。

実は、多項式でも「素数的な存在」であまりを求めることによって「体」を作ることができる。

## 多項式の世界の「素数」

掛け算で表すことができる。

$$49 = 7 \times 7$$

多項式の掛け算で表現できる

$$x^2 - 2x + 1 = (x - 1)^2$$

$$28 = 4 \times 7$$

$$x^2 - 1 = (x + 1)(x - 1)$$

7 → 掛け算で表現できない

$x^2 + 1$  → 虚数がない世界だと  
掛け算で表現できない。



## 多項式の世界の「素数」

掛け算で表すことができる。

$$49 = 7 \times 7$$

多項式の掛け算で表現できる

$$x^2 - 2x + 1 = (x - 1)^2$$

$$28 = 4 \times 7$$

$$x^2 - 1 = (x + 1)(x - 1)$$

多項式の世界の  
素数っぽい！

7  $\longrightarrow$  掛け算で表現できない

$$x^2 + 1 \longrightarrow$$

虚数がない世界だと  
掛け算で表現できない。

既約多項式という

## 多項式の世界で「体」を作る

1. は「体」でなければダメ。

1. 係数に使っていい数の種類を決める。(有理数、虚数、etc...)
2. 既約多項式(素数的存在)を決める。(1で決めた範囲内で。)
3. (1で決めた範囲内の)多項式を既約多項式でわった「あまり」たちの集合が、「体」になる！

ちゃんと証明するためにはもっと色々準備する必要があり、  
辛いので今日はやりません…  
教科書を読んで勉強してください！

## 多項式の「体」まとめ

一つが多項式

$$x^3 + 3x^2 + 2x + 4$$

対応



一つの数字

18

既約多項式（有理数の世界で）

$$(x^2 + 1)$$

対応



素数

7

あまり（有理数の世界で）

$$x + 1$$

対応



あまり

4

## 多項式の「体」まとめ

一つが多項式

$$x^3 + 3x^2 + 2x + 4$$

対応



一つの数字

18

既約多項式（有理数の世界で）

$$(x^2 + 1)$$



素数

7

あまり（有理数の世界で）

$$x + 1$$



あまり

4

↑  
係数に使える数が

無限個存在するので、この多項式のあまりも「有限体」  
ではない。

## 多項式で「有限体」を作る

多項式のあまりで有限体を作るためには、  
係数に使っていい数字も有限体にしなければならない。

→ おもいきって、係数を **mod 2 の世界**にしてみよう！

## mod 2 の世界の四則演算

### 足し算 (XOR)

$$0+0 = 0 \text{ mod } 2$$

$$0+1 = 1 \text{ mod } 2$$

$$1+0 = 1 \text{ mod } 2$$

$$1+1 = 0 \text{ mod } 2$$

### 掛け算 (AND)

$$0*0 = 0 \text{ mod } 2$$

$$1*0 = 0 \text{ mod } 2$$

$$0*1 = 0 \text{ mod } 2$$

$$1*1 = 1 \text{ mod } 2$$

### 引き算 (XOR)

$$0-0 = 0+0 = 0 \text{ mod } 2$$

$$0-1 = 0+1 = 1 \text{ mod } 2$$

$$1-0 = 1+0 = 1 \text{ mod } 2$$

$$1-1 = 1+1 = 0 \text{ mod } 2$$

### 割り算 (AND)

$$1/0 = 0 \text{ mod } 2$$

$$1/1 = 1*1 \text{ mod } 2$$

復習：

1 と足して 0 になる数は

1 なので、 $-1 = 1 \text{ mod } 2$

復習：

1 とかけて 1 になる数は

1 なので、 $1^{-1} = 1$

コンピューターで演算しやすい！

## 係数が mod 2 の世界の既約多項式

実は、mod 2 の世界では  $x^2 + 1$  は既約多項式ではない。

$$(x + 1)(x + 1) = x^2 + (1 + 1)x + 1 = x^2 + 1$$

例えば、

$$x^2 + x + 1$$

は mod 2 の世界で既約多項式。

## 多項式の世界の四則演算

$x^2 + x + 1$  の世界の足し算

+	1	$x$	$x + 1$
1	0	$x + 1$	$x$
$x$	$x + 1$	0	1
$x + 1$	$x$	1	0

各桁ごとに XOR を取るイメージ。（やってみよう！）

$x^2$  以上の項はあまりを求める過程で消滅。



## 多項式の世界の四則演算

$x^2 + x + 1$  の世界の引き算

—	1	$x$	$x + 1$
1	0	$x + 1$	$x$
$x$	$x + 1$	0	1
$x + 1$	$x$	1	0

各桁ごとに XOR を取るイメージ。（やってみよう！）

足し算も引き算も XOR で計算するから、同じ結果になる。

## 多項式の世界の四則演算

$$x^2 + x + 1$$

の世界の掛け算

$\times$	$1$	$x$	$x + 1$
$1$	$1$	$x$	$x + 1$
$x$	$x$	$x^2$	$x(x + 1) = x^2 + x$
$x + 1$	$x + 1$	$1$	$x^2 + 1$

$x^2$  は  $x^2 + x + 1$  のあまりの世界では存在しない。

## 多項式の世界の四則演算

$x^2$  はこんな感じで あまりの世界に落とす。

$$\begin{array}{r} 1 \\ x^2 + x + 1 \overline{) x^2} \\ \underline{x^2 + x + 1} \\ x + 1 \end{array}$$

## 多項式の世界の四則演算

$$x^2 + x + 1$$

の世界の掛け算

$\times$	$1$	$x$	$x + 1$
$1$	$1$	$x$	$x + 1$
$x$	$x$	$x + 1$	$1$
$x + 1$	$x + 1$	$1$	$x$

## 多項式の世界の四則演算

$$x^2 + x + 1$$

の世界の掛け算

$\times$	1	$x$	$x + 1$
1	1	$x$	$x + 1$
$x$	$x$	$x + 1$	1
$x + 1$	$x + 1$	1	$x$

$x$  とかけて  
1 になる数  
は  
 $x+1$

$x+1$  とかけて  
1 になる数は  
 $x$

## 多項式の世界の四則演算

$$x^2 + x + 1$$

の世界の掛け算

$\times$	1	$x$	$x + 1$
1	1	$x$	$x + 1$
$x$	$x$	$x + 1$	1
$x + 1$	$x + 1$	1	$x$

すべてに対して、  
逆元が存在している！

割り算ができる！

## 多項式を bit で表現する

mod 2 の世界での多項式の係数はすべて 1 か 0 になる。

さらに、あまりを求めているので、 $x$  の次数の上限も決まっている。

$$x^n$$

例えば、 $n$  の最大値が 8 だったら、係数は 8 個あり、

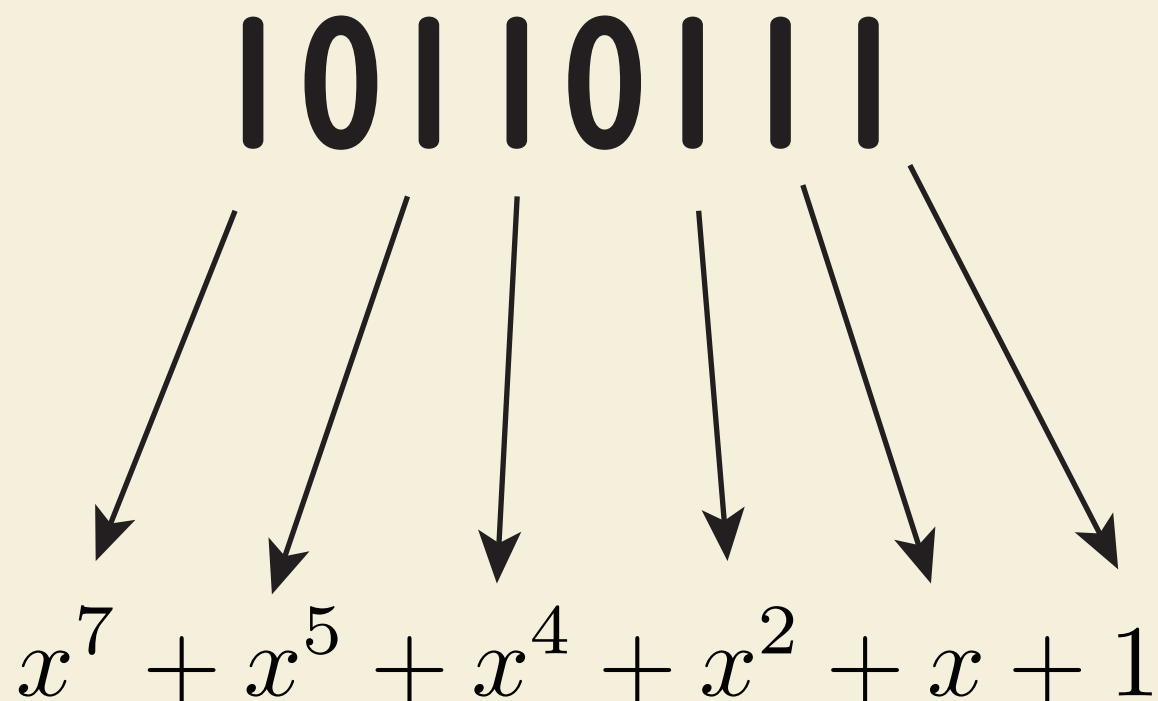
さらにすべての係数は 0 か 1 なので、

8bit で表現できる！  $2^8$  個多項式があるので、

$$GF(2^8)$$

と表記されることが多い。（具体例は次のページ）

## bit 列と多項式の対応



このように対応させると、xor を計算するだけで  
足し算や引き算をすることができる！



## bit 列と剰余の計算

$$x^2 + x + 1 : 111$$

$$x^4 : 10000$$

$x^2$  を  $x^2 + x + 1$  で割ったあまりは？

$$\begin{array}{r} 110 \\ 111 \overline{) 10000} \\ \underline{111} \phantom{00} \\ 1100 \\ \underline{111} \phantom{0} \\ 10 \end{array}$$

XOR →

割る数 (111) より  
bit が小さくなったので  
終わり  $x$

## CRC の概要

$$\begin{array}{r} 111 \overline{) 1000000} \\ \underline{111} \phantom{00} \\ 110000 \\ \underline{111} \phantom{00} \\ 1000 \\ \underline{111} \phantom{00} \\ 110 \\ \underline{111} \\ 01 \end{array}$$

1. 割られる数に、割る数の bit 数 - 1 個分 0 を入れる
2. あまりを求める。

10000( 生データ ) と 01 を相手に送信する。

## CRC の概要

$$\begin{array}{r} 111 \overline{) 1000001} \\ \underline{111} \phantom{00000} \\ 110001 \\ \underline{111} \phantom{0000} \\ 1001 \\ \underline{111} \phantom{000} \\ 111 \\ \underline{111} \\ 00 \end{array}$$

1. 受信者は生データ  
に **01** を付け加えたものの余りを求める。
2. データが書き換えられてなければ  
必ず **0** になる (0 にならないければ、  
データがなにかおかしい！)

これで改ざん検知ができる。

なぜこれでできるのか？

10000**00** を 111(3bit)

で割ったあまりは必ず 2bit 以内になる。

**0** を 2 つ付け加えたのは、余りを付け加える場所を作るため。

10000**00** からあまりの値 (**01**) をひくと、必ず割り切れる。[ 参考参照 ]

mod 2 の世界では 足し算も引き算も XOR で表される。

10000**01** は必ず割り切れる。

参考 :  $15/7 = 2$  あまり 1 だから  $15-1=14$  は 7 で割り切れる。

本日のまとめ。

CRC の理論を説明した。

多項式の有限体を構成して、あまりを求めることで  
誤り訂正をすることができた！

しかし、実用で使われているアルゴリズムはもう少し複雑に  
なっているので次回説明する。

次回 png の CRC を求める