

CTF 入門

CRC32 で 誤りを 検知する

実践編

@kurenai f

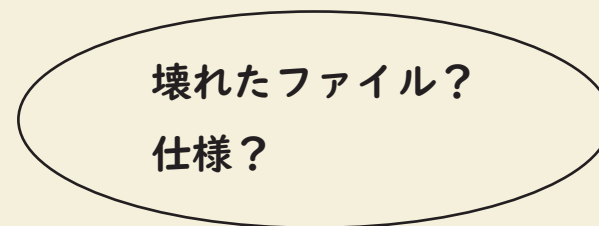
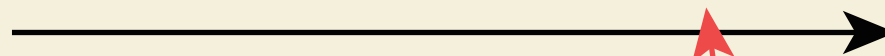
CRC32 って？

Cyclic Redundancy Check

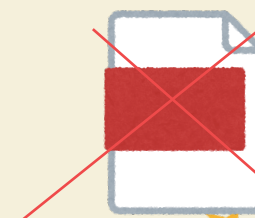
「誤りを検知する」検査方法



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)

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
```

今日のこの講義では
この CRC を求めるための
理論を勉強します

CRC の概要

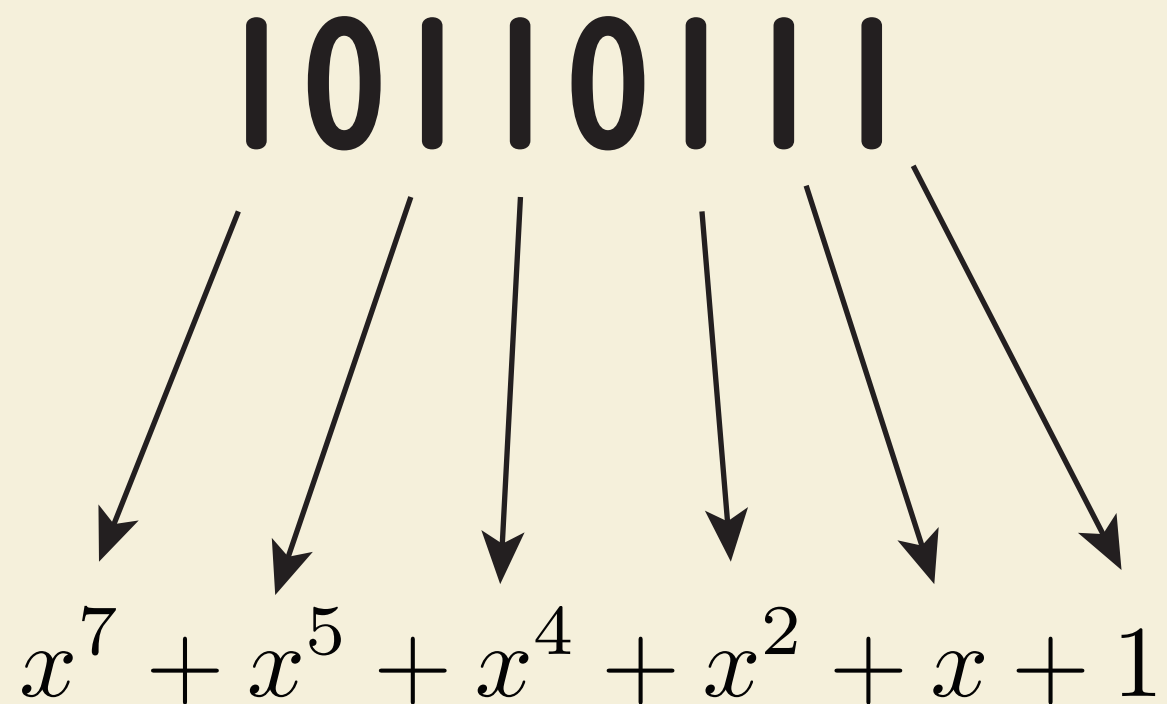
$$\begin{array}{r} 111 \overline{) 1000000} \\ \underline{111} \\ 110000 \\ \underline{111} \\ 1000 \\ \underline{111} \\ 110 \\ \underline{111} \\ 01 \end{array}$$

1. 割られる数（生データ）に、
割る数の bit 数 - 1 個分 0 を入れる
2. あまりを求める。

筆算の引き算の代わりに
XOR を計算する。

10000（生データ）と 01 を相手に送信する。

bit 列と多項式の対応



実はこの bit 列は多項式を表しています！

CRC の多項式

CRC の、割る数はファイルや用途によって異なる。

		(0x02CC)
CRC-16-Fletcher	CRCではない。フレッチャーの検査合計	Adler-32 A & B CRC で使用
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (X.25、V.41、CDMA、Bluetooth、XMODEM、HDLC、PPP、IrDA、BACnet; CRC-CCITTとも)	0x1021 / 0x8408 (0x8810 [5])
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$ (SDLC、USB、その他; CRC-16とも)	0x8005 / 0xA001 (0xC002)
CRC-24-Radix-64	$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ (FlexRay)	0x864CFB / 0xDF3261 (0xC3267D)
CRC-30	$x^{30} + x^{29} + x^{21} + x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + x + 1$ (CDMA)	0x2030B9C7 / 0x38E74301 (0x30185CE3)
CRC-32-Adler	CRCではない; Adler-32	Adler-32参照
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (V.42, MPEG-2, zlib, PNG [10])	0x04C11DB7 / 0xEDB88320 (0x82608EDB [7])

png の CRC の多項式

png の多項式はこれ（最上位 bit は省略されてるので、実際は 0x104C11DB7）

CRC-16-Fletcher	CRCではない。フレッチャーの検査合計	(0x02CC) Adler-32 A & B CRC で使用
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (X.25、V.41、CDMA、Bluetooth、XMODEM、HDLC、PPP、IrDA、BACnet; CRC-CCITTとも)	0x1021 / 0x8408 (0x8810 [5])
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$ (SDLC、USB、その他; CRC-16とも)	0x8005 / 0xA001 (0xC002)
CRC-24-Radix-64	$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ (FlexRay)	0x864CFB / 0xDF3261 (0xC3267D)
CRC-30	$x^{30} + x^{29} + x^{21} + x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + x + 1$ (CDMA)	0x2030B9C7 / 0x38E74301 (0x30185CE3)
CRC-32-Adler	CRCではない; Adler-32	Adler-32参照
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (V.42, MPEG-2, zlib, PNG [10])	0x04C11DB7 / 0xEDB88320 (0x82608EDB [7])

png の CRC

様々なトラブルに対応するため、様々な**前処理**・**後処理**がある。



<https://www.slideshare.net/7shi/crc32#31>

こちらの 7shi さんの資料を参考に作らせて
いただきました…！

前処理

バイナリデータ (16 進数表記)

0x0000deadbeefcafebabe



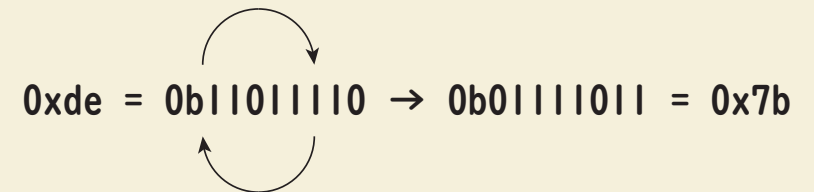
1byte=8bits ずつ分解

00 00 de ad be ef ca fe ba be



1byte ずつ bit の「位置を」反転させる

00 00 7b b5 7d f7 53 7f 5d 7d



0x00007bb57df7537f5d7d

前処理

CRC の計算

後処理

前処理

0x00007bb57df7537f5d7d



4bytes=32bits 分 0 を追加
(割る数 -1bit 分)

0x00007bb57df7537f5d7d00000000



上位 4bytes を bitflip

0xffff844a7df7537f5d7d00000000

0x10 と 0x0000000000010 を区別するために
flip する

前処理

CRC の計算

後処理

あまりを求める。

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

0xldb710640 で割り算すると ...

0xc87c14fc になります！！！！！！！！

前処理

CRC の計算

後処理

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル

タイトル