暗記のすすめ

暗号技術を理解するたった一つの方法

kobae964 2025-03-20

この値・式覚えてますか?

- あの最近はやりの SSH の鍵タイプ、なんだっけ…?
  - ed?????

この値・式覚えてますか?

- あの最近はやりの SSH の鍵タイプ、なんだっけ…?
  - ed25519

この値・式覚えてますか?

- ed25519 の元となっている Curve25519 の式、なんだっけ…?
    - v^2 = u^3 + ??????u^2 + u

この値・式覚えてますか?

- ed25519 の元となっている Curve25519 の式、なんだっけ…?
  - v^2 = u^3 + 486662u^2 + u

この値・式覚えてますか?

- では ed25519 の式、なんだっけ…?
  - -x^2 + y^2 = 1 - (??????/??????)x^2y^2

この値・式覚えてますか?

- では ed25519 の式、なんだっけ…?
  - -x^2 + y^2 = 1 - (121665/121666)x^2y^2

この値・式覚えてますか?

- 486662 と 121665/121666 の関係、どうだったかな…?
  - A = 486662, d = -121665/121666 とすると d = -(???)/(???)
- v^2 = u^3 + Au^2 + u と -x^2 + y^2 = 1 + dx^2y^2 の変換、どうだったかな…?
  - x = (....), y = (....)

この値・式覚えてますか?

- 486662 と 121665/121666 の関係、どうだったかな…?
  - A = 486662, d = -121665/121666 とすると d = -(A-2)/(A+2)
- v^2 = u^3 + Au^2 + u と -x^2 + y^2 = 1 + dx^2y^2 の変換、どうだったかな…?
  - x = sqrt(-(A+2))u/v, y = (u-1)/(u+1)

で、覚えて何の役に立つの?

メリットは色々あるが、以下のことが大きい:

- 頭の中で実験・考察できる
- CTF で変な実装を見た時、嗅覚が働いてすぐにわかる
- ワーキングメモリーが鍛えられる

頭の中で実験・考察できる

- Edwards25519 の基点の y = 4/5 って 16 進でどうだったっけ…?
- p = 2^255 - 19 だから p mod 5 = 8 - 4 = 4
- だから 4/5 mod p = (4p+4)/5 だ!
- 4p+4 = 2^257 - 72 = 0x1 ff ff ... ff b8 のはず!
- これを 5 で割ると 0x66 66 ... 66 58 だ!
- 実際にソースを読みに行くと合ってる

頭の中で実験・考察できる

- 暗算でやる意味は?
- こうした苦労したエピソードがあると y = 4/5 という値を忘れにくい
- Curve25519 で u = 9 というのも覚えておけば、y = (u-1)/(u+1) や u = (1+y)/(1-y) も忘れにくい
- 淡い記憶を複数持っておいて定期的に検算することで、記憶を強固にするイメージ

CTF で変な実装を見た時、嗅覚が働いてすぐにわかる

どこに脆弱性があるでしょう

```python
def main():
    signal.alarm(300)

    flag = os.environ.get("FLAG", "0nepoint{frog_pyokopyoko_3_pyokopyoko}")
    assert len(flag) < 2*8*n
    while len(flag) % 16 != 0:
        flag += "\0"

    G = (gx, gy)
    s = randrange(0, q)

    print("sG = {}".format(mul(s, G)))
    tG = ast.literal_eval(input("tG = "))  # you should input something like (x, y)
    assert len(tG) == 2
    assert type(tG[0]) == int and type(tG[1]) == int
    share = to_bytes(mul(s, tG))
```

CTF で変な実装を見た時、嗅覚が働いてすぐにわかる

どこに脆弱性があるでしょう→入力の validation をしていない!

https://alpacahack.com/ctfs/zer0pts-ctf-2022/challenges/eddh

```python
def main():
    signal.alarm(300)

    flag = os.environ.get("FLAG", "0nepoint{frog_pyokopyoko_3_pyokopyoko}")
    assert len(flag) < 2*8*n
    while len(flag) % 16 != 0:
        flag += "\0"

    G = (gx, gy)
    s = randrange(0, q)

    print("sG = {}".format(mul(s, G)))
    tG = ast.literal_eval(input("tG = "))  # you should input something like (x, y)
    assert len(tG) == 2
    assert type(tG[0]) == int and type(tG[1]) == int
    share = to_bytes(mul(s, tG))
```

ワーキングメモリーが鍛えられる

- CTF や暗号ライブラリーの読解では、そこそこ長いコードの理解が必要
- 頭の中に多くの情報が載せられると、役に立つ!
- 頭の中の情報がリンクされていると、忘却しにくい!

記憶メソッド

- 実装を読む (インプット)
- 論文/RFC を読む (インプット)
- 自分でコードゴルフしてみる (アウトプット)
- 暗算する (分析)

実装を読む (インプット)

- ビット演算を駆使して分岐を消せるのか…! (驚き)
- https://github.com/openssh/openssh-portable/blob/V_9_9_P2/

openssh-portable / ed25519.c

Top

Code Blame 2030 lines (1831 loc) · 197 KB

Raw

```c

static crypto_uint32 fe25519_equal(crypto_uint32 a,crypto_uint32 b) /* 16-bit inputs */
{
  crypto_uint32 x = a ^ b; /* 0: yes; 1..65535: no */
  x -= 1; /* 4294967295: yes; 0..65534: no */
  x >>= 31; /* 1: yes; 0: no */
  return x;
}

```

論文/RFC を読む (インプット)

- Curve25519
- 点の u 座標だけで楕円曲線のスカラー倍が計算できる!?
- https://cr.yp.to/ecdh/curve25519-20060209.pdf

- Use x/z inside scalar multiplication, not $(x/z, y/z)$ or $(x/z^2, y/z^3)$.

自分でコードゴルフしてみる (アウトプット)

- CRC32 とかはかなり簡単 (筆者実装は Go で 10 行)
  - https://sizu.me/koba_e964/posts/ub7ak2mdoknv

これだけです

```
func crc32(b []byte) uint32 {
    u := uint32(0xffff_ffff)
    for _, b := range b {
        u ^= uint32(b)
        for i := 0; i < 8; i++ {
            u = u>>1 ^ 0xedb8_8320*(u&1)
        }
    }
    return u ^ 0xffff_ffff
}
```

暗算する (分析)

- 怪しい公式を複数組み合わせて、矛盾を検出・解消する
- ワーキングメモリーをフル稼働させて、頭の中にすべて載せる
    - 載らなかったら頑張ろう

暗算する (分析)
やり方

- j-不変量が j の楕円曲線、$y^2 = x^3 - \frac{3j}{j-1728}x + \frac{2j}{j-1728}$ だったっけ…?
- $y^2 = x^3 + ax + b$ の j-不変量は $\frac{4a^3}{4a^3+27b^2}$ だったっけ…?
- そう思って j-不変量を計算すると $\frac{j}{1728}$ になり、1728 忘れに気付ける

暗算する (分析)
やり方

- 頭の中でやる方法
- $j - 1728 = A$ としてしまおう。a = -3j/A, b = 2j/A
- $$\frac{4a^3}{4a^3 + 27b^2} = \frac{\frac{-108j^3}{A^3}}{\frac{-108j^3}{A^3} + \frac{108j^2}{A^2}}$$
- 分子と分母を 108j^2/A^3 で割ると $\frac{-j}{-j + A} = \frac{j}{1728}$

おすすめ分野

- ASCII code
- 初等整数論
- 楕円曲線論

ASCII code

- 'A' == 0x41 とかだったりするアレ
- 英語のアルファベットにランダムアクセスできると強い
    - せめて途中のポイントを覚えよう
    - D => 4, H => 8, L => 12, P => 16, T => 20, X => 24
- 十進数と十六進数の変換もできるようになっておこう
    - 16, 32, 48, 64, 80, 96, 112 = 7 * 16 までの 16 の倍数でいいので楽

初等整数論

- 具体例の宝庫
- 平方剰余とか乗法群
  - $a^{(p-1)/2} \equiv 1 \pmod p$ なら a は平方剰余
  - $p \equiv 1 \pmod 3$ なら mod p で 1 の 3 乗根がある
    - 存在定理なので実際の構築との間にはギャップあり、実際に調べよう
    - ランダムに a をとれば $a^{(p-1)/3}$ が確率 2/3 で非自明な 3 乗根

楕円曲線論

- おもしろい
- 楕円曲線論は広大すぎて迷いやすいので、CTF で使いそうなところから
  - おすすめ初手: 有限体上の楕円曲線
    - 手前味噌: https://qiita.com/kobae964/items/e3927b57f1bf91f4caf6
    - SafeCurves: https://safecurves.cr.yp.to/

まとめ

- CTF でも暗号理論でも、楽しむためにはスムーズに記憶することが不可欠
- スムーズな記憶のためにはエピソードが大事
- 暗算スキルを高めてエピソードを頭の中で錬成しよう!

予備スライド

予備スライド

0xedb8_8320 って何?

- GF(2) の多項式
  - つまり mod 2 で色々やるということ
- 下位ビットが次数の高い側
- x^32 は省略されている
- つまり $(1 + x + x^2) + (x^4 + x^5 + x^7) + \cdots + x^{26} + x^{32}$ ということ
  - 0xe = 0b1110, 0xd = 0b1101 に注意。

Edwards25519 の加法公式は?

- https://ed25519.cr.yp.to/ed25519-20110926.pdf
- ほとんど複素数の積、ただし…
  - x と y が逆、i^2 = 1
  - 分母がある

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 + x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right)$$