2024 年度

情報科学実験 02

レポート課題

実験テーマ

KUE-CHIP を用いた公開伴暗号通信の実装(1)

実験実施日: 2024 年 10 月 10 日
レポート提出日: 2024 年 10 月 17 日
提出期限: 2024 年 10 月 17 日

報告者: J2200071 齊藤 隆斗
実験グループ: A 班

目次

第 1 章　実験内課題

1.1 予習問題(前期の復習)

下の手順を参考に 2 つの整数 x, n に対し Modular(剰余)Mod[x, n] を 求めるアセンブリ語プログラムを作成し,命令毎の動作を説明せよ.

このプログラムでは以下のような入出力を想定する.

入力

```
180: (X)**H (自然数: 1 バイト)
182: (N)**H (自然数: 1 バイト)
```

出力

```
ACC: Mod[X, N]
```

与えられた問題を解くプログラムは以下のようになった.

```
00: 65    LD ACC, (80)
01: 80
02: 6D    LD IX,  (82)
03: 82
04: A1    SUB ACC, IX
05: 32    BZP 02
06: 02
07: B1    ADD ACC, IX
08: 0F
```

プログラムの説明

プログラムの手順としては、X から n を引いていき、それが負数になったら、n を足すことによって剰余を求めるというものである. X から n を引く際に、繰り返す回数は自然数であるから、X は $\left(n \times \left(\textbf{自然数}\right) + \textbf{整数}\right)$ という形で表せる. 最後に n を足す理由は、剰余はnより小さく、かつ非負である必要があるからである.

1.2 課題 01

Mod$[ax]$ を求めるプログラムを作成し、KUECHIP で実行せよ.

このプログラムでは以下のような入出力を想定する.

入力

```
180: (X)**H (乗数)
182: (N)**H
1B0: (A)**H *(被乗数)
```

出力

```
ACC: Mod[AX,N]
```

与えられた問題を解くプログラムは以下のようになった.

```
00: C0    EOR ACC, ACC
01: C9    EOR IX,  IX
02: B5    ADD ACC, (80)
03: 80
04: A5    SUB ACC, (82)
05: 82
06: 32    BZP 04
07: 04
08: B5    ADD ACC, (82)
09: 82
0A: BA    ADD IX,  1
0B: 01
0C: FD    CMP IX,  (B0)
0D: B0
0E: 3A    BN 02
0F: 02
10: 0F    HLT
```

プログラムの説明

このプログラムでは、Mod 演算が足し算に対して不変であることを利用する. [1] すなわち、

$$s_0 = 0, s_{i+1} = \mathrm{Mod}[s_i + x, n]$$

としたとき

$$\mathrm{Mod}[ax, n] = s_a$$

を利用する.

よって、期待する出力である Mod[AX,N]を得るためには、s_iのiがAとなるまで

$$s_{i+1} = \mathrm{Mod}[s_i + x, n]$$

を計算していけば良い. このプログラムでは IX でs_iのiをカウントし、ACC にs_iを格納して計算を行っている. ここで、予習問題にて作成した Mod[x,n]を求めるプログラムを利用し、$\mathrm{Mod}[s_i + x, n]$を求めている.

1.3 課題 02

3 つの整数x, k, nに対し、$\mathrm{Mod}[x^k, n]$を計算するプログラムを実装せよ.

このプログラムでは以下のような入出力と作業領域を想定する.

入力

```
180: (X)
181: (K)
182: (N)
```

作業領域

```
1B0: (W1)**H (作業領域 1: 途中経過 A=Mod[X^p,N], 被乗数)
1C0: (W2)**H (作業領域 2: X の乗算を実行した回数 p)
```

出力

```
ACC: Mod[X^K, N]
```

与えられた問題を解くプログラムは以下のようになった.

```
00: 65    LD ACC, (80)
01: 80
02: 6A    LD IX,  1
03: 01
04: 7D    ST IX,  (C0)
05: C0
06: 75    ST ACC, (B0)
07: B0
08: C0    EOR ACC, ACC
09: C9    EOR IX,  IX
0A: B5    ADD ACC, (80)
0B: 80
0C: A5    SUB ACC, (82)
0D: 82
0E: 32    BZP 0C
0F: 0C
10: B5    ADD ACC, (82)
11: 82
12: BA    ADD IX,  1
13: 01
14: FD    CMP IX,  (B0)
15: B0
16: 3A    BN 0A
17: 0A
18: 6D    LD IX,  (C0)
19: C0
1A: BA    ADD IX, 1
1B: 01
1C: FD    CMP IX, (81)
1D: 81
1E: 3A    BN 04
1F: 04
20: 0F    HLT
```

プログラムの説明

このプログラムでは、Mod 演算が乗算に対して不変であることを利用する.すなわち、

$$r_0 = 0, r_{i+1} = \mathrm{Mod}[r_i x, n]$$

としたとき

$$\mathrm{Mod}[x^k, n] = r_k$$

を利用する.

よって、期待する出力である Mod[X^K,N]を得るためには、r_iのiがKとなるまで

$$r_{i+1} = \mathrm{Mod}[r_i x, n]$$

を計算していけば良い.

このプログラムにおいて、課題 01 において作成した Mod[ax,n]を利用し、$r_{i+1} = \mathrm{Mod}[r_i x, n]$ を求めている. また、このプログラムでは IX, ACC を Mod[AX,N]の計算を行う上で必要となるため、r_iのiは作業領域 1C0 に格納しておき、r_iは作業領域 1B0 に格納しておく. この作業領域 1C0 に格納しているiの値がKになるまで処理を繰り返すことでr_kを求めることができる.

1.4 課題 03

課題 02 をプログラム領域のアドレス 080H 以下に実装し、分岐命令 BA を使い、ルーチンとして 2 つの数値を連続して暗号化し、メモリに保存するプログラムを作成せよ.

このプログラムでは以下のような入出力と作業領域を想定する.

入力

```
170: (D) 02H 0FH (データ)
181: (K) 05H     (公開鍵)
182: (N) 5BH     (共通鍵)
```

作業領域

```
150: (LN)**H (作業領域: 残り繰り返し回数)
180: (X)**H  (作業領域: 暗号化するデータの一時保存)
```

出力

```
190: (EN) 20H 47H (暗号化データ)
```

与えられた問題を解くプログラムは以下のようになった.

```
00: 6A    LD IX, 1
01: 01
02: 7D    ST IX, (50)
03: 50
04: 67    LD ACC, (IX+70)
05: 70
06: 75    ST ACC, (80)
07: 80
08: 30    BA 80
09: 80
0A: 6D    LD IX, (50)
0B: 50
0C: 77    ST ACC, (IX+90)
0D: 90
0E: AA    SUB IX, 1
0F: 01
10: 32    BZP 02
11: 02
12: 0F    HLT


# Subroutine
80: 65    LD ACC, (80)
81: 80
82: 6A    LD IX,  1
```

```
83: 01
84: 7D    ST IX,  (C0)
85: C0
86: 75    ST ACC, (B0)
87: B0
88: C0    EOR ACC, ACC
89: C9    EOR IX,  IX
8A: B5    ADD ACC, (80)
8B: 80
8C: A5    SUB ACC, (82)
8D: 82
8E: 32    BZP 8C
8F: 8C
90: B5    ADD ACC, (82)
91: 82
92: BA    ADD IX,  1
93: 01
94: FD    CMP IX,  (B0)
95: B0
96: 3A    BN 8A
97: 8A
98: 6D    LD IX,  (C0)
99: C0
9A: BA    ADD IX, 1
9B: 01
9C: FD    CMP IX, (81)
9D: 81
9E: 3A    BN 84
9F: 84
A0: 30    BA 0A
A1: 0A
```

プログラムの説明

課題 02 で作成した 000H からのプログラム領域に格納したプログラムを 080H 以下に格納し、サブルーチンとして実装しておく. これにより、適切なデータ領域(180H)に引数が与えられれば、その引数(180H に格納されたデータ)を暗号化するような関数として動作する. この関数を 2 回呼び出すことによって、データ領域の連続した値(170H, 171H)を暗号化して、連続したデータ領域(190H, 191H)に出力することができる.

このプログラムにおいてまず、IX に 1 を格納し、171H(170H+IX)のデータから読みこみ、180H 領域に引数としてセットし、関数を呼び出し、1 つ目のデータを暗号化し 191H(190H+IX)に格納する.その後、IX の値をデクリメントし、170H(170H+IX)のデータに対しても同様の処理を行うことで、暗号化したデータを 190H(190H+IX)に格納する.

第 2 章　考察

2.1 考察課題 01

予習問題, 課題 01,課題 02 で実装したプログラムの計算量を評価せよ.命令回数を a, x, k, n 等を用いて定量的に評価すること(例えば, $O(ax^2kn)$ 等のように表す).

2.1.1 考察

2.1.1.1 予習問題

まず最初に予習問題のプログラムの計算量について考える. このプログラムでは、x から n を負になるまで繰り返し引いていくので、繰り返しの回数は

$$\left\lfloor \frac{x}{n} \right\rfloor + 1$$

となる.ここで、x が n で割り切れる場合もこの回数分繰り返しが行われることに注意する. よって、このプログラムの計算量は $O\left(\frac{x}{n}\right)$ となる.

2.1.1.2 課題 01

次に課題 01 のプログラムの計算量について考える. このプログラムでは、

$$s_{i+1} = \mathrm{Mod}[s_i + x, n]$$

をs_aを得るまで繰り返す.よって、$\mathrm{Mod}[s_i + x, n]$をa回繰り返す. ここで、$\mathrm{Mod}[s_i + x, n]$の計算について、$0 \leq s_i < n$であるから、 $0 \leq \frac{s_i}{n} < 1$ となる. よって、

$$O\left(\frac{s_i + x}{n}\right) = O\left(\frac{s_i}{n} + \frac{x}{n}\right) = O\left(\frac{x}{n}\right)$$

よって、課題 01 のプログラムの計算量は $O\left(\frac{ax}{n}\right)$ となる.

2.1.1.3 課題 02

最後に課題 02 のプログラムの計算量について考える. このプログラムでは、

$$r_{i+1} = \mathrm{Mod}[r_i x, n]$$

をr_kを得るまで繰り返す.よって、$\mathrm{Mod}[r_i x, n]$ を k 回繰り返す. ここで、$0 \leq r_i < n$ であるから $0 \leq \frac{r_i}{n} < 1$ である.よって、$\mathrm{Mod}[r_i x, n]$ の計算量は

$$O\left(\frac{r_i x}{n}\right) = O\left(\frac{r_i}{n} \cdot x\right) = O(x)$$

よって、課題 02 の計算量は $O(kx)$ となる.

2.2 考察課題 02

課題 01 において,Mod$[ax, N]$の値を,まずaxの計算を行った後,N による剰余を求めることにより計算した場合,課題 1 の方法で計算したときとどのような違いがあるか考察せよ.

2.2.1 考察

Mod$[ax, N]$の値を求める際に、先に ax の乗算を行った後に ax のNによる剰余を求める場合、ax の計算結果がオーバーフローしてしまい、その後の N による剰余の計算において正常な出力ができなくなってしまう可能性がある. そこで、課題 01 のように

$$s_{i+1} = \mathrm{Mod}[s_i + x, n]$$

を繰り返し求めることで、s_a を求める方法では、各s_iを計算したとき、s_iは常に0 以上 $n-1$ 以下となるので、次に計算すべきs_{i+1}の計算において、$s_i + x$ がオーバーフローする可能性を小さくすることができる.

参考文献

[1] 情報科学実験 *II* 群馬大学 情報学部 *2024* 年度後期.