РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Отчёт по лабораторной работе №7. Дискретное логарифмирование в конечном поле

Дисциплина: *Математические основы защиты информации и информационной безопасности*

Студент: Лапшенкова Любовь Олеговна, 10322127633
Группа: НФИмд-02-21
Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

List of Figures

List of Tables

1 Цель работы

Целью данной лабораторной работы является ознакомление с алгоритмом, реализующим Ро-метод Полларда для дискретного логарифмирования, а также программное воплощение данного алгоритма.

2 Задание

1. Реализовать рассмотренный в инструкции к лабораторной работе алгоритм программно.
2. Подставить численное значение из примера в программный код, проверить правильность полученного ответа.

3 Теоретическое введение

В данной лабораторной работе предметом нашего изучения стал Ро-метод Полларда для задач дискретного логарифмирования.

3.1 Ро-метод Полларда

Ро-метод Полларда для дискретного логарифмирования (ρ -метод) — алгоритм дискретного логарифмирования в кольце вычетов по простому модулю, имеющий экспоненциальную сложность. Предложен британским математиком Джоном Поллардом в 1978 году, основные идеи алгоритма очень похожи на идеи ро-алгоритма Полларда для факторизации чисел. Данный метод рассматривается для группы ненулевых вычетов по модулю p, где p — простое число, большее 3 [1].

3.2 Постановка задачи дискретного логарифмирования

Постановка задачи дискретного логарифмирования представлена следующим образом:

Для заданного простого числа p и двух целых чисел a и b требуется найти целое число x, удовлетворяющее сравнению:

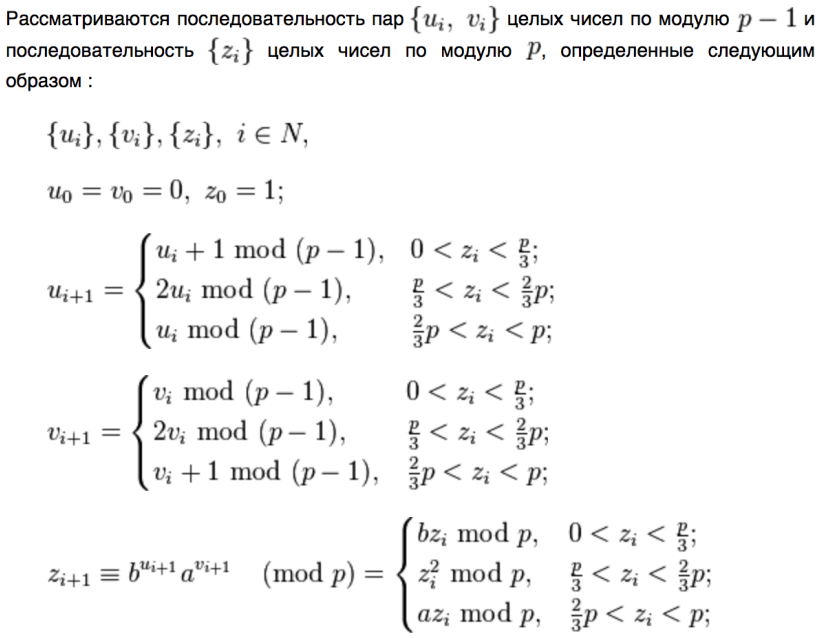
$$a^x \equiv b \pmod{p},$$

где b является элементом циклической группы G, порожденной элементом a.

Figure 3.1: Постановка задачи дискретного логарифмирования

3.3 Алгоритм Ро-метода Полларда

Исходя из теоретических сведений, алгоритм Ро-метода Полларда представлен ниже [2].



Рассматриваются последовательность пар $\{u_i,\ v_i\}$ целых чисел по модулю $p-1$ и последовательность $\{z_i\}$ целых чисел по модулю p, определенные следующим образом :

$$\{u_i\},\{v_i\},\{z_i\},\ i \in N,$$

$$u_0 = v_0 = 0,\ z_0 = 1;$$

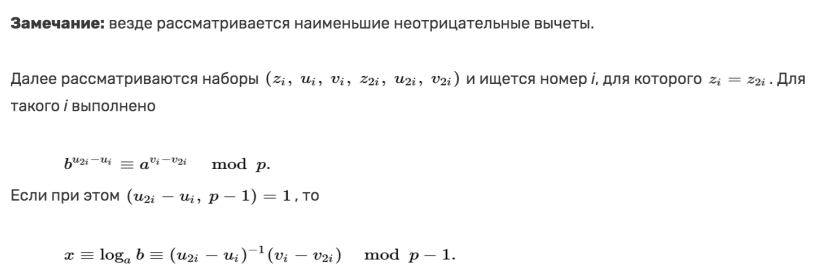
$$u_{i+1} = \begin{cases} u_i + 1 \bmod (p-1), & 0 < z_i < \frac{p}{3}; \\ 2u_i \bmod (p-1), & \frac{p}{3} < z_i < \frac{2}{3}p; \\ u_i \bmod (p-1), & \frac{2}{3}p < z_i < p; \end{cases}$$

$$v_{i+1} = \begin{cases} v_i \bmod (p-1), & 0 < z_i < \frac{p}{3}; \\ 2v_i \bmod (p-1), & \frac{p}{3} < z_i < \frac{2}{3}p; \\ v_i + 1 \bmod (p-1), & \frac{2}{3}p < z_i < p; \end{cases}$$

$$z_{i+1} \equiv b^{u_{i+1}} a^{v_{i+1}} \pmod{p} = \begin{cases} bz_i \bmod p, & 0 < z_i < \frac{p}{3}; \\ z_i^2 \bmod p, & \frac{p}{3} < z_i < \frac{2}{3}p; \\ az_i \bmod p, & \frac{2}{3}p < z_i < p; \end{cases}$$

Figure 3.2: Алгоритм Ро-метода Полларда. 1.

При этом, важно учесть следующие замечания [3]:



Замечание: везде рассматривается наименьшие неотрицательные вычеты.

Далее рассматриваются наборы $(z_i,\ u_i,\ v_i,\ z_{2i},\ u_{2i},\ v_{2i})$ и ищется номер *i*, для которого $z_i = z_{2i}$. Для такого *i* выполнено

$$b^{u_{2i}-u_i} \equiv a^{v_i - v_{2i}} \mod p.$$

Если при этом $(u_{2i} - u_i,\ p-1) = 1$, то

$$x \equiv \log_a b \equiv (u_{2i} - u_i)^{-1}(v_i - v_{2i}) \mod p-1.$$

Figure 3.3: Алгоритм Ро-метода Полларда. 2.

3.4 Сложность алгоритма

Эвристическая оценка сложности составляет $O(p^{1/2})$.

4 Выполнение лабораторной работы

Примечание: комментарии по коду представлены на скриншотах к каждому из проделанных заданий.

В соответствии с заданием, была написана программа по воплощению алгоритма Ро-метода Полларда для задач дискретного логарифмирования.

Программный код и результаты выполнения программ представлен ниже.

4.1 Ро-метод Полларда

```
def f(c,u,v):
  '''
  Ввели функцию, завис. от c,u,v
  '''
  if c<53:
    return 10*c%107,u+1,v
  else:
    return 64*c%107,u,v+1
```

Figure 4.1: Вспомогательная функция, зависящая от c,u,v

```
def rasshir_algorithm_Evklida(a,b):
  '''
  расширенный алгоритм Евклида
  '''
  r=[]
  x=[]
  y=[]
  r.append(a)
  r.append(b)
  x.append(1)
  x.append(0)
  y.append(0)
  y.append(1)
  i=1
  while r[i]!=0:
    i+=1
    r.append(r[i-2]%r[i-1])
    if r[i]==0:
      d=r[i-1]
      x=x[i-1]
      y=y[i-1]
    else:
      x.append(x[i-2]-((r[i-2]//r[i-1])*x[i-1]))
      y.append(y[i-2]-((r[i-2]//r[i-1])*y[i-1]))
  return d,x,y
```

Figure 4.2: Вспомогательная функция. Расширенный алгоритм Евклида

```
def Pollard(p,a,r,b,u,v):
  '''
  Метод Полларда для логарифмирования в конечном поле
  '''
  c=a**u*b**v%p
  d=c
  uc=u
  vc=v
  ud=u
  vd=v
  c,uc,vc=f(c,uc,vc)
  c%=p
  d,ud,vd=f(*f(d,ud,vd))
  d%=p
```

Figure 4.3: Реализация алгоритма Ро-метода Полларда для логарифмирования

```
while c%p!=d%p:
  '''
  условие работы цикла
  '''
  c,uc,vc=f(c,uc,vc)
  c%=p
  d,ud,vd=f(*f(d,ud,vd))
  d%=p

v=vc-vd
u=ud-uc

d,x,y=rasshir_algorithm_Evklida(v,r)

while d!=1:
  v/=d
  u/=d
  r/=d
  d,x,y=rasshir_algorithm_Evklida(v,r)

return x*u%r
```

Figure 4.4: Реализация алгоритма Ро-метода Полларда для логарифмирования

Были взяты данные из пояснения к лабораторной работе. Они были подставлены в программу. Получен следующий результат (см. рис. [-fig. 4.5).

```
Pollard(107,10,53,64,2,2)
```

```
20
```

Figure 4.5: Результат реализации Ро-метода Полларда на примере

5 Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: в результате выполнения данной лабораторной работы нам удалось изучить алгоритм Ро-Полларда осуществить программно алгоритм, рассмотренный в описании к лабораторной работе на языке Python 3. А также получить ответ, совпадающий с ответом из инструкции.

Список литературы

1. Википедия. ро-метод Полларда для дискретного логарифмирования [Электронный ресурс]. Википедия, свободная энциклопедия, 2021. URL: https://ru.wikipedia.org/wiki/%D0%A0%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0_%D0%B4%D0%BB%D1%8F_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F (дата обращения: 25.12.2021).
2. Wikiznanie. ро-метод Полларда для дискретного логарифмирования [Электронный ресурс]. Википедия, 2021. URL: https://www.wikiznanie.ru/wp/index.php/%D0%A0%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0_%D0%B4%D0%BB%D1%8F_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F (дата обращения: 25.12.2021).

3. Wikiznanie. ⊠-метод Полларда дискретного логарифмирования [Электронный ресурс]. Википедия, 2021. URL: https://mind-control.fandom.com/wiki/%CE%A1-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F (дата обращения: 25.12.2021).