ЛАБОРАТОРНА РОБОТА №2

з курсу

ТЕОРЕТИКО-ЧИСЛОВІ АЛГОРИТМИ В КРИПТОЛОГІЇ

Застосування алгоритму дискретного логарифмування

Мета роботи

Ознайомлення з алгоритмом дискретного логарифмування Сільвера-Поліга-Геллмана. Практична реалізація цього алгоритму. Пошук переваг, недоліків та особливостей застосування даного алгоритму дискретного логарифмування. Практична оцінка складності роботи алгоритму.

Хід роботи

Напишемо програму, що розв'язує задачу дисктерного логарифму шляхом звичайного перебору. Протестуємо

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 DLPBrute.py
Input base a : 583
Input result b : 380
Input mod p : 773
Bruteforce: Answer to equation 583^x = 380 mod 773 is 548. Time taken 0.00040984153
747558594

┌──(kali㉿kali)-[~/Desktop]
└─$ python3 DLPBrute.py
Input base a : 6
Input result b : 26
Input mod p : 103
Bruteforce: Answer to equation 6^x = 26 mod 103 is 10. Time taken 7.033348083496094
e-05
```

Напишемо програму, що реалізовує алгоритм Сільвера-Поліга-Геллмана для груп типу Z_p^* і протестуємо

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 Python-svarnyk-fb13-medvetskyi-fb13.py
Input base a : 583
Input result b : 380
Input mod p : 773
SPH: Answer to equation 583^x = 380 mod 773 is 548. Time taken 0.00012946128845214844

┌──(kali㉿kali)-[~/Desktop]
└─$ python3 Python-svarnyk-fb13-medvetskyi-fb13.py
Input base a : 6
Input result b : 26
Input mod p : 103
SPH: Answer to equation 6^x = 26 mod 103 is 10. Time taken 0.00015211105346679688
```

Тепер перейдемо до замірів часу і порівняння алгоритмів. Використовуючи допоміжну програму, згенеруємо задачі і протестуємо програми

```
Task Type 1:
a = 42476779;
b = 40822421;
p = 60990847.
2024-04-23 11:39:58 Please, found the discrete logarithm: a ^ x = b mod p. You have 5 minut
es, starting now.
Enter x value: x = 22751125


2024-04-23 11:40:29 BINGO!!! You solve the discrete logarithm correct.
Next, please, solve the type 2 task.


Task Type 2:
a = 9876342;
b = 3738953;
p = 12745483.
2024-04-23 11:40:29 Please, found the discrete logarithm: a ^ x = b mod p. You have 5 minut
es, starting now.
Enter x value: x = 7584915


2024-04-23 11:40:57 BINGO!!! You solve both tasks correct. You can be proud of yourself!
You can try again with bigger prime number decimal digit length. If you know what I mean ;)


2024-04-23 11:40:57 Tool closed.
```

SPH:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 Python-svarnyk-fb13-medvetskyi-fb13.py
Input base a : 42476779
Input result b : 40822421
Input mod p : 60990847
SPH: Answer to equation 42476779^x = 40822421 mod 60990847 is 22751125. Time taken 0.00044
75116729736328

┌──(kali㉿kali)-[~/Desktop]
└─$ python3 Python-svarnyk-fb13-medvetskyi-fb13.py
Input base a : 9876342
Input result b : 3738953
Input mod p : 12745483
SPH: Answer to equation 9876342^x = 3738953 mod 12745483 is 7584915. Time taken 2.06719636
91711426
```

Bruteforce:

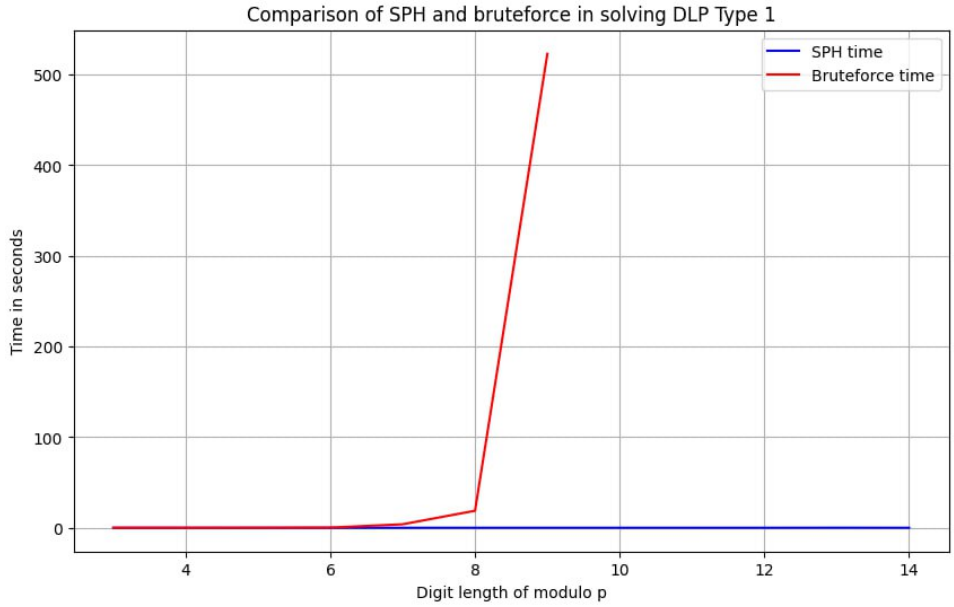
```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 DLPBrute.py
Input base a : 42476779
Input result b : 40822421
Input mod p : 60990847
Bruteforce: Answer to equation 42476779^x = 40822421 mod 60990847 is 22751125. Time
 taken 18.825295448303223

┌──(kali㉿kali)-[~/Desktop]
└─$ python3 DLPBrute.py
Input base a : 9876342
Input result b : 3738953
Input mod p : 12745483
Bruteforce: Answer to equation 9876342^x = 3738953 mod 12745483 is 7584915. Time ta
ken 5.9993157386779785
```

В результаті серії тестів, отримали таблички

	Кількість знаків p	СПГ	Перебір
Тип 1	3	0.00012	0.0004
	4	0.00024	0.00056
	5	0.00016	0.032
	6	0.0042	0.23
	7	0.0064	3.89
	8	0.00044	18.82
	9	0.013	523
	10	0.01	-
	11	0.0048	-
	12	0.0041	-
	13	0.068	-
	14	0.013	-

	Кількість знаків p	СПГ	Перебір
Тип 2	3	0.00015	0.00007
	4	0.0016	0.00054
	5	0.00058	0.0076
	6	0.096	0.093
	7	1.35	4.23
	8	2.06	5.99
	9	6.93	4.31
	10	11.72	-
	11	106	-
	12	0.851	-
	13	7.35	-
	14	-	-

Візуалізація



Висновок

Перший тип: до 6 знаків різниця між алгоритмами невелика, проте після 7 знаків час вирішення задачі суттєво збільшується для алгоритма перебору. Оскільки в першому типі задач спільне між (p-1) те, що вони розкладаються в невеликі прості множники і саме в цьому випадку SPH працює найбільш ефективно, в незалежності від довжини модуля, в той час як ефективність роботи алгоритму перебору напряму залежить від порядку модуля: чим більший порядок - тим більший інтервал перебору. Як видно з графіку, алгоритм перебору не вкладається в адекватний час, а SPH працює добре як мінімум до 14 знаків

Другий тип: до 9 знаків різниця не суттєва, після 10 знаків спостерігаємо погіршення роботи SPH, в той час як метод перебору взагалі перестає справлятися. Погіршення в SPH зумовлено тим, що серед дільників (p-1) зустрічається велике просте число і задача зводиться до перебору.

Загалом, метод перебору ефективний тільки у випадку, коли шукане x є невеликим числом (на тестах до 10 знаків). Для першого типу задач алгоритм SPH виявився дуже ефективним, в той час як для другого типу задач - не набагато краще ніж перебір. Випадок для 12/13 знаків в другому типі можемо враховувати як удачу, бо не було великих простих дільників серед (p-1).

```
Task Type 2:
a = 32534600804;
b = 2383512058454;
p = 3609407314207.
2024-04-23 12:23:27 Please, found the discrete logarithm: a ^ x = b
 mod p. You have 5 minutes, starting now.
Enter x value: x = 1207078320620


2024-04-23 12:25:21 BINGO!!! You solve both tasks correct. You can
be proud of yourself!
You can try again with bigger prime number decimal digit length. If
 you know what I mean ;)
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 Python-svarnyk-fb13-medvetskyi-fb13.py
Input base a : 32534600804
Input result b : 2383512058454
Input mod p : 3609407314207
SPH: Answer to equation 32534600804^x = 2383512058454 mod 3609407314207 is 1207078320620.
 Time taken 7.356338739395142
```

Input

3 609 407 314 206

Prime factorization

$2 \times 3^6 \times 1499 \times 1651493$