Difi-Helman nulaužimo (bendrojo slaptojo rakto gavimo) algoritmas

Kodas:

Diskretaus logaritmo radimas:

```
from quickPow_7LD import getQuickRemainder
from reverseMod_6LD import calcReverseMod
from math import sqrt

def bruteForceLog(g, gxmod, n):
    gj = 0
    j = 1
    while gj != gxmod:
        j += 1
        gj = getQuickRemainder(g, j, n)
    return j

def BstepGstepLog(g, gxmod, n):
    m = int(sqrt(n)) + 1
    B = []
    for i in range(m):
        gamaR = calcReverseMod(g, n)
        gamaR = getQuickRemainder(gamaR, i, n)
        agr = (gxmod * gamaR) % n
        if agr == 1:
            return i
        else:
            B.append([agr, i])
    delta = getQuickRemainder(g, m, n)

    for q in range(1, 1000000):
        deltaQ = getQuickRemainder(delta, q, n)
        for pair in B:
            if pair[0] == deltaQ:
                return q*m + pair[1]

    print(TimeoutError)
    exit()
```

Parašiau kruopštaus parinkimo algoritmą, nes jo pilnai užtenka, kad atlikti duotą užduotį, bet taip pat realizavau ir „Baby-step Giant-step“ algoritmą, nes be jo šis laboratorinis atrodė per daug paprastas.

Nulaužimo algoritmas:

```
def getDHSecretKey(g, n, gb, ga):
    gabmod = 0

    if (gb < ga):
        gbmod = gb % n
        # b = bruteForceLog(g, gbmod, n)
        b = BstepGstepLog(g, gbmod, n)
        gabmod = getQuickRemainder(ga, b, n)
    else:
        gamod = ga % n
        # a = bruteForceLog(g, gamod, n)
        a = BstepGstepLog(g, gamod, n)
        gabmod = getQuickRemainder(gb, a, n)

    print (gabmod)
    print ("---------------")
    return gabmod
```

Patikrina kuris tarp gavėjo ir siutėjo skaičių mažesnis ir randa jo logaritmą, atkomentavus atitinkamas eilutes galima naudoti brute force arba baby-step giant-step algoritmą. Tada greitojo kėlima laipsnio algorimu randamas ir išvedamas $(g^b)^a$ arba $(g^a)^b$ priklausomai nuo to kurio skaičiaus logaritmo buvo ieškoma.

Išvedimas:

```
g = 7
n = 131071
gb = 79792266297612001
ga = 11398895185373143
getDHSecretKey(g, n, gb, ga)

g = 7
n = 331
gb = 79792266297612001
ga = 11398895185373143
getDHSecretKey(g, n, gb, ga)
```

Pirmame variante naudojami jūsų duoti užduoties skaičiai, o antrame variante panaudojau tokius pačius g, GS ir SS kaip pirmame, bet pakeičiau n kad jis turėtų panaudoti giant-step, o ne tik baby-step ir patikrinau ar gaunamas geras atsakymas su brute force algoritmu.

Rezultatai: