Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Лабораторная работа № 4

“Компьютерная реализация алгоритма шифрования Эль-Гамаля”

Выполнил студент гр. 753504

Горбачёнок К. Н.

Проверил

Протько М.И.

Минск, 2020

# Постановка задачи

Реализовать программные средства шифрования и дешифрования текстовых файлов при помощи aлгоритма шифрования Эль-Гамаля.

# Описание алгоритма

Схема Эль-Гамаля (Elgamal) — криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94).

**Генерация ключей**

* Генерируется случайное простое число p.
* Выбирается целое число g — первообразный корень p.
* Выбирается случайное целое число x такое, что 1 < x < p - 1.
* Вычисляется y = g^x mod p.
* Открытым ключом является y, закрытым ключом — число x.

**Шифрование**

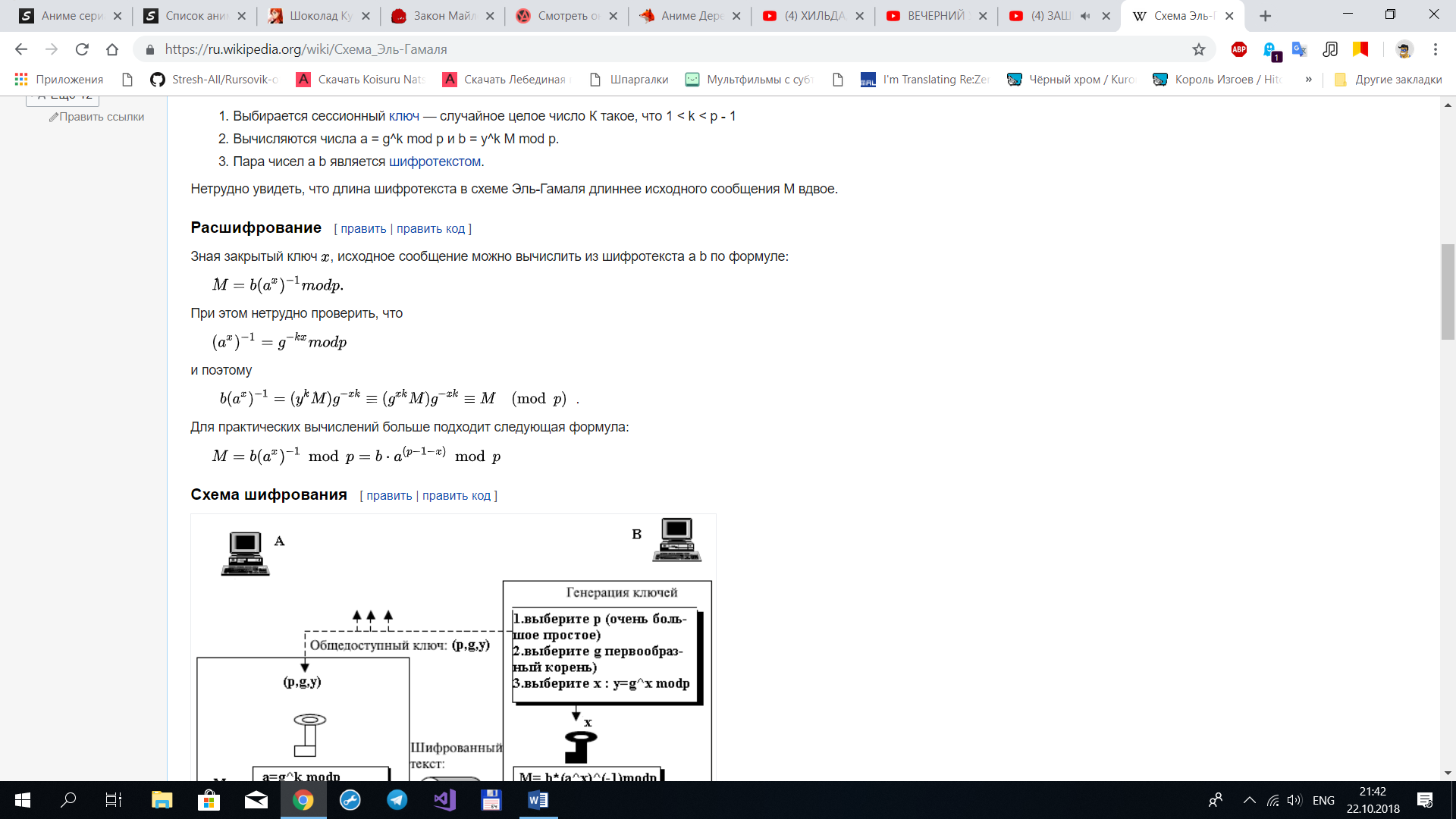
Сообщение M должно быть меньше числа p. Сообщение шифруется следующим образом:

1. Выбирается сессионный ключ — случайное целое число К такое, что

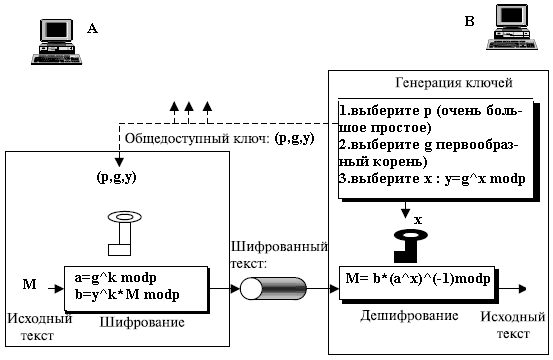
1 < k < p - 1

1. Вычисляются числа a = g^k mod p и b = y^k M mod p.
2. Пара чисел a b является шифротекстом.

Нетрудно увидеть, что длина шифротекста в схеме Эль-Гамаля длиннее исходного сообщения M вдвое.

**Расшифрование**

# Блок-схема алгоритма



***Блок-схема алгоритма Эль-Гамаля***

# Результаты работы программы

# 

***Исходный файл***

******

***Результат работы программы***

# Прграммный код

import random

from math import pow, gcd

def gen\_key(q):

key = random.randint(pow(10, 20), q)

while gcd(q, key) != 1:

key = random.randint(pow(10, 20), q)

return key

def power(a, b, c):

x = 1

y = a

while b > 0:

if b % 2 == 0:

x = (x \* y) % c

y = (y \* y) % c

b = int(b / 2)

return x % c

def encode(message, q, h, g):

en\_msg = []

k = gen\_key(q)

s = power(h, k, q)

p = power(g, k, q)

for i in range(0, len(message)):

en\_msg.append(message[i])

for i in range(0, len(en\_msg)):

en\_msg[i] = s \* ord(en\_msg[i])

return en\_msg, p

def decode(en\_message, p, key, q):

dr\_message = []

h = power(p, key, q)

for i in range(0, len(en\_message)):

dr\_message.append(chr(int(en\_message[i] / h)))

return dr\_message

import random

import elgamal

def file\_info(filename="file.txt"):

with open(filename) as file\_handler:

text = file\_handler.readline().strip()

return text

def main():

q = random.randint(pow(10, 20), pow(10, 50))

g = random.randint(2, q)

key = elgamal.gen\_key(q)

h = elgamal.power(g, key, q)

text = file\_info()

encoded\_text, p = elgamal.encode(text, q, h, g)

print("\033[95m\nENCODED:\033[0m ", ''.join(str(i) for i in encoded\_text))

decoded\_text = elgamal.decode(encoded\_text, p, key, q)

print("\033[95m\nDECODED:\033[0m ", "".join(decoded\_text))

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Вывод

В настоящее время криптосистемы с открытым ключом считаются наиболее перспективными. К ним относится и схема Эль-Гамаля, криптостойкость которой основана на вычислительной сложности проблемы дискретного логарифмирования, где по известным p, g и y требуется вычислить x, удовлетворяющий сравнению: *y* ≡ *gx*(*mod p*)

В ходе написания лабораторной работы был изучен алгоритм шифрования и дешифрования Эль-Гамаля, а также написана программная реализация.