**Введение**

Схема Эль-Гамаля (Elgamal) — [криптосистема**​**](https://ru.wikipedia.org/wiki/Криптосистема) с открытым ключом,

основанная на трудности вычисления [дискретных**​** логарифмов](https://ru.wikipedia.org/wiki/Дискретный_логарифм) в [конечном​](https://ru.wikipedia.org/wiki/Конечное_поле)

[поле](https://ru.wikipedia.org/wiki/Конечное_поле).**​** Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов [электронной цифровой подписи](https://ru.wikipedia.org/wiki/Электронная_цифровая_подпись)**​**в [США​](https://ru.wikipedia.org/wiki/США)(**​**[DSA​](https://ru.wikipedia.org/wiki/DSA))**​** и [России​](https://ru.wikipedia.org/wiki/Россия)​([ГОСТ**​** Р 34.10-94)](https://ru.wikipedia.org/wiki/ГОСТ_Р_34.10-94)**​**.

Схема была предложена [Тахером**​** Эль-Гамалем](https://ru.wikipedia.org/wiki/Тахер_Эль-Гамал) в [1985**​** году](https://ru.wikipedia.org/wiki/1984_год).**​**

Эль-Гамаль разработал один из вариантов [алгоритма**​** Диффи-Хеллмана](https://ru.wikipedia.org/wiki/Алгоритм_Диффи_—_Хеллмана).**​** Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

В рамках лабораторной работы необходимо реализовать программные средства шифрования и дешифрования при помощи алгоритма Эль-Гамаля.

# **Теоретическая справка**

Генерация ключей:

1. Генерируется случайное простое число p.
2. Выбирается целое число g — первообразный корень p.
3. Выбирается случайное целое число x такое, что 1<x<p-1.
4. Вычисляется y = gx mod p.
5. Открытым ключом является y, закрытым ключом — число x.

Шифросистема Эль-Гамаля является фактически одним из способов выработки открытых ключей Диффи — Хеллмана. Шифрование по схеме Эль-Гамаля не следует путать с алгоритмом цифровой подписи по схеме Эль-Гамаля.

**Шифрование**

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

1. Выбирается сессионный ключ — случайное целое число k такое, что 1<k<p-1.
2. Вычисляются числа a=gk mod p и b=yk \* M mod p.
3. Пара чисел (a,b) является шифротекстом.

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

**Расшифрование**

Зная закрытый ключ x, исходное сообщение можно вычислить из шифротекста (a,b) по формуле: M=b\*(ax)-1 mod p

При этом нетрудно проверить, что (ax)-1 = g –kx mod p

и поэтому b\*(ax)-1 = (yk\*M) \* g –kx = (gxk \* M) \* g –kx = M (mod p).

Для практических вычислений больше подходит следующая формула: M=b (ax)-1 = ba p-1-x (mod p).

**Блок-схема алгоритма**

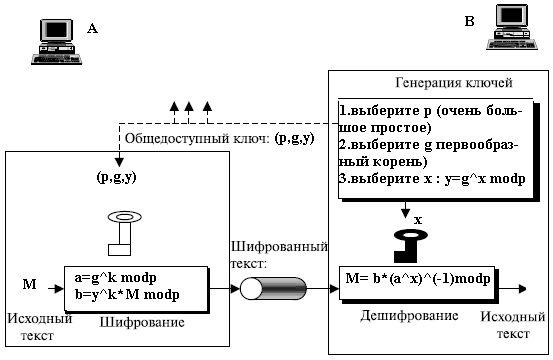


Рис.1. Схема алгоритма

**Пример работы программы**

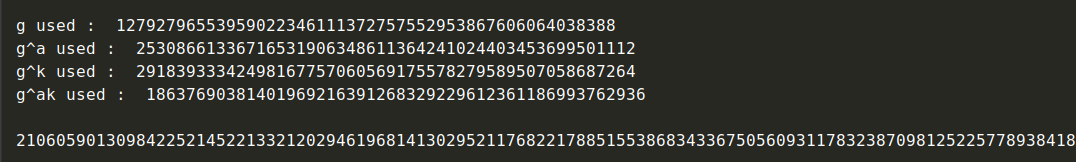


Рис.2. Пример работы

**Код программы**

***import* random**

*from* math *import* pow

a = random.randint(2, 10)

*def* gcd(*a*, *b*):

*if a* < *b*:

*return* gcd(*b*, *a*)

*elif a* % *b* == 0:

*return b*

*else*:

*return* gcd(*b*, *a* % *b*)

# Generating large random numbers

*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

# Modular exponentiation

*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*

# Asymmetric encryption

*def* encrypt(*message*, *q*, *h*, *g*):

en\_msg = []

k = gen\_key(*q*) # Private key for sender

s = power(*h*, k, *q*)

p = power(*g*, k, *q*)

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

en\_msg.append(*message*[i])

print("g^k used : ", p)

print("g^ak used : ", s)

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

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

*return* en\_msg, p

*def* decrypt(*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

**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*)

* ходе написания лабораторной работы были изучены алгоритмы шифрования и дешифрования Эль-Гамаля, а также написаны их программные реализации.