МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

НГТУ НЭТИ

Кафедра прикладной математики

Практическое задание № 2

по дисциплине «Основы криптографии»

Алгоритм Эль-Гамаля

Группа	ПМ-81
Бригада	3
	БАСОВ ДЕНИС
	ЮРГАНОВ ЕГОР
Преподаватель	СТУПАКОВ ИЛЬЯ МИХАЙЛОВИЧ
Дата	02.12.2020

Новосибирск

Часть 1

1. Цель:

Знакомство с ассиметричными криптографическими алгоритмами.

2. Задача:

Реализовать алгоритм Эль-Гамаля

3. Алгоритм

Написать программы реализующие алгоритм Эль-Гамаля.

1.Генерация ключей

Прочитать из консоли числа p и g.

Проверить что p простое число, а g первообразный корень по модулю p.

Сгенерировать закрытый ключ x и открытый ключ y по алгоритму Эль-Гамаля

2.Шифрование текста

Прочитать из консоли числа p, g и y.

Прочитать сообщение M (число меньшее p), зашифровать его алгоритмом Эль-Гамаля и вывести результат.

3.Расшифровка текста

Прочитать из консоли числа p, g и x.

Прочитать зашифрованное сообщение, расшифровать его алгоритмом Эль-Гамаля и вывести результат.

4. Программа

Subroutine.h

```
#pragma once

#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <vector>
#include <iostream>
#include <random>

using namespace std;

uint64_t modpow(uint64_t a, uint32_t b, uint32_t c)
{
	uint64_t t = 1;
	while (b) {
		if (b % 2 == 0) {
			b /= 2;
			a = (a * a) % c;
		}
		else {
			b--;
			t = (t * a) % c;
		}
	}
	return t;
}

bool PrimeN(uint32_t p) {
	uint64_t t = 2;
	while ((t*t <= p) && (p % t != 0)) t++;
	if (t*t > p) return true;
```

```
	else return false;
}

uint32_t random(uint32_t p, uint32_t &a){
	random_device rd;
	mt19937 gen(rd());
	uniform_int_distribution<uint32_t> dist(2, p-2); // тк 1 < x < p-1
	a = dist(gen);
	return a;
}

vector<uint32_t> factorize(uint32_t x) {
	vector<uint32_t> factors;
	for (int i = 2; i <= x; i++) {
		if ((x % i == 0) && (PrimeN(i))) {
			factors.push_back(i);
			x /= i;
		}
	}
	return factors;
}

bool root(uint32_t p, uint32_t g) {
	if (modpow(g, p - 1, p) == 1) {
		vector<uint32_t> dividers;
		dividers = factorize(p - 1);
		for (int j : dividers) {
			//cout << j << "\n";
			if(modpow(g, (p-1)/j, p)==1)
				return false;
		}
		return true;
	}
	else return false;
}

uint32_t NOD(uint32_t a, uint32_t b) {
	return b ? NOD(b, a % b) : a;
}

void keys(uint32_t p, uint32_t g, uint32_t &y, uint32_t &x) {
	x = 0;
	while (NOD(x, p - 1) != 1)
		x = random(p, x);
	y = modpow(g, x, p);
}

void encrypt(uint32_t  M, uint32_t p, uint32_t g, uint32_t y, uint32_t &a, uint32_t &b) {
	uint32_t k = 0;
	while (NOD(k, p - 1) != 1)
		k = random(p, k);
	a = modpow(g, k,p);
	b = (modpow(y, k, p) * M) %p ;
}

void decrypt(uint32_t p, uint32_t g, uint32_t x, uint32_t a, uint32_t b, uint32_t&M) {
	M = (modpow(a, (p - 1 - x), p) * b) % p;
}
```

Main.cpp

```
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <string>
#include <fstream>
```

```cpp
#include <iostream>
#include "subroutine.h"


using namespace std;


int main() {
	uint32_t p, g, y, x, f=1, a, b, M;
	while (f) {
		cout << "enter p, g\n";
		cin >> p >> g;
		//cout << modpow( g, p-1, p);
		if ((PrimeN(p)) && (root(p, g))) {
			f = 0;
			keys(p, g, y, x);
			cout << "keys x,y " << x << " " << y << "\n";
		}
		else printf("\nwrong p or g\n");
	}

	cout << "enter M (M<p) " ;
	cin >> M;

	cout << "enter p g y ";
	cin >> p >> g >> y;

	encrypt(M, p, g, y, a, b);
	cout << "a,b " << a << " " << b << "\n";

	cout << "enter p g x a b" << "\n";
	cin >> p, g, x, a, b;
	decrypt(p, g, x, a, b, M);
	cout << M << "\n";
}
```

5. Тесты

Тест в верными входными данными

```
enter p, g
23
21
keys x,y 19 20
enter M (M<p) 14
enter p g y 23 21 20
a,b 14 2
enter p g x a b
23 21 19 14 2
14
```

Тест с неверными входными данными

```
enter p, g
15
2

wrong p or g
enter p, g
11
2
keys x,y 9 6
enter M (M<p) 7
enter p g y 11 2 6
a,b 2 9
enter p g x a b
11 2 9 2 9
7
```

Тест с p = 2^32 – 1

```
enter p, g
2147483647
122
1keys x,y 161108245 1782677507
enter M (M<p) 100012324
enter p g y 2147483647 122 1782677507
a,b 580574449 200127024
enter p g x a b
2147483647 122 161108245 580574449 200127024
100012324
```