Харківський національний університет імені В.Н. Каразіна
Навчально-науковий інститут комп’ютерних наук та штучного інтелекту

ЗВІТ

З ПРАКТИЧНОЇ РОБОТИ №13

дисципліна: «Алгоритмізація та програмування»

Виконав: студент 2 курсу групи КС22
Спеціальності 122 «Комп’ютерні науки»
Скрипняк Тарас Артемович
Прийняв: викладач
Олешко О.І.

Харків – 2024

Завдання №1: Реалізувати алгоритм несиметричного шифрування RSA.
Функції програми:
- Шифрування файлу (вказати ім'я)
- Розшифрування файлу (вказати ім'я)

Ключі шифрування/дешифрування фіксовані та задані у програмі.

Бонусні бали за генерацію ключової пари.

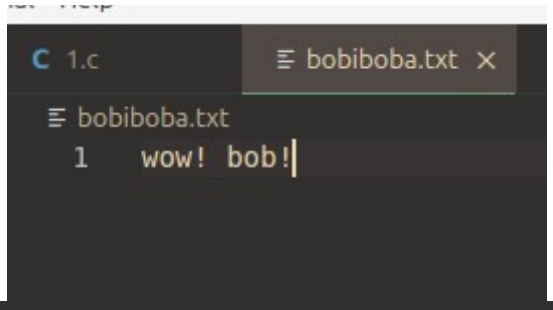
Рекомендується вибирати модуль перетворення N коротким, 15-17 біт завдовжки.
У цьому випадку для піднесення до степеня числа по модулю немає потреби в реалізації арифметики багатократної точності і можна користуватися звичайними арифметичними операціями процесора.
Для функції генерації ключів вам знадобиться навчитися вирішувати діофантове рівняння виду ax + by = c

```
#include <stdio.h>
#include <stdlib.h>

// Прості числа для генерації ключів
const int p = 61;
const int q = 53;
const int N = p * q;
const int phi = (p - 1) * (q - 1);
const int e = 17; // Публічний ключ
int d = 0;        // Приватний ключ

// Функція обчислення НСД (алгоритм Евкліда)
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

// Функція обчислення мультиплікативного оберненого (розширений алгоритм
Евкліда)
int mod_inverse(int a, int m) {
    int m0 = m, y = 0, x = 1;
    while (a > 1) {
        int q = a / m;
        int t = m;
        m = a % m;
        a = t;
        t = y;
        y = x - q * y;
        x = t;
    }
    if (x < 0) x += m0;
    return x;
}
```

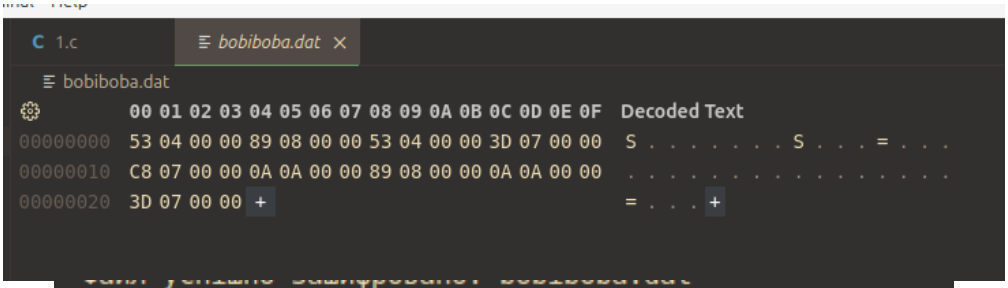
```
// Функція для піднесення до степеня за модулем
int mod_exp(int base, int exp, int mod) {
    int result = 1;
    base = base % mod;
    while (exp > 0) {
        if (exp % 2 == 1)
            result = (result * base) % mod;
        exp = exp >> 1;
        base = (base * base) % mod;
    }
    return result;
}

// Функція шифрування файлу
void encrypt_file(const char *input_file, const char *output_file) {
    FILE *fin = fopen(input_file, "rb");
    FILE *fout = fopen(output_file, "wb");
    if (!fin || !fout) {
        printf("Помилка відкриття файлу.\n");
        return;
    }

    int ch;
    while ((ch = fgetc(fin)) != EOF) {
        int encrypted_char = mod_exp(ch, e, N);
        fwrite(&encrypted_char, sizeof(int), 1, fout);
    }

    fclose(fin);
    fclose(fout);
    printf("Файл успішно зашифровано: %s\n", output_file);
}

// Функція розшифрування файлу
void decrypt_file(const char *input_file, const char *output_file) {
    FILE *fin = fopen(input_file, "rb");
    FILE *fout = fopen(output_file, "wb");
    if (!fin || !fout) {
        printf("Помилка відкриття файлу.\n");
        return;
    }

    int encrypted_char;
    while (fread(&encrypted_char, sizeof(int), 1, fin) > 0) {
        int decrypted_char = mod_exp(encrypted_char, d, N);
        fputc(decrypted_char, fout);
    }

    fclose(fin);
    fclose(fout);
    printf("Файл успішно розшифровано: %s\n", output_file);
}

// Функція генерації ключів
void generate_keys() {
    if (gcd(e, phi) != 1) {
        printf("Помилка: e і φ(N) не взаємно прості.\n");
        return;
    }
    d = mod_inverse(e, phi);
    printf("Пара ключів згенерована:\n");
    printf("Публічний ключ (e, N): (%d, %d)\n", e, N);
    printf("Приватний ключ (d, N): (%d, %d)\n", d, N);
}
```

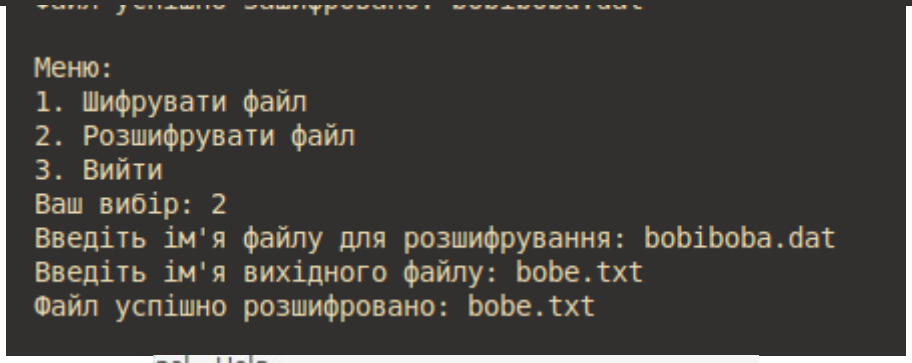
```c
int main() {
    generate_keys();

    char input_file[100], output_file[100];
    int choice;

    while (1) {
        printf("\nМеню:\n");
        printf("1. Шифрувати файл\n");
        printf("2. Розшифрувати файл\n");
        printf("3. Вийти\n");
        printf("Ваш вибір: ");
        scanf("%d", &choice);

        if (choice == 1) {
            printf("Введіть ім'я файлу для шифрування: ");
            scanf("%s", input_file);
            printf("Введіть ім'я вихідного файлу: ");
            scanf("%s", output_file);
            encrypt_file(input_file, output_file);
        } else if (choice == 2) {
            printf("Введіть ім'я файлу для розшифрування: ");
            scanf("%s", input_file);
            printf("Введіть ім'я вихідного файлу: ");
            scanf("%s", output_file);
            decrypt_file(input_file, output_file);
        } else if (choice == 3) {
            break;
        } else {
            printf("Неправильний вибір. Спробуйте ще раз.\n");
        }
    }

    return 0;
}
```

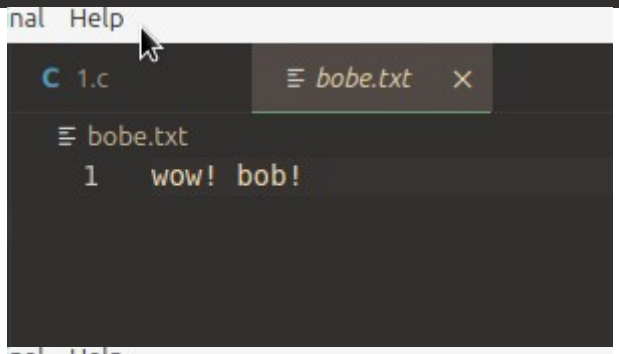
Лістинг - вихідний код програми

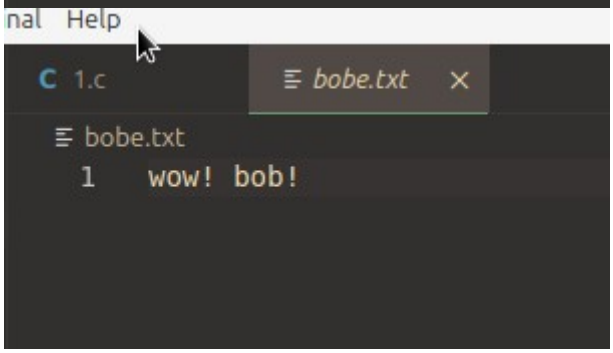



```
bouncytorch@AORUS:~/Repos/homework-c/pr13/tarik$ ./a.out
Пара ключів згенерована:
Публічний ключ (e, N): (17, 3233)
Приватний ключ (d, N): (2753, 3233)

Меню:
1. Шифрувати файл
2. Розшифрувати файл
3. Вийти
Ваш вибір: 1
Введіть ім'я файлу для шифрування: bobiboba.txt
Введіть ім'я вихідного файлу: bobiboba.dat
Файл успішно зашифровано: bobiboba.dat
```









Рисунки 1, 2, 3, 4, 5, 6 - результат виконання програми