**Міністерство освіти та науки України**

**НТУУ «КПІ ім. І.Сікорського»**

**Кафедра АПЕПС**

Звіт до лабораторної роботи №7

з предмету: «Безпека програм і даних»

на тему: «Шифрування даних методом гаміювання»

12 варіант

Виконав: студент 4 курсу  
 ТЕФ, ТІ-31  
 Ніколаєв В.В.

Перевірив: Тарнавський Ю.А.

Київ-2016

Цілі роботи:

Ознайомитись з методом шифрування даних на основі гаміювання і надати його програмну реалізацію.

Алгоритм шифрування:

1. Нумеруємо букви вибраного алфавіту для шифрування.

2. Кожному символу M вихідного повідомлення ставимо у відповідність номер m з вибраного алфавіту.

3. Конструюємо генератор псевдовипадкових чисел (ПВЧ).

4. Задаємо параметри генератора ПВЧ в якості секретного ключа.

5. Генеруємо послідовність ПВЧ - гаму, для якої Т> L, де Т - період гами L - довжина повідомлення, що шифрується.

6. Накладаємо гаму на повідомлення, що шифрується: l = m XOR ПВЧ. Тут l - код символу криптограми, ПВЧ - випадковий номер гами для символу М.

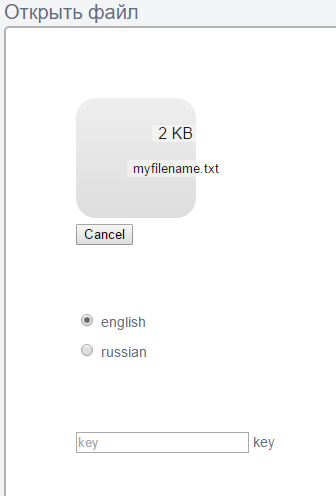
Алгоритм розшифрування:

1. Генеруємо гаму за допомогою секретного ключа.

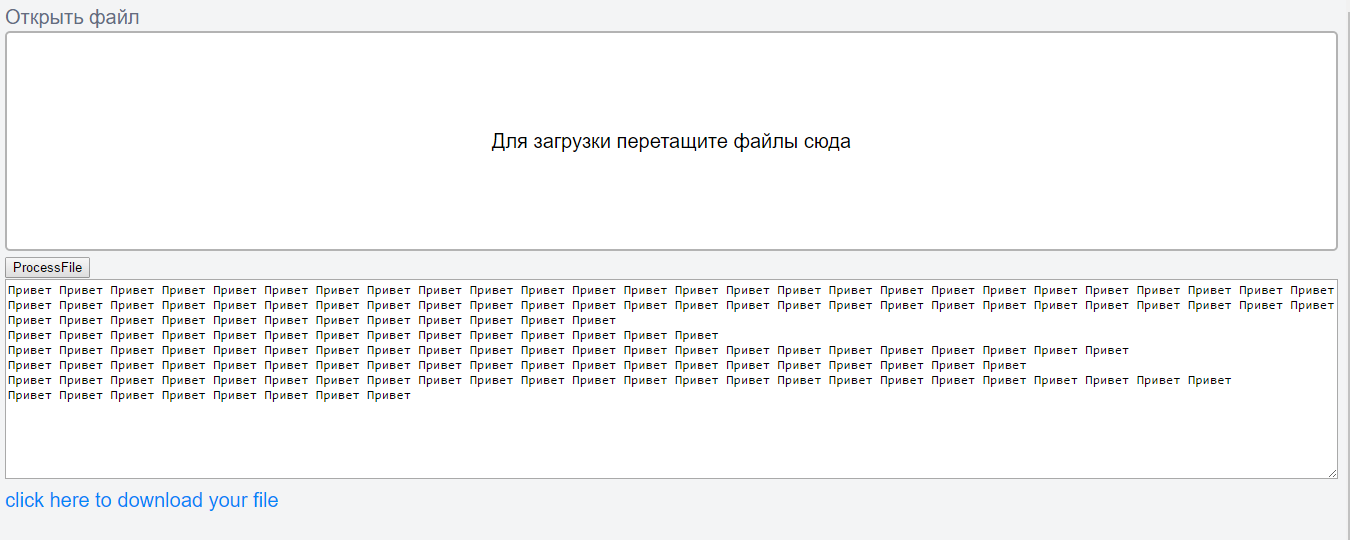
2. Виконуємо гаміювання криптограми.

3. Перекодовуємо повідомлення з цифрового виду в символьний відповідно до обраного алфавіту.

В розробленому програмному продукті був реалізований інтерфейс Драг-Н-Дроп для полегшення відкриття необхідних файлів користувачем.



Користувачу надається можливість введення ключа для кодування/ декодування та вибір алфавіту кодування/ декодування. Після вибору опцій програма опрацьовує вхідні дані на основі алгоритму гаміювання.  
 Користувач має змогу скачати результат шифрування в текстовому файлі для подальшого використання.



Код програми

function Encoder(alph,key){

alphabet=[];//алфавіт

size=0;//розмір алфавіту

/\*\*

функція кодування/ декодування

str - вхідні дані (текст)

decode - флаг декодування

\*/

this.encode=function(str,flag){

var result="";

var gamma=pseudoRand(str.length);

for(var i=0;i<str.length;i++){

var m=alphabet.indexOf(str[i]);

if(m==-1){

throw new Error('Unknown symmbol! '+ str[i])

}

var l=m ^ gamma[i];

result+=alphabet[l];

}

return result;

}

//ЗАПОВНЕННЯ АЛФАВІТУ

if(alph=="english"){

for(var i='A';;i=nextChar(i)){

alphabet.push(i);

if(i=='z')

break;

}

}else if(alph=="russian"){

for(var i='А';;i=nextChar(i)){

alphabet.push(i);

if(i=='я')

break;

}

}

else{

throw new Error('Unimplemented alphabet '+alph);

}

for(var i=0;i<10;i++){

alphabet.push(i+'');

}

alphabet.push(' ','\n','=','-','\_','+','\*','/','!');

size=alphabet.length;

//console.log(alphabet)

//console.log(this.alphabet);

function pseudoRand(leng){

var num=35593571; // очень большое простое число.

//key

var seed=key; // затравка - то самое инициализирующее число.

var start\_number=10; // основа

var counts = new Array();

var max\_num = 10; // Инт, который не превосходит наше число. У вас 0-9 числа.

for(var i=0; i<=leng/\*1 more that alphabet size\*/; i++){

counts[i] = (Math.pow(start\_number+i,seed)%num)%max\_num;

}

return counts;

}

/\*\*

Отримуємо символ з кодом більшим на 1

\*/

function nextChar(c) {

return String.fromCharCode(c.charCodeAt(0) + 1);

}

}

Висновки:

Було вивчено алгоритм вищевказаного шифру та реалізовано його з відповідністю до його специфікації. Був розроблений дружній інтерфейс користувача для спрощення роботи з програмою.