**OIB || lab\_1**

## файл с дополнительной информацией для лабораторной работы ##

**[Алгоритм Меркля-Хеллмана:]**

Основные параметры ключей:

* **w[KEY\_SIZE]** — сверхвозрастающая последовательность, используемая в качестве закрытого ключа.  
    
  *Сверхвозрастающая последовательность — это такая последовательность чисел, где каждый элемент больше суммы всех предыдущих.*
* **n** — модуль, используемый в преобразовании закрытого ключа в открытый. *(модуль, который должен быть больше суммы всех чисел в w)*
* **e** — множитель, который является частью открытого ключа и должен быть взаимно прост с n. (*gcd(e, n) = 1*)
* **beta[KEY\_SIZE]** — массив, содержащий элементы открытого ключа, вычисляемые по формуле beta[i] = (w[i] \* e) mod n.
* **d** — обратный элемент к e по модулю n, используемый при расшифровании (вычисляется с помощью mod\_inverse).

**Переменные для шифрования и дешифрования**

* **message[KEY\_SIZE]** — двоичный массив, представляющий сообщение перед шифрованием.
* **c** — зашифрованное сообщение (шифртекст), вычисляемое по формуле c = sum(message[i] \* beta[i]).
* **s** — вспомогательное значение при расшифровании, вычисляемое как s = (c \* d) mod n.
* **decrypted[KEY\_SIZE]** — массив, в котором сохраняется расшифрованное двоичное сообщение.

**Вспомогательные переменные**

* **sum** — используется при генерации закрытого ключа для формирования сверхвозрастающей последовательности.
* **temp** — вспомогательная переменная в функции gcd для вычисления наибольшего общего делителя.
* **x** — перебираемое значение в mod\_inverse для поиска обратного элемента по модулю.
* **i** — счетчик циклов в различных функциях.

1. **Генерация открытого ключа**

**Открытый ключ** получается путём *умножения* **каждого элемента w[i]** на **e** по модулю **n**:

*В результате получаем псевдослучайные числа, которые не образуют сверхвозрастающую последовательность.*

1. **Шифрование сообщения**

Сообщение — это бинарный массив (например, 10110010).

Чтобы зашифровать его, мы *суммируем* **элементы открытого ключа beta[i],** взятые там, где **message[i]** **== 1**:

Результат — **одно число c**, которое отправляется как зашифрованное сообщение.

1. **Дешифрование (нахождение d)**

Чтобы *расшифровать* **c**, нам нужно вычислить:

s=(c⋅d) mod  n

s=(c⋅d) mod n

где **d** — **обратный элемент** для **e** по модулю **n**, то есть:

d= e^(−1) (mod n)

После этого **s** является **суммой подмножества** **w**, а так как **w** — сверхвозрастающая, можно восстановить оригинальное сообщение.

1. **Восстановление сообщения**

Так как **s** является суммой подмножества **w**, мы можем восстановить **message[i],** начиная с самого большого элемента **w[i]** и проверяя, можно ли его вычесть:

Если s ≥ w[i], то message[i] = 1, иначе message[i] = 0.

Вычитаем w[i] из s, если message[i] = 1.

Таким образом, исходное сообщение полностью восстанавливается!

1. **Заключение.**

* Ключи:

Закрытый ключ — w, n, e

Открытый ключ — β

* Шифрование:

Берём сообщение message

Вычисляем c = sum(message[i] \* beta[i])

* Дешифрование:

Вычисляем s = (c \* d) % n

Восстанавливаем message, начиная с самых больших w[i]

Таким образом, безопасность основана на том, что без w восстановить message из c очень сложно!

**[Разбор всех пунктов работы]**

**Пункт 2.1 Определение параметра K**

Дано:

* Номер учебной группы Nгр = 513
* Порядковый номер в списке группы Nсп=6
* Порядковый номер в алфавите третьей буквы фамилии Ф3=5

Формула:

K = ((Nгр + Nсп)^11 + Ф3) mod 11

Подстановка значений:

**K** = ((513 + 6)^11 + 5) mod 11 = **7**

*Данный параметр будет использоваться в следующих вычислениях.*

**Пункт 2.2 Определение параметра K**

**Исходная строка**: ВеденеевАртёмСергеевич

**K** = 7 – сдвиг из первого пункта.

**Алфавит**: русский, 33 буквы.

Формула шифрования: C = (P + K) mod N, где:

• C – шифросимвол,

• P – позиция исходной буквы в алфавите (0–32),

• K = 7 – сдвиг,

• N = 33 – размер алфавита.

Пример расчета для первой буквы:

• “В” – 2-я буквы алфавита (позиция 1, если считать от 0).

• C = (1 + 7) mod 33 = 8.

• 8-я буква алфавита – ‘й’

Результат:

**Зашифрованная строка**: ЙмлмфммйЗчщёуШмчкммйпю

**Пункт 2.3 Определение параметра K**

* 1. Высчитываем A:

A = (Nгр \* (8 + Nсп (mod 7))) ^ 2

**A** = (513 \* (8 + (6 mod 7))) ^ 2 = **51581124**

* 1. Поиск НОД (A, B):

B = 3042006 (03.04.2006)

Всего необходимо посчитать 4 НОД:  
1. НОД (A, (B % 95) + 900)

2. НОД (A, ((B + 50) % 97) + 700)

3. НОД (A, ((B + 20) % 101) + 1500)

4. НОД (A, ((B - 40) % 103) + 2500)

**Последовательность вычислений для НОД1**:

НОД (A, (B % 95) + 900)

51581124 ÷ 911 = 56620 (304)

911 ÷ 304 = 2 (303)

304 ÷ 303 = 1 (1)

303 ÷ 1 = 303 (0)

НОД (51581124, (3042006 % 95) + 900) = 1

**Последовательность вычислений для НОД2**:

Последовательность вычислений для НОД2:

НОД (A, ((B + 50) % 97) + 700)

((B + 50) % 97) + 700 = 739 => GCD (51581124, 739)

51581124 ÷ 739 = 69798 (402)

739 ÷ 402 = 1 (337)

402 ÷ 337 = 1 (65)

337 ÷ 65 = 5 (12)

65 ÷ 12 = 5 (5)

12 ÷ 5 = 2 (2)

5 ÷ 2 = 2 (1)

2 ÷ 1 = 2 (0)

НОД (51581124, 739) = 1