Мацуев Илья 3 курс 4 группа ФИТ

1. Используя некоторый вектор S = {103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610}, вычислить ключи для зашифрования и расшифрования сообщений.

Ei = Di \* a (mod N)

S = 35484

1. 35484 – 27610 = 7874 (1)
2. 7874 – 13807 = 0 (0)
3. 7874 – 6907 = 967 (1)
4. 967 – 3449 = 0 (0)
5. 967 – 1716 = 0 (0)
6. 967 – 864 = 103 (1)
7. 103 – 430 = 0 (0)
8. 103 – 211 = 0 (0)
9. 103 – 107 = 0 (0)
10. 103 – 103 = 0 (1)

1010010001

N = 55207 (большое суммы веса ранца), A = 77 (взаимнопростое с N)

103 \* 77 mod 55207 = 7931

107 \* 77 mod 55207 = 8239

211 \* 77 mod 55207 = 16247

430 \* 77 mod 55207 = 33110

863 \* 77 mod 55207 = 11244

1716 \* 77 mod 55207 = 21718

3449 \* 77 mod 55207 = 44745

6907 \* 77 mod 55207 = 34976

13807 \* 77 mod 55207 = 14206

27610 \* 77 mod 55207 = 28104

O = {7931, 8239, 16247, 33110, 11244, 21718, 44745, 34976, 14206, 28104};

A – **192 = 11000000 -** 7931 + 8239 + 14206 + 28104 = 58480

B – 193 = **11000001 – 21718 + 44745 + 34976 = 101439**

**C – 194 = 11000010 - 16247**

**M = 1100000011 0000011100 0010000000**

C` = 58480 101439 16247

РАСШИФРОВАНИЕ

A = 77

N = 55207

A-1 = A (mod N)

(A \* A-1) mod N = 1

(77 \* A-1) mod 55207 = 1

27962

77x + 55207y = 1

55207 = 77 \* 716 + 75

77 = 75 \* 1 + 2

75 = 2 \* 37 + 1

2 = 1 \* 2 + 0

1 = 75 – 2 \* 37 = (55207 – 77\* 716) – 2 \* 37 =

= (55207 – 77 \* 716) – (77 – 75) \* 37 =

= (55207 – 77 \* 716) – (77 – (55207 – 77\* 716)) \* 37 =

= (55207 – 77 \* 716) – (– 55207 + 77\* 717) \* 37 =

= (55207 – 77 \* 716) + 55207 \* 37 – 77\* 717 \* 37 =

= 55207 – 77 \* 716 + 55207 \* 37 – 77\* 26529 =

= - 77 \* 27245 + 55207 \* 38

55207 – 27245 = 27962

A = 77

A-1 = 27962

N = 55207

S = {103, 107, 211, 430, 863, 1716, 3449, 6907, 13807, 27610}

**M = 1100000011 0000011100 0010000000**

C` = 58480 101439 16247

S1 = C1 \* A-1 (mod N) = 58480 \* 27962 mod 55207 = 41627

S2 = C2 \* A-1 (mod N) = 101439 \* 27962 mod 55207 = 12072

S3 = C3 \* A-1 (mod N) = 16247 \* 27962 mod 55207 = 211

41627 – 27610 = 14017 (1)

14017 – 13807 = 210 (1)

1. \* 6

210 – 107 = 103 (1)

103 – 103 = 0 (1)

1100000011

1. \* 2

12072 – 6907 = 5165 (1)

5165 – 3449 = 1716 (1)

1716 – 1716 = 0 (1)

1. \* 5

0000011100

0010000000

**ДЕЛАЙ РЕВЕРС БЛЯТЬ**

Считаем сумму элементов вектора: 55203

55203 – простое число

55207 – следующее простое число после него

Выбираем второе взаимно простое первому число в диапазоне от 1 до 55207: 25013

a = 25013

n = 55203

103 \* 25013 mod 55203 = 37001

107 \* 25013 mod 55203 = 26647

211 \* 25013 mod 55203 = 33458

430 \* 25013 mod 55203 = 46208

863 \* 25013 mod 55203 = 1846

1716 \* 25013 mod 55203 = 29577

3449 \* 25013 mod 55203 = 42751

6907 \* 25013 mod 55203 = 34604

13807 \* 25013 mod 55203 = 4523

27610 \* 25013 mod 55203 = 19400

Получаем открытый ключ:

{37001, 26647, 33458, 46208, 1846, 29577, 42751, 34604, 4523, 19400}

Для зашифрования будем использовать открытый ключ. Для расшифровки Q и R, как обратное по модулю Q.

2. Найти первообразные корни (если они существуют) чисел (p): 19, 23, 27, 31, 37.

1) Вычисляем функцию Эйлера для p = 19: f(19) = 18

Количество первообразных корней: f(f(19)) = 6

Наименьшее число взаимно простое 19: 2

2 ^ 1 mod 19 = 2

2 ^ 2 mod 19 = 4

2 ^ 3 mod 19 = 8

2 ^ 4 mod 19 = 16

2 ^ 5 mod 19 = 13

2 ^ 6 mod 19 = 7

2 ^ 7 mod 19 = 14

2 ^ 8 mod 19 = 9

2 ^ 9 mod 19 = 18

2 ^ 10 mod 19 = 17

2 ^ 11 mod 19 = 15

2 ^ 12 mod 19 = 11

2 ^ 13 mod 19 = 3

2 ^ 14 mod 19 = 6

2 ^ 15 mod 19 = 12

2 ^ 16 mod 19 = 5

2 ^ 17 mod 19 = 10

2 ^ 18 mod 19 = 1

Таким же образом ищем первообразные корни на промежутке от 2 до 18

Первообразные корни: 2, 3, 10, 13, 14, 15

2) Вычисляем функцию Эйлера для p = 23: f(23) = 22

Количество первообразных корней: f(f(22)) = 10

Наименьшее число взаимно простое 23: 2

2 ^ 1 mod 23 = 2

2 ^ 2 mod 23 = 4

2 ^ 3 mod 23 = 8

2 ^ 4 mod 23 = 16

2 ^ 5 mod 23 = 9

2 ^ 6 mod 23 = 18

2 ^ 7 mod 23 = 13

2 ^ 8 mod 23 = 3

2 ^ 9 mod 23 = 6

2 ^ 10 mod 23 = 12

2 ^ 11 mod 23 = 1

2 ^ 12 mod 23 = 2

2 ^ 13 mod 23 = 4

2 ^ 14 mod 23 = 8

2 ^ 15 mod 23 = 16

2 ^ 16 mod 23 = 9

2 ^ 17 mod 23 = 18

2 ^ 18 mod 23 = 13

2 ^ 19 mod 23 = 3

2 ^ 20 mod 23 = 6

2^ 21 mod 23 = 12

2^ 22 mod 23 = 1

Таким же образом ищем первообразные корни на промежутке от 2 до 22

Первообразные корни: 5, 7, 10, 11, 14, 15, 17, 19, 20, 21

3) Вычисляем функцию Эйлера для p = 31: f(31) = 30

Количество первообразных корней: f(f(30)) = 8

Алгоритмом, аналогичным предыдущим примерам вычисляем:

Первообразные корни: 3, 11, 12, 13, 17, 21, 22, 24

4) Вычисляем функцию Эйлера для p = 37: f(37) = 36

Количество первообразных корней: f(f(36)) = 12

Алгоритмом, аналогичным предыдущим примерам вычисляем:

Первообразные корни: 2, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35

3. Пользователю системы RSA с его собственными ключевыми параметрами n = 3337, е = 79 передано зашифрованное сообщение С, состоящее из трех блоков: чисел, соответствующих порядковым номерам Ваших фамилии, имени и отчества. Расшифровать это сообщение (взломав систему RSA пользователя).

Мацуев Илья Михайлович

Запишем сообщение из 3-х блоков: 14, 10, 14.

Найдем число d по формуле: d = e^-1 (mod n) = 1019

Расшифруем блоки по формуле: mi = (ci) ^ d mod n, где m – исходное сообщение, а c – шифротекст.

14 ^ 1019 mod 3337 = 776

10 ^ 1019 mod 3337 = 1267

14 ^ 1019 mod 3337 = 776

Исходное сообщение: 776, 1267, 776.

4. Предположим, что в системе применяется алгоритм шифрования/расшифрования Эль-Гамаля. Известны некоторые параметры системы: р = 167, g = 5, y = g ^ 29 = 55 mod p.

Используя указанные и недостающие (выбрать самостоятельно) параметры, зашифровать свое имя (в любом языке) в предположении:

a) первая буква алфавита соответствует числу 0 и т. д.;

b) первая буква алфавита соответствует числу 1.

Проанализировать результат.

Открытый ключ: (167, 5, 55)

Закрытый ключ: х = 29

Илья:

a) 9, 12, 29, 32

b) 10, 13, 30, 33

Выбираем случайное число, меньшее 167 – 1 = 166, 3

A = 5 ^ 3 mod 167 = 125

1)

B = 55 ^ 3 \* 9 mod 167 = 53

B = 55 ^ 3 \* 12 mod 167 = 15

B = 55 ^ 3 \* 29 mod 167 = 78

B = 55 ^ 3 \* 32 mod 167 = 40

Получаем зашифрованный текст: (125, 53), (125, 15), (125, 78), (125, 40).

2)

B = 55 ^ 3 \* 10 mod 167 = 96

B = 55 ^ 3 \* 13 mod 167 = 58

B = 55 ^ 3 \* 30 mod 167 = 121

B = 55 ^ 3 \* 33 mod 167 = 83

Получаем зашифрованный текст: (125, 96), (125, 58), (125, 121), (125, 83).

Суть в том, что зашифровывать текст необходимо начиная с первой буквы, т. к. нулевая буква всегда будет 0.

5. Сколько попыток нужно сделать, чтобы с вероятностью более 0,5 (0,7; 0,8; 0,9) обнаружить коллизию при длине хеша (l) 64 (128; 256; 512) бит?

Нужно решить задачу с тождеством: Р(Аn) = 0.5.

Или: Р(Аn) ≈ ехр (-n^2 /2m) = 0.5.

Это можно записать в виде равенства (n ^ 2 / 2m) = ln 50.

Откуда мы находим n = (2m \* ln 50) ^ 0.5

Длина хэша - 1, тогда: m = 2 ^ l.

Следовательно, получаем n = ((2 \* 2 ^ 64) \* ln 50) ^ 0.5 = 5,1 \* 10 ^ 9

Считаем количество попыток при разных длинах хэша:

l = 128 -> n = 2.2 \* 10 ^ 19

l = 256 -> n = 4 \* 10 ^ 38

l = 512 -> n = 1.4 \* 10 ^ 77

Для 0.7:

l = 64 -> n = 6.7 \* 10 ^ 9

l = 128 -> n = 2.86 \* 10 ^ 19

l = 256 -> n = 5.28 \* 10 ^ 38

l = 512 -> n = 1.79 \* 10 ^ 77

Для 0.8:

l = 64 -> n = 7.7 \* 10 ^ 9

l = 128 -> n = 3.31 \* 10 ^ 19

l = 256 -> n = 6.1 \* 10 ^ 38

l = 512 -> n = 2.08 \* 10 ^ 77

Для 0.9:

l = 64 -> n = 9.2 \* 10 ^ 9

l = 128 -> n = 3.96 \* 10 ^ 19

l = 256 -> n = 7.3 \* 10 ^ 19

l = 512 -> n = 2.48 \* 10 ^ 77

6. Рассчитать общую длину (L') хешируемого сообщения после предварительной стадии на основе алгоритма MD, если объем (L) исходного сообщения составлял: 0; 484; 512; 1000; 2000; 16000 бит. Какова в каждом случае будет длина хеша?

Сообщение начальной длиной L бит расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина L

0 – длина сообщения, 0 так как ничего не передаем

484 – при добавлении в конце 64 битов мы получаем сообщение размером 548, для того чтобы оно было кратно 512 увеличим его на 476 до 1024

512 – при добавлении 64 битов получаем 576, для кратности увеличиваем сообщение до 1024

1000 – при добавлении 64 битов получаем 1064, для кратности увеличиваем сообщение до 1536

2000 – при добавлении 64 битов получаем 2064, для кратности увеличиваем сообщение до 2560

16000 – при добавлении 64 битов получаем 16064, для кратности увеличиваем сообщение до 16384

7. Входное сообщение (прообраз) состоит из а) вашего имени, б) ваших фамилии\_имени\_отчества (алфавит – на свой выбор). Используя представление сообщения в кодах ASCII, представить в табличной форме (как выше в примере 5) содержание каждого 32- битного подблока расширенного входного сообщения.

**а) Илья**

**б) Мацуев\_Илья\_Михайлович**

**А)** в кодах ASCII : 200 235 252 255

Двоичные значения: 11001000 11101011 11111100 11111111

**Б)** в кодах ASCII : 204 224 246 243 229 226\_200 235 252 255\_204 232 245 224 233 235 238 226 232 247

Двоичные значения:

11001100 11100000 11110110 11110011 11100101 11100010

11001000 11101011 11111100 11111111

11001100 11101000 11110101 11100000 11101001 11101011 11101110 11100010 11101000 11110111

Значение для символа «\_» бралось из таблицы ниже (код символа 95)

А) В последнюю часть из 64 бит полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 40: 101000. В остальные разряды мы впишем «0». После этого полученный расширенный блок делим на 16-ти 32-разрядных подблоков:

Илья:

11001000111010111111110011111111

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000101000

Б) В последнюю часть из 64 бит полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 174: 10101110. В остальные разряды мы впишем 0. После этого полученный расширенный блок делим на 16-ти 32-разрядных подблоков:

Мацуев\_Илья\_Михайлович:

11001100111000001111011011110011

11100101111000101100100011101011

11111100111111111100110011101000

11110101111000001110100111101011

1110111011100010 1110100011110111

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000010101110



