*Индивидуальная работа №1*

## xc mod n = 32095059mod 4219 = 152

## (5059)10 = (1001111000011)2

|  |  |  |
| --- | --- | --- |
| ***i*** | ***ci*** | ***z = 1*** |
| 12 | 1 | z = z2mod n = 1\*3209mod 4219 = 3209 |
| 11 | 0 | 32092mod 4219 = 3321 |
| 10 | 0 | 33212mod 4219 = 575 |
| 9 | 1 | (5752)\*3209mod 4219 = 2600 |
| 8 | 1 | (26002)\*3209mod 4219 = 3481 |
| 7 | 1 | (34812)\*3209mod 4219 = 3875 |
| 6 | 1 | (38752)\*3209mod 4219 = 691 |
| 5 | 0 | 6912mod 4219 = 734 |
| 4 | 0 | 7342mod 4219 = 2943 |
| 3 | 0 | 29432mod 4219 = 3861 |
| 2 | 0 | 38612mod 4219 = 1594 |
| 1 | 1 | (15942)\*3209mod 4219 = 361 |
| 0 | 1 | (3612)\*3209mod 4219 = ***152*** |

*Индивидуальная работа №2*

## b-1 mod n = 3583 -1mod 5107 = 2396

b = 3583

n = 5107

q = r =

temp =

*temp > 0 => temp = temp mod n*

*temp < 0 => temp = n – (- temp mod n)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  | 5107 | 3583 | 1 | 1524 | 0 | 1 | 5106 |
|  | 3583 | 1524 | 2 | 535 | 1 | 5106 | 3 |
|  | 1524 | 535 | 2 | 454 | 5106 | 3 | 5100 |
|  | 535 | 454 | 1 | 81 | 3 | 5100 | 10 |
|  | 454 | 81 | 5 | 49 | 5100 | 10 | 5050 |
|  | 81 | 49 | 1 | 32 | 10 | 5050 | 67 |
|  | 49 | 32 | 1 | 17 | 5050 | 67 | 4983 |
|  | 32 | 17 | 1 | 15 | 67 | 4983 | 191 |
|  | 17 | 15 | 1 | 2 | 4983 | 191 | 4792 |
|  | 15 | 2 | 7 | 1 | 191 | 4792 | ***2396*** |
|  | 2 | ***1*** | 2 | ***0*** |  |  |  |

1. = 3583

= 5107

q = = = 1

r = = 5107 – 1\*3583 = 1524

t = 1 = 0

temp = = 0 – 1\*1= -1

temp = n – (- temp mod n) = 5107 – (-(-1) mod 5107) = 5107 – 1mod 5107 = 5106

1. = 1524

= 3583

q = = = 2

r = = 3583 – 2\*1524 = 535

t = temp = 5106 = t = 1

temp = = 1 – 2\*5106 = 1 – 10212 = -10211

temp = n – (- temp mod n) = 5107 – (-(-10211) mod 5107) = 5107 – 10211mod 5107 = = 5107 – 5104 = 3

1. = 535

= 1524

q = = = 2

r = = 1524 – 2\*535 = 454

t = temp = 3 = t = 5106

temp = = 5106 – 2\*3 = 5100

temp = temp mod n = 5100 mod 5107 = 5100

1. = 454

= 535

q = = = 1

r = = 535 – 1\*454 = 81

t = temp = 5100 = t = 3

temp = = 3 – 1\*5100 = -5097

temp = n – (- temp mod n) = 5107 – (-(-5097) mod 5107) = 5107 – 5097mod 5107 = 10

1. = 81

= 454

q = = = 5

r = = 454 – 5\*81 = 49

t = temp = 10 = t = 5100

temp = = 5100 – 5\*10 = 5050

temp = temp mod n = 5050mod 5107 = 5050

1. = 49

= 81

q = = = 1

r = = 81 – 1\*49 = 32

t = temp = 5050 = t = 10

temp = = 10 – 5050 = -5040

temp = n – (- temp mod n) = 5107 – (-(-5040) mod 5107) = 5107 – 5040mod 5107 = 67

1. = 32

= 49

q = = = 1

r = = 49 – 1\*32 = 17

t = temp = 67 = t = 5050

temp = = 5050 – 67 = 4983

temp = temp mod n = 4983mod 5107 = 4983

1. = 17

= 32

q = = = 1

r = = 32 – 1\*17 = 15

t = temp = 4983 = t = 67

temp = = 67 – 4983 = -4916

temp = n – (- temp mod n) = 5107 – (-(-4916) mod 5107) = 5107 – 4916mod 5107 = 191

1. = 15

= 17

q = = = 1

r = = 17 – 1\*15 = 2

t = temp = 191 = t = 4983

temp = = 4983 – 191 = 4792

temp = temp mod n = 4792 mod 5107 = 4792

1. = 2

= 15

q = = = 7

r = = 15 – 7\*2 = 1

t = temp = 4792 = t = 191

temp = = 191 – 7\*4792 = -33353

temp = n – (- temp mod n) = 5107 – (-(-33353) mod 5107) = 5107 – 33353mod 5107 =

= 5107 – 2711 = **2396**

*Индивидуальная работа №3*

1. p = 19 q = 31 e = 41
2. n = p\*q = 19\*31 = 589
3. φ(n) = (p - 1)(q - 1) = 18\*30 = 540
4. d = 41-1mod 540 = 461

*Шифрование:*

c = me mod n m = Michael

m1 = M = 77

m2 = i = 105

m3 = c = 99

m4 = h = 104

m5 = a = 97

m6 = e = 101

m7 = l = 108

c1 = m1e mod n = 7741mod 589 = 77

c2 = m2 e mod n = 10541mod 589 = 269

c3 = m3 e mod n = 9941mod 589 = 150

c4 = m4 e mod n = 10441mod 589 = 396

c5 = m5 e mod n = 9741mod 589 = 469

c6 = m6 e mod n = 10141mod 589 = 442

c7 = m7 e mod n = 10841mod 589 = 356

*Дешифрование:*

m = cd mod n d = 461

m1 = c1d mod n = 77461mod 589 = 77

m2 = c2 d mod n = 269461mod 589 = 105

m3 = c3 d mod n = 150461mod 589 = 99

m4 = c4 d mod n = 396461mod 589 = 104

m5 = c5 d mod n = 469461mod 589 = 97

m6 = c6 d mod n = 442461mod 589 = 101

m7 = c7 d mod n = 356461mod 589 = 108

*Индивидуальная работа №4*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ASCII** | **A** | **B** |
| M | 77 | 1 + 77 = 78 | 0 + 78 = 78 |
| i | 105 | 78 + 105 = 183 | 78 + 183 = 261 |
| c | 99 | 183 + 99 = 282 | 261 + 282 = 543 |
| h | 104 | 282 + 104 = 386 | 543 + 386 = 929 |
| a | 97 | 386 + 97 = 483 | 929 + 483 = 1412 |
| e | 101 | 483 + 101 = 584 | 1412 + 584 = 1996 |
| l | 108 | 584 + 108 = **692** | 1996 + 692 = **2688** |

A = 692 mod 65521 = 692 (692)10 = (2B4)16

B = 2688 mod 65521 = 2688 (2688) 10 = (A80)16

Hash(Alder-32) = **A8002B4**

*Индивидуальная работа №5*

zp= 1319 g = 2 a = 1049 b = 1069

A = gamod p = 21049mod 1319 = 567

B = gbmod p = 21069mod 1319 = 704

KA= Bamod p = 7041049mod 1319 = 250

KB= Abmod p = 5671069mod 1319 = 250 => KA= KB= 250

*Индивидуальная работа №6 (Поточные криптосистемы)*

1. Создаем проток битов длиной 2n n = 3, 23 = 8 (количество перемешиваний)

Пусть вектор ключа S – 01234567, а вектор ключа – 18062973

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вектор ключа | 1 | 8 | 0 | 6 | 2 | 9 | 7 | 3 |
| ASCII | 49 | 56 | 48 | 54 | 50 | 57 | 55 | 51 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| K | 49 | 56 | 48 | 54 | 50 | 57 | 55 | 51 |

1. i = 0 j = 0

j = (j + k[i] + s[i]) mod 8 = (0 + k [0] + s [0]) = (0 + 49 + 0) mod 8 = 50 mod 8 = 2

Swap (s[i], s[j]) = swap (s [0], s [2])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | ***2*** | 1 | ***0*** | 3 | 4 | 5 | 6 | 7 |
| K | ***48*** | 56 | ***49*** | 54 | 50 | 57 | 55 | 51 |

1. i =1 j = 2

j = (j + k[i] + s[i]) mod 8 = (2 + k [1] + s [1]) = (2 + 56 + 1) mod 8 = 59 mod 8 = 3

Swap (s[i], s[j]) = swap (s [1], s [3])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 2 | ***3*** | 0 | ***1*** | 4 | 5 | 6 | 7 |
| K | 48 | ***54*** | 49 | ***56*** | 50 | 57 | 55 | 51 |

1. i = 2 j = 3

j = (j + k[i] + s[i]) mod 8 = (3 + k [2] + s [2]) = (3 + 49 + 0) mod 8 = 52 mod 8 = 4

Swap (s[i], s[j]) = swap (s [2], s [4])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 2 | 3 | ***4*** | 1 | ***0*** | 5 | 6 | 7 |
| K | 48 | 54 | ***50*** | 56 | ***49*** | 57 | 55 | 51 |

1. i = 3 j = 4

j = (j + k[i] + s[i]) mod 8 = (4 + k [3] + s [3]) = (4 + 56 + 1) mod 8 = 61 mod 8 = 5

Swap (s[i], s[j]) = swap (s [3], s [5])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 2 | 3 | 4 | ***5*** | 0 | ***1*** | 6 | 7 |
| K | 48 | 54 | 50 | ***57*** | 49 | ***56*** | 55 | 51 |

1. i = 4 j = 5

j = (j + k[i] + s[i]) mod 8 = (5 + k [4] + s [4]) = (5 + 49 + 0) mod 8 = 54 mod 8 = 6

Swap (s[i], s[j]) = swap (s [4], s [6])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 2 | 3 | 4 | 5 | ***6*** | 1 | ***0*** | 7 |
| K | 48 | 54 | 50 | 57 | ***55*** | 56 | ***49*** | 51 |

1. i = 5 j = 6

j = (j + k[i] + s[i]) mod 8 = (6 + k [5] + s [5]) = (6 + 56 + 1) mod 8 = 63 mod 8 = 7

Swap (s[i], s[j]) = swap (s [5], s [7])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 2 | 3 | 4 | 5 | 6 | ***7*** | 0 | ***1*** |
| K | 48 | 54 | 50 | 57 | 55 | ***51*** | 49 | ***56*** |

1. i = 6 j = 7

j = (j + k[i] + s[i]) mod 8 = (7 + k [6] + s [6]) = (7 + 49 + 0) mod 8 = 56 mod 8 = 0

Swap (s[i], s[j]) = swap (s [6], s [0])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | ***0*** | 3 | 4 | 5 | 6 | 7 | ***2*** | 1 |
| K | ***49*** | 54 | 50 | 57 | 55 | 51 | ***48*** | 56 |

1. i = 7 j = 0

j = (j + k[i] + s[i]) mod 8 = (0 + k [7] + s [7]) = (0 + 56 + 1) mod 8 = 57 mod 8 = 1

Swap (s[i], s[j]) = swap (s [7], s [1])

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| S | 0 | ***1*** | 4 | 5 | 6 | 7 | 2 | ***3*** |
| K | 49 | ***56*** | 50 | 57 | 55 | 51 | 48 | ***54*** |

1. *Шифрование:*

m = Michael

i = 0 j = 0

i = (i + 1) mod 8 = 1mod 8 = 1

j = (j + s[i]) mod 8 = (0 + 1) mod 8 = 1mod 8 = 1

t = (s[i] + s[j]) mod 8 = (s[1] + s[1]) mod 8 = (1 + 1) mod 8 = 2

m1 = M = 77 (77)10 -> (**1001101**)2

m2 = i = 105 (105)10 -> (**1101001**)2

m3 = c = 99 (99)10 -> (**1100011**)2

m4 = h = 104 (104)10 -> (**1101000**)2

m5 = a = 97 (97)10 -> (**1100001**)2

m6 = e = 101 (101)10 -> (**1100101**)2

m7 = l = 108 (108)10 -> (**1101100**)2

s[t] = s[2] = 4

k[t] = k[2] = 50 (50)10 -> (**110010**)2

## m XOR k[t]

|  |  |  |
| --- | --- | --- |
| XOR | A | B |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |

c1 = **1001101 XOR 110010 = 1111111 (1111111)2 = (127)10 = ‘Del’**

c2 = **1101001 XOR 110010 = 1011011 (1011011)2 = (91)10 = ‘[’**

c3 = **1100011 XOR 110010 = 1010001 (1010001)2 = (81)10 = ‘Q’**

c4 = **1101000 XOR 110010 = 1011010 (1011010)2 = (90)10 = ‘Z’**

c5 = **1100001 XOR 110010 = 1010011 (1010011)2 = (83)10 = ‘S’**

c6 = **1100101 XOR 110010 = 1010111 (1010111)2 = (87)10 = ‘W’**

c7 = **1101100 XOR 110010 = 1011110 (1011110)2 = (94)10 = ‘^’**

1. *Дешифрование: m = ‘Del [ Q Z S W ^’*

i = 0, j = 0

i = (i + 1) mod 8 = 1

j = (j+ s[i]) mod 8 = (0 + 1) mod 8 = 1

t = (s[i] + s[j]) mod 8 = (1 + 1) mod 8 = 2

s[t] = s[2] = 4

k[t] = k[2] = 50 (50)10 -> (**110010**)2

## c XOR k[t]

m1 = **1111111 XOR 110010 = 1001101 (1001101)2 = (77)10 = ‘M’**

m2 = **1011011 XOR 110010 = 1101001 (1101001)2 = (105)10 = ‘i’**

m3 = **1010001 XOR 110010 = 1100011 (1100011)2 = (99)10 = ‘c’**

m4 = **1011010 XOR 110010 = 1101000 (1101000)2 = (104)10 = ‘h’**

m5 = **1010011 XOR 110010 = 1100001 (1100001)2 = (97)10 = ‘a’**

m6 = **1010111 XOR 110010 = 1100101 (1100101)2 = (101)10 = ‘e’**

m7 = **1011110 XOR 110010 = 1101100 (1101100)2 = (108)10 = ‘l’**

*Индивидуальная работа №7*

Пусть, имеется следующая супервозрастающая последовательность:

W = {3, 12, 23, 45, 91, 180, 397}

Вес рюкзака: = 751

=> q = 881

l < r < q => r = 269

W, q, r – числа, образующие секретные ключи

1. *Шифрование:*

βi = (Wi\*r) mod q – публичные ключи

β1 = (3\*269) mod 881 = 807

β2 = (12\*269) mod 881 = 585

β3 = (23\*269) mod 881 = 20

β4 = (45\*269) mod 881 = 652

β5 = (91\*269) mod 881 = 692

β6 = (180\*269) mod 881 = 846

β7 = (397\*269) mod 881 = 192

βi = {807, 585, 20, 652, 692, 846, 192}

m1 = D = 68 (68)10 -> (1000100)2

m2 = a = 97 (97)10 -> (1100001)2

m3 = r = 114 (114)10 -> (1110010)2

m4 = i = 105 (105)10 -> (1101001)2

m5 = a = 97 (97)10 -> (1100001)2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 807 | 585 | 20 | 652 | 692 | 846 | 192 |  |
|  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1499 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 807 | 585 | 20 | 652 | 692 | 846 | 192 |  |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1584 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 807 | 585 | 20 | 652 | 692 | 846 | 192 |  |
|  | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 2258 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 807 | 585 | 20 | 652 | 692 | 846 | 192 |  |
|  | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 2236 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 807 | 585 | 20 | 652 | 692 | 846 | 192 |  |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1584 |

1. *Дешифрование:*

Описание: 269^(-1) mod 881 = 750

1499\*750 mod 881= 94

1584\*750 mod 881= 412

2258\*750 mod 881= 218

2236\*750 mod 881= 457

1584\*750 mod 881= 412

1. 94 - 91= 3

3 – 3 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 3 | 12 | 23 | 45 | 91 | 180 | 397 |  |
|  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | (68)10 ‘D’ |

1. 412 - 397 = 15

15 – 12 = 3

3 – 3 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 3 | 12 | 23 | 45 | 91 | 180 | 397 |  |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | (97)10 ‘a’ |

1. 218 – 180 = 38

38 – 23 = 15

15 – 12 = 3

3 – 3 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 3 | 12 | 23 | 45 | 91 | 180 | 397 |  |
|  | 1 | 1 | 1 | 0 | 0 | 1 | 0 | (114)10 ‘r’ |

1. 457 – 397 = 60

60 – 45 = 15

15 – 12 = 3

3 – 3 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 3 | 12 | 23 | 45 | 91 | 180 | 397 |  |
|  | 1 | 1 | 0 | 1 | 0 | 0 | 1 | (105)10‘i’ |

1. 412 – 397 = 15

15 – 12 = 3

3 – 3 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 3 | 12 | 23 | 45 | 91 | 180 | 397 |  |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | (97)10 ‘a’ |

*Индивидуальная работа №8 (Операции в поле “Galois”)*

Пусть A = и B =

Сложение: A + B = A XOR B = XOR =

Умножение:

1) на 1: A \* 1 = A =

2) на 2:

1. Если число < 0x80, тогда сдвиг влево на 1:

A \* 2 = << 1 = =

1. Если число > 0x80, то выполняется операция XOR с 0x1b , затем сдвиг влево на 1:

B \* 2 = ( XOR ) << 1 = << 1 = =

3) на любое другое число:

1. 90 \* 5 = 90 \* (2+2+1) = 90 \* 2 + 90 \* 2 + 90 \* 1

90 \* 2 = << 1 = =

90 \* 1 = 90

90 \* 5 = 180 +180 + 90 = 450

1. 250 \* 7 = 250 \* (2+2+2+1) = 250 \* 2 + 250 \* 2 + 250 \* 2 + 250 \* 1

250 \* 2 = ( XOR ) << 1 = << 1 = =

250 \* 1 = 250

250 \* 7 = 500 + 500 + 500 + 250 = 1750

*Индивидуальная работа №9 (Разделение секрета “Shamir”)*

Пусть, секрет равен S = 71. Разделим его на 7 частей.

Возьмем три простых числа и сформируем полином: a1 = 19 a2 = 11 a3 = 5

Полученный полином: f(x) = 19x3 + 11x2 + 5x + 71

Формируем тени:

f(1) = 19\*13 + 11\*12 + 5\*1 + 71 = 106

f(2) = 19\*23 + 11\*22 + 5\*2 + 71 = 277

f(3) = 19\*33 + 11\*32 + 5\*3 + 71 = 698

f(4) = 19\*43 + 11\*42 + 5\*4 + 71 = 1483

f(5) = 19\*53 + 11\*52 + 5\*5 + 71 = 2746

f(6) = 19\*63 + 11\*62 + 5\*6 + 71 = 4601

f(7) = 19\*73 + 11\*72 + 5\*7 + 71 = 7162

Полученные тени: (1, 106)(2, 277)(3, 698)(4, 1483)(5, 2746)(6, 4601)(7, 7162)

Далее выберем четыре из вышеприведенных теней: (2, 277)(3, 698)(4, 1483)(5, 2746)

Формируем секрет S, используя интерполяционный полином Лагранжа: L(x) = \sum_{j=0}^n y_j l_j(x), где l_j(x)=\prod_{i=0, j\neq i}^{n} \frac{x-x_i}{x_j-x_i} = \frac{x-x_0}{x_j-x_0} \cdots \frac{x-x_{j-1}}{x_j-x_{j-1}} \frac{x-x_{j+1}}{x_j-x_{j+1}} \cdots \frac{x-x_{n}}{x_j-x_{n}}\,\! :

l0 = = = (x3 – 12x2 + 47x - 60)

l1 = = = (x3 – 11x2 + 38x - 40)

l2 = = = (x3 – 10x2 + 31x - 30)

l3 = = = (x3 – 9x2 + 26x - 24)

L(x) = \sum_{j=0}^n y_j l_j(x)

(x3 – 12x2 + 47x - 60) + (x3 – 11x2 + 38x - 40) (x3 – 10x2 + 31x - 30) +

+ (x3 – 9x2 + 26x - 24) = x3( ) +

+ x2( ) +

+ x( ) + ( ) =

= 19x3 + 11x2 + 5x + 71

*Индивидуальная работа №10 (Разделение секрета “Blakley”)*

Выберем семь простых чисел с условием, что: m1\* m2\*m3\*m4 < S < m5\* m6\*m7

m1 = 11 m2 = 13 m3 = 17 m4 = 19

m1\* m2\*m3\*m4 = 46189

m5 = 23 m6 = 29 m7 = 31

m5\* m6\*m7 = 20677

M = m1\* m2\*m3\*m4 \* m5\* m6\*m7 = 955 049 953

Пусть S = 25000

Формируем тени: Si ≡ S mod mi

S1 = 25000 mod 11 = 8

S2 = 25000 mod 13 = 1

S3 = 25000 mod 17 = 10

S4 = 25000 mod 19 = 15

S5 = 25000 mod 23 = 22

S6 = 25000 mod 29 = 2

S7 = 25000 mod 31 = 14

Каждый человек получает параметры: (Si, mi, M)

P1 = (8, 11, 955 049 953)

P2 = (1, 13, 955 049 953)

P3 = (10, 17, 955 049 953)

P4 = (15, 19, 955 049 953)

P5 = (22, 23, 955 049 953)

P6 = (2, 29, 955 049 953)

P7 = (14, 31, 955 049 953)

Восстановление ключа:

Mi = Ni = Mi-1mod (mi)

M1 = = 86 822 723 N1 = 86 822 723-1mod 11 = 5

M2 = = 73 465 381 N2 = 73 465 381-1mod 13 = 7

M3 = = 56 179 409 N3 = 56 179 409-1mod 17 = 9

M4 = = 50 265 787 N4 = 50 265 787-1mod 19 = 15

M5 = = 41 523 911 N5 = 41 523 911-1mod 23 = 7

M6 = = 32 932 757 N6 = 32 932 757-1mod 29 = 13

M7 = = 30 808 063 N7 = 30 808 063-1mod 31 = 29

li = Si \* Mi \* Ni

l1 = 8 \* 86 822 723 \* 5 = 3472908920

l2 = 1 \* 73 465 381 \* 7 = 514257667

l3 = 10 \* 56 179 409 \* 9 = 5056146810

l4 = 15 \* 50 265 787 \* 15 = 11309802075

l5 = 22 \* 41 523 911 \* 7 = 6394682294

l6 = 2 \* 32 932 757 \* 13 = 856251682

l7 = 14 \* 30 808 063 \* 29 = 12508073578

S = (l1 + l2 + l3 + l4 + l5 + l6 + l7) mod (m1\* m2\*m3\*m4 \* m5\* m6\*m7) =

= (3472908920 + 514257667 + 5056146810 + 11309802075 + 6394682294 + 856251682 +

+ 12508073578) mod 955 049 953 = 40112123026 mod 955 049 953 = 25000

***Схема цифровой подписи RSA***

##### **Генерация ключей**

Генерация ключей в схеме цифровой подписи RSА точно такая же, как и генерация ключей в криптографической системе RSА.

Выбираем два простых числа **p** и **q** и вычисляем **n = p \* q**.

p = 823 и q = 953 n = 784319

Вычисляем Описание: \varphi  (n) = (p - 1) (q - 1). Значение Описание: \varphi  (n)- 782544.

Затем выбираем **e** = 313, для общедоступного ключа и вычисляем **d** для частного ключа, такое, что Описание: e x d = 1 mod \varphi \  (n)= d = 160009.

В схеме цифровой подписи RSA. d **является частным;**

e **и** n **- общедоступными**.

**Подписание**. На основе сообщения создаем подпись, используя частный (секретный) ключ, S = Md mod n, и передаем сообщение и подпись.

**Проверка**. Получатель принимает M = 19070 ->

S = (19070160009) mod 784319 = 210625313 mod 784319

Далее применяет общедоступный ключ к подписи, чтобы создать копию сообщения

М' = Se mod n.

Описание: М' = 210625313 \ mod \ 784319 = 19070 \ mod \ 784319 \ ? \ М 
\\
 М' \equiv \ mod \ n

После чего сравнивает значение М' со значением М. Т.к. два значения совпадают, принимаем сообщение.

#### Схема цифровой подписи Эль-Гамаля

##### Генерация ключей

Выберем достаточно большое простое число **p**, чтобы в поле **Z p\*** проблема дискретного логарифма была достаточно трудной. Пусть **e1** - простой элемент в **Z p\*.** Выбираем свой секретный ключ **d**, чтобы он был меньше, чем (**p – 1**). p = 3119, e1 = 2, d = 127

Вычисляем **e2 = e1d =** e2 = 2127 mod 3119 = 1702.

Открытый ключ - кортеж **(e1, e2, p)** ; секретный ключ - **d**.

**Подписывающийся** может подписать дайджест сообщения, направленный к любому объекту.

1. Выбираем секретное случайное число **r = 307**.
2. Вычисляем первую подпись **S1 = er mod p = 2307mod 3119 = 2083**.
3. Вычисляем вторую подпись **S2 = (M - d \* S1) \* r-1 = (320 - 127 \* 2083) \* 307-1 = 2105 mod 3118**, где **r** - мультипликативная инверсия **r** по модулю (**p – 1**).
4. Передаем **М = 320, S1 = 2083** и **S2** получателю.

**Проверка**. Получатель принимает **М, S1** и **S2** и может проверить их следующим образом.

1. Проверяет 0 < **S1** < p.
2. Проверяет 0 < **S2** < p - 1.
3. Вычисляет **V1 = e1M mod p =** **2320mod 3119** **= 3006 mod 3119**
4. Вычисляет **V2 = e2S1 \* e2S2 mod p** **= 17022083 \* 20832105 = 3006 mod 3119**
5. Если **V1** является конгруэнтным **V2**, сообщение принято; иначе оно будет отклонено.

#### Схема цифровой подписи Шнорра

##### Генерация ключей

Перед подписанием сообщения нужно сгенерировать ключи и объявить общедоступные ключи.

1. Выбираем простое число **p** = 2267 (p = 22 \* q + 1), которое обычно равно по длине 1024 битам.
2. Выбираем другое простое число ***q***  **= 103**, которое имеет тот же размер, что и дайджест, созданный функцией криптографического хэширования (160 битов). Простое число **q** должно делиться на **(p - 1)**. Т.е., **(p - 1) = 0 mod q**.
3. Выбираем **e1, q** -тый корень которого был бы равен **1 mod p**. Чтобы сделать это, выбираем примитивный элемент в **Zp, e0** = 2 и вычисляем **e1 = e0(p-1)/q mod p** = 222 mod 2267 = 354 ((p - 1) / q = 22).
4. Выбираем целое число, **d =** 30, как свой секретный ключ.
5. Вычисляем **e2 = e1d mod p** = 35430 mod 2267 = 1206.
6. Общедоступный ключ - **(e1, e2, p, q)**, секретный ключ - **(d)**.

**Подписание**

1. Выбираем случайное число **r =** 11, r должен иметь значение между 1 и q.
2. Вычисляем первую подпись **S1 = h(M | e1r mod p) =** 200 . Сообщение присоединяется (конкатенируется) спереди к значению **e1r mod p =** 35411 = 630 mod 2267, затем применяется хэш-функция, чтобы создать дайджест, функция получается из последовательного соединения М. и e1r mod p.
3. Вычисляем вторую подпись **S2 = r + d \* S1 mod q =** 11 + 1026\* 200 mod 103 = 11 + 24 = 35, по модулю q.
4. Передаем М =1000, S1 = 200 и S2 = 35.

**Верификация (проверка) сообщения**. Получатель принимает М, S1 и S2.

1. Вычисляет **V = h (М | e1S2 e2-S1 mod p) = 35**.
2. Если S2 конгруэнтно V по модулю p, сообщение принято; иначе оно отклоняется.

***Алгоритм MD4***

Предполагается, что на вход подано сообщение, состоящее из b бит, хеш которого нам предстоит вычислить. Здесь b — произвольное неотрицательное целое число; оно может быть нулем, не обязано быть кратным восьми, и может быть сколь угодно большим.   
Запишем сообщение побитово, в виде:

Описание: http://kriptografea.narod.ru/1.png

Ниже приведены 5 шагов, используемые для вычисления хеша сообщения.

##### **Шаг 1. Добавление недостающих битов.**

Сообщение расширяется так, чтобы его длина в битах по модулю 512 равнялась 448. Таким образом, в результате расширения, сообщению недостает 64 бита до длины, кратной 512 битам. Расширение производится всегда, даже если сообщение изначально имеет нужную длину.

Расширение производится следующим образом: один бит, равный 1, добавляется к сообщению, а затем добавляются биты, равные 0, до тех пор, пока длина сообщения не станет равной 448 по модулю 512. В итоге, к сообщению добавляется как минимум 1 бит, и как максимум 512

##### **Шаг 2. Добавление длины сообщения.**

64-битное представление b (длины сообщения перед добавлением набивочных битов) добавляется к результату предыдущего шага. В маловероятном случае, когда b больше, чем 2 в степени 64, используются только 64 младших бита. Эти биты добавляются в виде двух 32-битных слов, и первым добавляется слово, содержащее младшие разряды.

На этом этапе (после добавления битов и длины сообщения) мы получаем сообщение длиной кратной 512 битам. Это эквивалентно тому, что это сообщение имеет длину, кратную 16-ти 32-битным словам. Пусть M[0...N-1] означает массив слов получившегося сообщения (здесь N кратно 16).

##### **Шаг 3. Инициализация MD-буфера.**

Для вычисления хеша сообщения используется буфер, состоящий из 4 слов (A,B,C,D)(32-битных регистров). Эти регистры инициализируются следующими шестнадцатеричными числами (младшие байты сначала):

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

##### **Шаг 4. Обработка сообщения блоками по 16 слов.**

Для начала определим три вспомогательные функции, каждая из которых получает на вход три 32-битных слова, и по ним вычисляет одно 32-битное слово.

Описание: http://kriptografea.narod.ru/7.png<="" center="">

Описание: http://kriptografea.narod.ru/2.png

Описание: http://kriptografea.narod.ru/8.png

На каждую битовую позицию F действует как условное выражение: если X, то Y; иначе Z. Функция F могла бы быть определена с использованием вместо V, поскольку XY и \_XZ не могут равняться 1 одновременно. G действует на каждую битовую позицию как функция максимального значения: если по крайней мере в двух словах из X,Y,Z соответствующие биты равны 1, то G выдаст 1 в этом бите, а иначе G выдаст бит, равный 0. Интересно отметить, что если биты X, Y и Z статистически независимы, то биты F(X,Y,Z) и G(X,Y,Z) будут также статистически независимы. Функция H реализует побитовый xor, она обладает таким же свойством, как F и G.

##### **Шаг 5. Формирование хеша.**

Результат (хеш-функция) получается как ABCD. То есть, мы выписываем 128 бит, начиная с младшего бита A, и заканчивая старшим битом D.

***Алгоритм MD5***

На вход алгоритма поступает входной поток данных, хеш которого необходимо найти. Длина сообщения может быть любой (в том числе нулевой). Запишем длину сообщения в L. Это число целое и не отрицательное. Кратность каким-либо числам не обязательна. После поступления даных идет процесс подготовки потока к вычислениям.

##### **Шаг 1. Выравнивание потока**

Входные данные выравниваются так, чтобы их размер был сравним с 448 по модулю 512 (L’ = 512 ? N + 448). Сначала дописывают единичный бит в конец потока, затем необходимое число нулевых бит (выравнивание происходит, даже если длина уже конгруэнтна — сравнима с 448).

##### **Шаг 2. Добавление длины сообщения.**

В оставшиеся 64 бита дописывают 64-битное представление длины данных до выравнивания. Если длина превосходит 2 64 ? 1, то дописывают только младшие биты. После этого длина потока станет кратной степеням двойки — 16, 32. Вычисления будут основываться на представлении этого потока данных в виде массива слов по 512 бит.

##### **Шаг 3. Инициализация MD-буфера.**

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения шестнадцатеричными числами:

А = 01 23 45 67;

В = 89 AB CD EF;

С = FE DC BA 98;

D = 76 54 32 10.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние ABCD называется инициализирующим вектором.   
Определим еще функции и константы, которые нам понадобятся для вычислений.

* Потребуются 4 функции для четырех раундов. Введем функции от трех параметров — слов, результатом также будет слово.

Описание: http://kriptografea.narod.ru/3.png

Описание: http://kriptografea.narod.ru/4.png

Описание: http://kriptografea.narod.ru/5.png

Описание: http://kriptografea.narod.ru/6.png

* Определим таблицу констант T[1..64] — 64-элементная таблица данных, построенная следующим образом: T[i] = int(4294967296 \* | sin(i) | ) и s — циклический сдвиг влево на s бит полученого 32-битного аргумента.   
  Выравненные данные разбиваются на блоки (слова) по 32 бита, и каждый блок проходит 4 раунда из 16 операторов. Все операторы однотипны и имеют вид: [abcd k s i], определяемый как a = b + ((a + Fun(b,c,d) + X[k] + T[i]) < < < s), где X — блок данных. X[k] = M [n \* 16 + k], где k — номер 32-битного слова из n-го 512-битного блока сообщения.

##### **Шаг 4. Вычисление в цикле**

Заносим в блок данных элемент n из массива. Сохраняются значения A, B, C и D, оставшиеся после операций над предыдущими блоками (или их начальные значения, если блок первый).

AA = A

BB = B

CC = C

DD = D

* Раунд 1

/\*[abcd k s i] a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). \*/

[ABCD 0 7 1][DABC 1 12 2][CDAB 2 17 3][BCDA 3 22 4]

[ABCD 4 7 5][DABC 5 12 6][CDAB 6 17 7][BCDA 7 22 8]

[ABCD 8 7 9][DABC 9 12 10][CDAB 10 17 11][BCDA 11 22 12]

[ABCD 12 7 13][DABC 13 12 14][CDAB 14 17 15][BCDA 15 22 16]

* Раунд 2

/\*[abcd k s i] a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). \*/

[ABCD 1 5 17][DABC 6 9 18][CDAB 11 14 19][BCDA 0 20 20]

[ABCD 5 5 21][DABC 10 9 22][CDAB 15 14 23][BCDA 4 20 24]

[ABCD 9 5 25][DABC 14 9 26][CDAB 3 14 27][BCDA 8 20 28]

[ABCD 13 5 29][DABC 2 9 30][CDAB 7 14 31][BCDA 12 20 32]

* Раунд 3

/\*[abcd k s i] a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). \*/

[ABCD 5 4 33][DABC 8 11 34][CDAB 11 16 35][BCDA 14 23 36]

[ABCD 1 4 37][DABC 4 11 38][CDAB 7 16 39][BCDA 10 23 40]

[ABCD 13 4 41][DABC 0 11 42][CDAB 3 16 43][BCDA 6 23 44]

[ABCD 9 4 45][DABC 12 11 46][CDAB 15 16 47][BCDA 2 23 48]

* Раунд 4

/\*[abcd k s i] a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). \*/

[ABCD 0 6 49][DABC 7 10 50][CDAB 14 15 51][BCDA 5 21 52]

[ABCD 12 6 53][DABC 3 10 54][CDAB 10 15 55][BCDA 1 21 56]

[ABCD 8 6 57][DABC 15 10 58][CDAB 6 15 59][BCDA 13 21 60]

[ABCD 4 6 61][DABC 11 10 62][CDAB 2 15 63][BCDA 9 21 64]

Суммируем с результатом предыдущего цикла:

A = AA + A

B = BB + B

C = CC + C

D = DD + D

После окончания цикла необходимо проверить, есть ли еще блоки для вычислений. Если да, то изменяем номер элемента массива (n++) и переходим в начало цикла.

##### **Шаг 5. Результат вычислений**

Результат вычислений находится в буфере ABCD, это и есть хеш. Если вывести слова в обратном порядке DCBA, то мы получим наш MD5 хеш.

***Алгоритм MD6***

В алгоритме хэш-функции использованы весьма оригинальные идеи. За один проход обрабатывается 512 байт, затрудняя проведение ряда атак и облегчая распараллеливание, для чего так же применяются древовидные структуры.

Помимо традиционного бесключевого режима, применяется хэширование с ключом 512-бит. Нелинейность функции основывается на использовании ряда простейших операций:

* XOR
* сложения и сдвига

Количество раундов составляет: r = 40 + (d / 4), что нетрадиционно велико.