X = 51; Y = X%4 == 3

note:

^: xor

(1)

y = f(x) = x\*z1 + z2 (%n) where n=30

there are 30 possibles for z2

z1 must be inversable in (Zmod(30)) => z1 must coprime 30

=> 8 possibles for z1

there are 30\*8 possible keys for this cryptography

(2)

1.

e = Y\*2+3 = 9

p,q in range(10,20) and gcd(e,(p-1)\*(q-1))==1

p,q = 17,11

phi = (p-1)\*(q-1) = 160

=> n = p\*q = 187

2.

d = pow(e,-1,phi)= 89

3.

N = X = 51 => N%16=3

m = 0b00010010 ^ 3 == 17

c = pow(m,e,n) = 68

4.

send => E(m) || D'(E(m)) || pub\_key'

D' for create\_signature

E' for verify signature

(3)

output\_bits = (Y+9)\*8 (bits) = 96(bits)

=> hash\_space = pow(2,output\_bits)

chip[i] can do: pow(10,i) \* 1000 (hash/sec)

price : pow(i,i/2) \* 1000 ($)

time hacking:

24 (hours) = 3600\*24 (sec)

with probability of success 75% (base on birthday paradox)

P = 75% =~= 1- pow(exp,-k\*(k-1)/(2\*hash\_space))

k: number\_hash\_gen

k = 468686579571925

hash\_per\_sec = k/(3600\*24) = 5424613189.489873

price = ceil(hash\_per\_sec/(pow(10,i) \* 1000)) \* pow(i,i/2) \* 1000 (min this function)

we can proof that => The more advanced chips, the higher efficiency (but not too advance)

buy 1 chip = 7 => 907492.7 $