

2014-08-01-125745

Dr.F.G. Garvan

8/2/2014

## Contents

```
# RSA EXAMPLE
# Suppose Alice wants to send Bob the message
# JON SKATES
# First Alice converts this is a number using spce=00, A=01,
# B=02, C=03, ..., Z=26
# M = 10 15 14 00 19 11 01 20 05 19
# Then Alice looks up Bob's public n and r:

n=11096351737
r=684315297

# Then she breaks up M into blocks less than n:

M1=1015140019
M2=1101200519

# Alice next calculates M1^2 and M2^r mod n

mod(M1^r,n)
Error in lines 1-1
Traceback (most recent call last):
  File "/projects/17ab5038-8f40-4654-a1b9-176371ed4c00/.sagemathcloud/sage_server.py",
line 736, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in <module>
  File "integer.pyx", line 1994, in sage.rings.integer.Integer.__pow__
(sage/rings/integer.c:14677)
  File "c_lib.pyx", line 73, in sage.ext.c_lib.sig_raise_exception (sage/ext/c_lib.c:872)
KeyboardInterrupt

# We see that using the mod function is not practical. We
# need to use the power_mod function (which naturally uses the power mod \
algorithm)

E1=power_mod(M1,r,n);E1
9974872058

E2=power_mod(M2,r,n);E2
```

```
7148290747
```

```
# Also then encodes the message as
# E = E1 . E2 = 99748720587148290747
# Bob decodes the message by calculating E1^s and E2^s mod n using his \
secret s.
#
# Instead we can find Bob's secret s by factoring n = p*q
# After finding p,q we find s      that r*s == 1 mod ( (p-1)*(q-1))
factor(n)
104729 * 105953

p=104729
q=105953

SL=xgcd(r,(p-1)*(q-1)); SL
(1, 657804897, -40567793)

s=mod(657804897,(p-1)*(q-1));s
mod(r*s,(p-1)*(q-1))
s=Integer(s)
657804897
1

# Hence we find Bob's secret s = 657804897

D1=power_mod(E1,s,n);D1
1015140019

D2=power_mod(E2,s,n);D2
1101200519

D1==M1
D2==M2
True
True

# Observe that M1 == E1^s mod n and M2 == E2^s mod n
# and we are able to find M1, M2, M and the original message.
```