The RSA Encryption Scheme

RSA is one of the first practical _____ cryptosystems and is widely used in _____ _____. In such a cryptosystem the encryption key is _____ and differs from the decryption key which is _____. In RSA, this asymmetry is based on the practical difficulty of _____. RSA is made of the initial letters of _____, _____, and _____, who first publicly described the algorithm in _____. _____, an _____ mathematician, had developed a equivalent system in _____, but it was not declassified until _____.

In an RSA scheme any number of people can send each other messages, which _____ be read by _____ other than the _____. Each individual in the scheme finds two _____, p and q. Each person then calculates

$$n = \_\_\_\_\_.$$

Then an integer r is chosen that is _____ _____. Finally, s is calculated so that

$$\_\_\_\_\_ \equiv \_\_\_\_\_$$

The integers ___ and ___ are made _____
(perhaps each person's ___ and __ are published in
a directory) ——— while keeping ___, ___,
and ___ secret.

Now suppose A wants to send a secret message to B
let us say the message

JON SKATES

First A must turn the message into a _____.
It doesn't matter how so long as everyone does it the same
way, let us simply let A = __, B = __, C = __, ...,
Z = __, space = ___. The message is now the
number

M = ...

Now __ looks up ___'s __ and __ in the
directory and calculates

E ≡ .....,

and sends ___ to __ as the encoded message.
__ decodes the message E by calculating

.....

using his secret ___. As we shall see

..... ≡ M .....

so that B recovers the message M.

The description here only works if M < __;
if M ≥ __ then M must be broken into _____
_____. The word ___
gets encoded as _____, and decoded
by _____.

Example:

Suppose A finds B's entry in the directory to be $n = 1109635737$ and $r = 68431529^7$. (This value of n is too small to occur in practice (as we shall see), but it will do to illustrate the scheme.) A breaks M into 10 digit pieces:

$M_1 =$ $M_2 =$

In general, if the last piece does not contain 10 digits some _____. Then using $r = 68431529^7$ A calculates

$E_1 \equiv$

$E_2 \equiv$

Then A transmits to B the message

$E =$

Now B breaks this up into 10 digit pieces

$E_1 =$ $E_2 =$

He then uses his secret ___ which is _____ to calculate

$M_1 \equiv$

$M_2 \equiv$

Thus B has recovered

$$M =$$

which easily translates to

Theorem (Proof of the Decoding Procedure)
Suppose p and q are distinct _____.

$$n =$$,

Suppose the positive integer r satisfies _____
and s is a positive integer such that

_____.

Let M be a positive integer. Suppose $M <$ __
and

$$E \equiv$$.

Then

$$\equiv M \pmod{\quad}.$$

PROOF: Suppose all the key hypotheses hold.
Since p and q are _____,

$$\phi(n) =$$.

Since $rs \equiv \pmod{\quad}$, there is
a _____ integer d such that

_____.

We consider 2 cases.
We must show that

_____.

Case 1. $(M, n) = 1$. Then

Case 2 $(M, n) > 1$. We may assume without loss of generality that ____.

Security of the RSA Scheme

If a third party C can _____, that is if he can find ___ and ___, and then he can compute _____, and hence compute the secret key s using the _____ since

$$\equiv$$

Thus the scheme is not secure unless _____ that C is _____.

A well chosen 2048 bit n is unlikely to be factored. This means that p and q should be about _____ bits; ie about ___ digits.

Is there a way to break the code without _____? Nobody has found one (or, if they have, they aren't telling).

Advantages over traditional coding methods

In the usual coding schemes A and B have to agree on a secret key before they can start sending messages. If the communication of the secret key is _____, then _____. In the RSA scheme, _____.

Suppose k people wish to participate in an information network in such a way that no one can decode messages sent between any two others. Then each pair needs a secret key; each person needs to keep track of ___ different secret keys.

Thus the total number of keys needed is _____.
In the RSA scheme each person has only
two keys, his _____ and _____;
in all, only ___ keys.

Finding large primes

How do we find a 300 digit prime?
One way is to _____ and
_____. If it proves composite
then _____. How many numbers can
you expect to test before finding a prime?
The Prime Number Theorem says that among the integers
near x roughly one out of every _____
is prime.
Since $\log(10^{300}) =$ _____ $\approx$ ___,
you would expect to test about ___ numbers
to find one that is prime.

Some algorithms

To implement the RSA algorithm we need
(1) a method for computing ___;
(2) an efficient method for computing
_____.

(1) Modular Inversion Algorithm

Input r and n

Let $s' = 0$ and $s = 1$

while $r > 0$ do

$t \leftarrow n$

$n \leftarrow r$

$q \leftarrow [t/n]$

$r \leftarrow t - rq$

$u \leftarrow s$

$s \leftarrow s' - sq$

$s' \leftarrow u$

Output s'

STOP

When applying the Euclidean Algorithm, we have

$$n = q_1 r + r_1$$
$$r = q_2 r_1 + r_2$$
$$r_1 = q_3 r_2 + r_3$$
$$\vdots$$
$$r_m = q_{m+1} r_{m+1} + r_{m+2}$$
$$r_{m+1} = q_{m+2} r_{m+2} + 0$$

where $r_{m+2} = 1$

at each stage we have

$$r_1 = 1 \cdot n - q_1 r$$
$$\vdots$$
$$r_j = (\cdot) n + s' r$$
$$r_{j+1} = (\cdot) n + s r$$
$$r_{j+2} = r_j - q_{j+2} \cdot r_{j+1}$$
$$= (\cdot) n + s' r - q_{j+2}\left((\cdot) n + s r\right)$$
$$= (\cdot) n + (s' - q_{j+2} s) r$$

Initially $(\cdot) = (\cdot)n + \_\_\_ r$

$r = (\cdot)n + \_\_\_ r$

$r_1 = (\cdot)n + \_\_\_\_ r$

& initially $s' =$ ___

$s =$ _____

We find that when the algorithm finishes

$=$

and

The Power Algorithm

We describe an efficient method for computing _____.
First we describe an efficient method for computing powers.
As an example suppose we wish to compute a^{82} for some given a. The naive way would be to do __ multiplications but this is clearly inefficient.
The idea is to convert 82 to binary

$82 =$

so that

$82 =$

We compute powers of a by doing repeated ____:

a^2,

This takes ___ multiplications

Then we complete the computation of a^{82} by doing ___ more multiplications:

$a^{82} =$

This gives a total of ___ multiplications instead of ___. This leads to the following algorithm:

The Power Algorithm

```
Input a and N
Let j = 1, Q = 1, Y = 1 and n = N.
    while n > 0 do
        if n is odd then
            b ⟵ 1
            if n = N then
                Q ⟵ a
                Y ⟵ a
            else
                Q ⟵ Q²
                Y ⟵ Q * Y
        else
            b ⟵ 0
            if n = N then
                Q ⟵ a
            else
                Q ⟵ Q²
        n ⟵ [n/2]
Output Y
STOP
```

The input of this algorithm is _____

The output is ____.

To compute $a^N \pmod m$ requires a simple modification of the Power Algorithm. We need a function that does reduction mod m. One needs to define a function whose input is _____ and whose output is the _____:

$\mathrm{mod}(a, m) =$ _____.

The Power Algorithm mod m

```
Input a, N and m
Let j = 1, Q = 1, Y = 1 and n = N.
    while n > 0 do
        if n is odd then
            b ⟵ 1
            if n = N then
                Q ⟵ a
                Y ⟵ a
            else
                Q ⟵ mod (Q^2, m)
                Y ⟵ mod (Q * Y, m)
        else
            b ⟵ 0
            if n = N then
                Q ⟵ a
            else
                Q ⟵ mod (Q^2, m)
        n ⟵ [n/2]
Output Y
STOP
```

The input of this algorithm is _____.

The output is _____.

(p.12)

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
```

(p.13)

```
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.
```

Extra Credit Assignment

(a) Using the usual alphabet encoding encode the phrase

SEND MONEY

as three 8 digit numbers:

$M_1 = 19051404$, $M_2 =$, $M_3 =$

(b) Now use the RSA encryption scheme to encode M_1, M_2, M_3 with $n = 536813567$, and $r = 3602561$ to obtain three numbers $(< n)$:

$E_1 = 463099189$, $E_2 =$, $E_3 =$

(c) Break the code by factoring n & obtaining the secret key s.
Explain clearly all your reasoning, work and detail of computations.
Check that E_1 decodes as M_1 etc showing details.