Network Security, IT653
Assignment 1
Karl-Johan Grahn, roll no 03005202

1 Part 1

The message m is encrypted as follows with encryption-key $e = 3$

$$\begin{cases} E(m) = m^e \pmod{n_1} = c_1 \\ E(m) = m^e \pmod{n_2} = c_2 \\ E(m) = m^e \pmod{n_3} = c_3 \end{cases}$$

Since n_1, n_2, n_3 are relatively prime to each other, i.e.

$$\begin{cases} \gcd\,(n_1, n_2) = 1 \\ \gcd\,(n_1, n_3) = 1 \\ \gcd\,(n_2, n_3) = 1 \end{cases}$$

if follows from the Chineese Remainder Theorem that the system

$$\begin{cases} m^e \equiv c_1 \pmod{n_1} \\ m^e \equiv c_2 \pmod{n_2} \\ m^e \equiv c_3 \pmod{n_3} \end{cases}$$

has a solution, which is unique modulo $n = n_1 \cdot n_2 \cdot n_3$. Let us prove this. First of all, define

$$N_i = \prod_{j \neq i} n_j = \frac{n}{n_i}.$$

From this definition we see that N_i is relatively prime to n_i, and therefore has a unique multiplicative inverse modulo n_i. From $\gcd(N_i, n_i) = 1$ it follows that there exists numbers b_i such that

$$b_i N_i \equiv 1 \pmod{n_i}, \quad 1 \leq i \leq 3.$$

Let

$$m^e = c_1 b_1 N_1 + c_2 b_2 N_2 + c_3 b_3 N_3.$$

From

$$b_j N_j \equiv \begin{cases} 1 \pmod{n_j} \\ 0 \pmod{n_i}, \quad i \neq j, \end{cases}$$

it follows that

$$m^e = c_1 b_1 N_1 + c_2 b_2 N_2 + c_3 b_3 N_3 \equiv c_j \pmod{n_j}, \quad 1 \leq j \leq 3,$$

so $m^e = c_1 b_1 N_1 + c_2 b_2 N_2 + c_3 b_3 N_3$ is a solution to the system.

The difficult part of the calculations is to calculate the numbers b_i, the multiplicative inverses of N_i modulo n_i. This can efficiently be done with Euclide's algorithm. Left is just to raise m^e and its right hand side with $\frac{1}{e}$.

2 Part 2

Generate an RSA Public-Private key pair as function of key size

Key size (bits)	128	256	512	768	1024
Generation time (ms)	301	510	3095	3856	8081
	250	371	1092	11857	4236
	290	481	2143	3906	7050
	250	451	1152	9994	15702
	311	440	961	4887	20980
Average time (ms)	280	451	1687	6900	11210

Up to 256 bits, the running time seems to be linear, and after 256 bits the running time seems to be polynomial. For example, to double the bits from 128 to 256 causes a double in generation time. To double the bits from 512 to 1024 however, causes more than a double in generation time. The latter could therefore be slightly polynomial. But as test data shows, the generation time varies a lot for higher values of key size, so it is difficult to make precise statements about the running time as the key size grows.

Encrypt and decrypt a message of size 1K as function of public key sizes

Key size (bits)	128	256	512	768	1024
Encryption time (ms)	5047	34480	217353	648313	1459008
Decryption time (ms)	5137	35301	201590	627913	1514218

Encryption- and decryption time as a function of public key sizes seems to be polynomial, since a double in key size causes a polynomial increase in encryption and decryption time.

Encryption and decryption of messages as function of text sizes

Key size (bits)	128				
Message size	0K	1K	2K	3K	4K
Encryption time (ms)	80	4867	7481	10715	14040
Decryption time (ms)	70	4386	7050	10004	13169

Key size (bits)	256				
Message size	0K	1K	2K	3K	4K
Encryption time (ms)	511	30183	43503	66195	91481
Decryption time (ms)	501	33398	43192	70191	93855

Encryption and decryption time as a function of message sizes seems to be linear, since an increase in text size causes an increase in encryption and decryption time with a constant.

Key generation time for different values as function of the probability value

The BigInteger class has a parameter which measures the probability that the new BigInteger represents a prime number. This probability will exceed $(1 - \frac{1}{2}^{(\text{probability value})})$. The execution time of the BigInteger constructor is proportional to the value of this parameter. The table below shows the generation time of key for different values of the probability variable. The key size used is 128 bits.

Probability value	10	20	100
$1 - \frac{1}{2}^{(\text{probability value})}$	0.9990234	0.9999990	0.9999999
Generation time (ms)	260	300	340
	360	430	350
	260	271	270
	481	190	280
	241	270	280
Average generation time (ms)	320	292	304

After inspecting the table, it is evident that the generation time of the key is almost independent of the probability value, within the margins of systematic error.

All claims about running time would be more accurate if I had collected more data. Finding the running time for an algorithm is indeed an intricate problem. A lot of factors need to be taken in concern before we make precise statements.

I tried the most to perform the tests under equal conditions. The tests were made on a Pentium Celeron 800 MHz with 256 MB Ram.

3 References

- Björner, Anders: Kinesiska restsatsen (Chineese remainder theorem) http://www.math.kth.se/∼bek/diskret/kinrest.ps
- Björner, Anders: Kryptografi och primalitet (Cryptography and primality) http://www.math.kth.se/∼bek/diskret/krypto.ps
- Stallings, W: Cryptography and Network Security, 3rd ed.

Encryption text of size 1K:

This article examines the military organization and expertise of Cade's Rebellion in 1450. Contrary to the views of some modern historians that late-medieval rural populations lacked military skills or arms, it offers evidence

that the rebels possessed an instrument of military organization through the traditional structure of the posse comitatus or militia under the command of locally-elected constables, was led by men with considerable and current skill in warfare, and was at least as well-armed as most other military forces in the era. It argues that the rebellion should be understood not as the incoherent 'jacquerie' of Shakespeare's (and latter historians') version, but as an attempt by the politically responsible element of Kent's common population to assert a right of participation in the national political dialogue through armed protest. It concludes by suggesting that for late-medieval English society the right to political expression was still largely defined in 'chivalric' terms, as the monopoly of the literally armigerous or arms-bearing elite, and that for a common population, which had its own arms and the organizational wherewithall to use them effectively, rebellion was a way to interrupt this monopoly in order to interject its voice in the national political dialogue.