Jeremy Barthélemy  
ECE 646  
Hw. 5

Problem 12  
The fact that Alice is encrypting something that can only take 26 different values and sending them one at a time does not seem very secure. Anybody could have Bob’s public key and generate an encrypted form of a “0,” for example, and compare this with what Alice sent to Bob (the ciphertext for 0, encrypted with Bob’s public key). Thus, somebody could easily make a table of and determine the mapping between the ciphertext and the plaintext for each of the 26 integers. This method is not secure.

Problem 13

*Using a spreadsheet (such as Excel), or a calculator, perform the described below operations, step by step.  
Document results of all intermediate modular multiplications.   
Determine and report the number of modular multiplications per each encryption and decryption.*

A. Encrypt the message M=10 using RSA with the following values of parameters e and N:   e=17 and N=17,869 using two different algorithms  
  
a. right-to-left binary exponentiation  
  
C = Me mod 17869  
C = 1017 mod 17869  
E = 17 = (10001)2  
Where e0= 1, e1- e3= 0, and e4 = 1.  
   
Calculations of S:  
10  
102 mod 17869 = 100 mod 17869 = 100  
1002 mod 17869 = 10000 mod 17869 = 10000  
100002 mod 17869 = 100000000 mod 17869 = 5076  
50762 mod 17869 = 25765776 mod 17869 = 16547  
  
(10 \* 1 \* 1 \* 1 \* 16547) mod 17869  
165470 mod 17869 = 4649  
  
C= 4649  
Number of modular multiplications: 5 + 2 = 7  
  
  
  
  
  
  
  
  
  
  
b. left-to-right binary exponentiation

Again, E = 17 = (10001)2 and e0= 1, e1- e3= 0, and e4 = 1.  
(((102 mod 17869) )2 mod 17869)2 mod 17869)2 mod 17869 \* 10 mod 17869  
(((100 mod 17869) )2 mod 17869)2 mod 17869)2 mod 17869 \* 10 mod 17869  
((10000 mod 17869)2 mod 17869)2 mod 17869 \* 10 mod 17869  
((10000)2 mod 17869)2 mod 17869 \* 10 mod 17869  
(100000000 mod 17869)2 mod 17869 \* 10 mod 17869  
(5076)2 mod 17869 \* 10 mod 17869  
25765776 mod 17869 \* 10 mod 17869  
16547 \* 10 mod 17869  
165470 mod 17869  
4649  
C = 4649  
  
Number of modular multiplications: 5 + 2 = 7  
  
  
B. Compute a private key {d, P, Q} corresponding to the given above public key {e, N}.   
e = 17  
N = 17869  
P = 107, Q = 167  
d\*e = 1 mod φ(n)  
17\*d = 1 (mod 106\*166)  
d = 17-1 \* 1 (mod 17596)  
gcd(17, 17596) = 1, so only one solution

|  |  |  |  |
| --- | --- | --- | --- |
| I | qi | ri | xi |
| -2 | - | 17596 | 0 |
| -1 | 1035 | 17 | 1 |
| 0 | 17 | 1 | -1035 |
| 1 | - | 0 |  |

16561 mod 17596  
d = 16561  
  
{d, P, Q} = {16561, 107, 167}  
  
  
C. Perform the decryption of the obtained ciphertext using two different methods:  
C = 4649  
{d, P, Q} = {16561, 107, 167}  
  
     a. without using Chinese Remainder Theorem,  
M = Cd mod n  
464916561 (mod 17869)  
16561 = (100000010110001)2  
Here I am using right-to-left exponentiation.  
We invert the binary form of the exponent, and apply the formula of s2 mod N where s is the value of the previous column in the second row. For the first column, the value is simply C1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|  | 4649 | 9580 | 1216 | 13398 | 12299 | 4316 | 8358 | 6243 | 2760 | 5406 | 9021 | 3015 | 12773 | 5559 | 6980 |
| Multiply: | 4649 | 1 | 1 | 1 | 12299 | 4316 | 1 | 6243 | 1 | 1 | 1 | 1 | 1 | 1 | 6980 |
| M = | 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

b. using Chinese Remainder Theorem.  
C = 4649  
{d, P, Q} = {16561, 107, 167}  
M = MP \* RQ + MQ \* RP mod N  
where  
RP = (P-1 mod Q) \* P = PQ-1 mod N  
RQ = (Q-1 mod P) \* Q= QP-1 mod N  
  
First,  
RP = (107-1 mod 167) \* 107 = 107166 mod 17869  
RQ = (167-1 mod 107) \* 167 = 167106 mod 17869  
We solve for RP and RQ first so as to decrypt the message M.  
Here, I am using right-to-left exponentiation as in the previous case.  
We convert 166 to binary: 10100110  
and 106 as well: 1101010

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RTL\_BIN | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
|  | 107 | 11449 | 10486 | 8239 | 14659 | 11556 | 6099 | 12412 |
| Mult: | 1 | 11449 | 10486 | 1 | 1 | 11556 | 1 | 12412 |
| RP = | 6848 |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RTL\_BIN | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
|  | 167 | 10020 | 12358 | 11690 | 11857 | 13026 | 10521 |
| Mult: | 1 | 10020 | 1 | 11690 | 1 | 13026 | 10521 |
| RQ = | 11022 |  |  |  |  |  |  |

CP = C mod P = 4649 mod 107 = 48  
CQ = C mod Q = 4649 mod 167 = 140  
dp = d mod (P-1) = 16561 mod 106 = 25  
dQ = d mod (Q-1) = 16561 mod 166 = 127  
  
M = MP \* RQ + MQ \* RP mod N  
 = MP \* 11022 + MQ \* 6848 mod N  
  
  
MP = CPdp mod P = 4825 mod 107 = (488 \* 488 \* 488 \* 48) mod 107  
 = (23 \* 23 \* 23 \* 48) mod 107 = 10  
MQ = CQdq mod Q = 140127 mod 167

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 140 | 61 | 47 | 38 | 108 | 141 | 8 |
| MQ= | 10 |  |  |  |  |  |  |

M = MP \* RQ + MQ \* RP mod N  
 = 10 \* 11022 + 10 \* 6848 mod 17869 = 10

**Problem 14 (**

**Derive formulas for the execution time of the given below operations in software and in hardware and rank them from smallest to largest.**

A. RSA signature generation using CRT with the 2048-bit d, 1024-bit P, and 1024-bit Q.  
SOFTWARE:  
tEXP(e, L, k) = #modular\_multiplications(e, L) \* tMULMOD(k)  
tMULMOD(k) = csm\*k22 \* tEXP(d = 2048, L = 512, k = 1024) = 2 \* (512)\*csm\*(2 **\*** 512)2 = 4096 \* (5122)\*csm  
  
HARDWARE:  
tEXP(e, L, k) = #modular\_multiplications(e, L) \* tMULMOD(k)  
tMULMOD(k) = chm \* k  
2 \* tEXP(d = 2048, L = 512, k = 1024) = 2 \* 512\*chm \* 2 **\*** 512 = 4 \* 512 \* chm

B. RSA signature verification with e=2^16+1, and the modulus N length 4096 bits.   
SOFTWARE:   
tEXP(e = 216+1 , L = 17, k = 4096) = 17 \* csm \* (4096)2 = 17 \* (8\* 512)2\* csm = 1088 \* (5122) \* csm  
HARDWARE:  
tEXP(e = 216+1 , L = 17, k = 4096) = 17 \* chm \* (4096) = 17 \* 8 \* 512 \* chm = 136 \* 512 \* chm

C. RSA decryption using CRT with the 2048-bit d, 512-bit P, and 1536-bit Q.   
SOFTWARE:  
tEXP(d = 2048, L = 1024, kP = 512) + tEXP(d = 2048, L = 1024, kQ = 1536)   
= 1024 \* csm \* [5122 + (3 \* 512)2]= 1024 \* csm \* (4 \* 512)2  
= 16384 \* (5122) \* csm

HARDWARE:  
tEXP(d = 2048, L = 1024, kP = 512) + tEXP(d = 2048, L = 1024, kQ = 1536)   
= 1024 \* chm \* [512 + (3 \* 512)]= 1024 \* chm \* 4 \* 512 = 4096 \* 512 \* chm

D. RSA decryption without CRT with the 2048-bit d, 512-bit P, and 1536-bit Q.  
SOFTWARE:  
tEXP(d = 2048, L = 512, kP = 512) + tEXP(d = 2048, L = 512, kQ = 1536)   
= 512 \* csm \* [5122 + (3 \* 512)2]= 512 \* csm \* (4 \* 512)2  
= 8192 \* (5122) \* csm

HARDWARE:  
tEXP(d = 2048, L = 512, kP = 512) + tEXP(d = 2048, L = 512, kQ = 1536)   
= 512 \* chm \* [512 + (3 \* 512)]= 512 \* chm \* 4 \* 512 = 2048 \* 512 \* chm

E. RSA encryption with e=5, and the modulus N length 8192 bits.  
SOFTWARE:  
tEXP(e = 5 , L = 3, k = 8192) = 3 \* csm \* (8192)2 = 3 \* (16 \* 512)2\* csm  
 = 768 \* (5122) \* csm

HARDWARE:  
tEXP(e = 5 , L = 3, k = 8192) = 3\* chm\* 8192 = 3 \* (16 \* 512) \* chm = 48 \* 512\* chmF. RSA encryption with e of the size of the modulus N, and N of the size of 1024 bits.  
SOFTWARE:  
tEXP(erandom, L = 1024, k = 1024) = 1024\* csm\* (1024)2 = 1024 \* (2 \* 512)2\* csm = 4096 \* (5122) \* csm  
HARDWARE:  
tEXP(erandom, L = 1024, k = 1024) = 10242 \* chm = 1024 \* (2 \* 512) \* chm = 2048 \* 512 \* chm

Ranking:   
Software: E, B, A=F, D, C  
Hardware: A, E, B, D, C

Problem 15  
Find value of an integer A in the range from 0 to 263, such that a1=A mod 3 = 2, a2 = A mod 8 = 5, and a3= A mod 11 = 4.  
  
N = 264 = n1\*n2\*n3 = 3\*8\*11  
N1 = N/n1 = 264/3 = 88  
N2 = N/n2 = 264/8 = 33  
N3 = N/n3 = 264/11= 24  
A = (a1\*N1(N1-1 mod n1) + a2\*N2(N2-1 mod n2) + a3\*N3(N3-1 mod n3)) mod N  
= (2\*88(88-1 mod 3) + 5\*33(33-1 mod 8) + 4\*24(24-1 mod 11)) mod 264  
(176(88-1 mod 3) + 165(33-1 mod 8) + 96(24-1 mod 11)) mod 264  
(176(1-1 mod 3) + 165(1-1 mod 8) + 96(2-1 mod 11)) mod 264

|  |  |  |  |
| --- | --- | --- | --- |
| i | qi | ri | xi |
| -2 | - | 3 | 0 |
| -1 | 3 | 1 | 1 |
| 0 | - | 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| i | qi | ri | xi |
| -2 | - | 8 | 0 |
| -1 | 8 | 1 | 1 |
| 0 | - | 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| i | qi | ri | xi |
| -2 | - | 11 | 0 |
| -1 | 5 | 2 | 1 |
| 0 | 2 | 1 | -5 |
| 1 | - | 0 |  |

(176\*1 + 165 \* 1 + 96\*6) mod 264 = 125  
A = 125  
Testing with this value, all of the criteria are satisfied!

Problem 16  
How many messages (i.e., values of M in the range between 0 and N-1) are not concealed by RSA (i.e., have the ciphertext value equal to the message value) for the following choices of RSA keys:  
  
σ = (1 + gcd(*e*-1, *P*-1)) · (1 + gcd(*e*-1, *Q*-1))

 a. {e=865; N=10,573}  
(1 + gcd(864, P-1))\*(1 + gcd(864, Q-1))  
P = 97  
Q = 109  
(1 + gcd(864, 96))\*(1 + gcd(864, 108)) =  
(1 + 96) \* (1 + 108) = 97\*109 = 10573  
  
 b. {e=37; N=7,031}  
(1 + gcd(e-1, P-1)) \* (1 + gcd(e-1, Q-1))  
P = 79  
Q = 89  
(1 + gcd(36, 78)) \* (1 + gcd(36, 88))  
(1 + 6) \* (1 + 4) = 7 \* 5 = 35  
  
  
 c. {e=31; N=4,897}.  
(1 + gcd(e-1, P-1)) \* (1 + gcd(e-1, Q-1))  
P = 59  
Q = 83  
(1 + gcd(30, 58)) \* (1 + gcd(30, 82))  
(1 + 2) \* (1 + 2) = 9