University College London
Department of Computer Science

Cryptanalysis Exercises Lab 04

N. Courtois, J. Bootle

February 27, 2019

n.courtois@ucl.ac.uk
Version 1.2

1. Modular Exponentiation

The following function performs modular exponentiation. It computes $a^k \mod n$ and outputs the answer. Two lines are missing: complete the code:

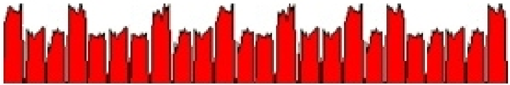
```
def MyPower(a,k,n):
    K = bin(k)[2:]
    A = a % n
    c = 1
    if int(K[0])==1:
        c = ????
    for j in range(1,len(K)):
        c = (c^2) % n
        if int(K[j])==1:
            c = ????
    return c
```

Copy and paste the code into SAGE. This code will be reused in later exercises.

2. A Side Channel Attack on RSA

We recall that RSA encryption is defined by $c = m^e \mod N$. Bob's RSA implementation has public key $(N, e) = (183181, 5)$ where N is a product of two primes p and q. He receives a ciphertext c from Alice. Bob uses the following square-and-multiply algorithm to compute $m = c^d \mod N$.

```
def BobPower(a,k,n):
    K = bin(k)[2:]              # K is binary expansion of k,
    A = a % n                   # with the most significant bit
    c = 1                       #stored in K[0]
    if int(K[0])==1:
        c = (c*A) % n           #modular multiplication here
    for j in range(1,len(K)):
        c = (c^2) % n           #modular squaring is cheap
        if int(K[j])==1:
            c = (c*A) % n       #modular multiplication uses
    return c                    #more power
```



Click on the green letter before each question to get a full solution. Click on the green square to go back to the questions.

Exercise 1.

(a) The power usage of Bob's CPU as he decrypts the ciphertext is given in the graph shown. What value for the decryption exponent d is suggested by the power usage graph?

(b) Using the values of d, e and N, can we compute p and q?

3. Primality Testing

Click on the green letter in front of each sub-question (e.g. (a)) to see a solution. Click on the green square at the end of the solution to go back to the questions.

Click here for a reminder square-and-multiply algorithms.

Exercise 2.

(a) Create a function 'MyPower' which takes inputs a, k and n, and computes $a^k \mod n$ using a square-and-multiply algorithm.

(b) In the Fermat primality test, we test whether a number n is prime by computing $a^{n-1} \mod n$ and then checking whether the result is equal to 1. If the result is not 1, then the number is not prime! Using your function, and the `is_prime` function, find all of the *composite* numbers between 2 and 2000 that pass the Fermat test with $a = 2$. Repeat for $a = 5$.

(c) Using your answer to the previous question, or otherwise, find all of the Carmichael numbers between 2 and 2000. Hint: remember that if $\gcd(a, n) > 1$, then n does not need to pass the Fermat test to base a to be a Carmichael number.

(d) Test any Carmichael numbers that you have found using the Miller-Rabin test, again with $a = 2$ and $a = 5$. Do any of them pass the test?

(e) (Bonus Question) Find a number larger than 5000 which passes the Fermat test with base a, but fails the Miller-Rabin test to base a. Using the sequence of values from the Miller-Rabin test, can you factor the number without using trial division?

4. Rabin Cryptosystem

Click on the green letter in front of each sub-question (e.g. (a)) to see a solution. Click on the green square at the end of the solution to go back to the questions.

EXERCISE 3. Let p, q be two large primes which are congruent to 3 modulo 4. Set $N = pq$.

(a) Let $c \equiv m^2 \in \mathbb{Z}/p\mathbb{Z}$. Set $m' \equiv c^{(p+1)/4} \mod p$. What is $(m')^2$?

(b) The Rabin cryptosystem encrypts a message $m \mod N$ by setting $c \equiv m^2 \mod N$. Suppose that you know p, q. Use the first part of the question to describe how to decrypt a message. Hint: use the Chinese Remainder Theorem.

(c) With a partner, generate two primes which are suitable for the Rabin cryptosystem. Now, using SAGE, write programs which can encrypt and decrypt a message. The CRT command is very useful for this.

5. Continued Fractions and RSA

For any real number r, its **continued fraction representation** is a (possibly infinite) sequence of integers $[q_0; q_1, q_2, \ldots]$ such that

$$r = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{q_4 + \ldots}}}}$$

Click on the green letter before each question to get a full solution. Click on the green square to go back to the questions.

EXERCISE 4.

(a) (Bonus Question) If $r = \frac{a}{b}$, show that the continued fraction representation of r can be computed with Euclid's Algorithm on (a, b).

(b) SAGE contains functions for computing continued fraction expansions. Try "$a = continued\_fraction(pi); a$".

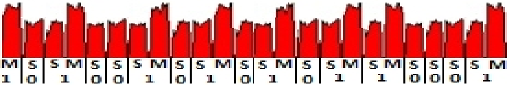
(c) By truncating the continued fraction expansion of a number, we can obtain a rational approximation to that number. The rational number A_n/B_n representing the continued fraction expansion $[q_0; q_1, \ldots, q_n]$ is called the nth convergent. Try "$a.convergent(3)$", and compare the decimal expansion of this number to that of π. To how many decimal places do the two values agree?

(d) (Bonus Question) It is known that if $|r - m/n| < 1/2n^2$, then m/n is a convergent to r. For an RSA public/private key-pair, show that if $N = pq$ with $q < p < 2q$, and $d < N^{1/4}/3$, then k/d is a convergent to e/N, where $ed - 1 = k\phi(N)$.

(e) Let $N = 90581, e = 17993$ be an RSA public-key. Use continued fractions to find d.

Solutions to Exercises

Exercise 1(a) When computing $c^d \mod N$, the square-and-multiply algorithm will either do a squaring operation, or a squaring operation then a multiplication, depending on whether each bit in the binary representation of d is a 0 or a 1. The multiplication is usually more computationally intensive. This means that we can read off the binary representation of d straight from the graph.



This gives us $d = 72357$. □

Exercise 1(b) Since $N = pq$, we know that $\phi(N) = (p-1)(q-1) = pq - p - q + 1$. Thus $p + q = N - \phi(N) + 1$. Furthermore, in RSA, we know that $ed = 1 \mod \phi(N)$. Therefore, $ed - 1 = k\phi(N)$ for some positive integer k.

Now, consider the quadratic equation

$$X^2 - \left(N - \frac{ed-1}{k} + 1\right) X + N = X^2 - (p+q)X + pq = (X-p)(X-q) = 0$$

We already know N, e and d. If we guess values of k, we can try to use the quadratic formula to obtain p and q. Guessing $k = 2$ gives us $X^2 - 2290X + 183181$, and then we recover $p = 2207$ and $q = 83$ from the quadratic formula.

The disadvantage of this approach is that it seems to involve guessing k and we might have given up if k was large and prime.

Here is a second solution. We know that $ed - 1 = k\phi(n)$. For any a with $\gcd(a, N) > 1$, we have $a^{\phi(N)} \equiv 1 \mod N$. Substituting in the values of e and d, we know that $a^{361784} \equiv 1 \mod N$. Taking inspiration from the Miller-Rabin test, we can use this fact to try and find square roots of 1 not congruent to $\pm 1 \mod N$.

We divide 361784 by 2 as many times as possible, to get 45223. Now, we pick a random value of a between 1 and $N-1$. We check that $\gcd(a, N) = 1$ (if not, we have already factored N). Then, we raise to the power $45223 \mod N$, and then square repeatedly, hoping that we get a non-trivial square-root. For example, with $a = 2$, we get $A = 97109$, and find that $A^2 \equiv 1 \mod N$. Therefore, $(A+1)(A-1) \equiv 0 \mod N$, and $\gcd(A \pm 1, N)$ give factors of N. Finally, $\gcd(97110, 183181) = 83$ and $183181 = 83 \times 2207$.

It can be shown, using the Chinese Remainder Theorem, that this approach has a success probability of roughly $\frac{1}{2}$, in the case that N is a product of two distinct primes. □

Exercise 2(a) The following code implements the square-and-multiply Algorithm.

```
def MyPower(a,k,n):
    K = bin(k)[2:]
    A = a % n
    c = (A^ int(K[0]))
    for j in range(1,len(K)):
        c = (c^ 2) % n
        c = c*(A^ int(K[j])) % n
    return c
```

□

Exercise 2(b) The following code finds the answer for $a = 2$. For $a = 2$ you should get $341, 561, 645, 1105, 1387, 1729, 1905$. For $a = 5$, you should get $4, 124, 217, 561, 781, 1541, 1729, 1891$.

```
for i in range(2,2000):
    if is_prime(i)==False and MyPower(2,i-1,i)==1:
        print(i)
```

□

Exercise 2(c) The Carmichael numbers between 2 and 2000 are $561, 1105, 1729$. □

Exercise 2(d) The following code carries out the Miller-Rabin test to base a. You should find that none pass with either $a = 2$ or $a = 5$.

```
def StrongTest(a,n):
    if (n%2)==0:
        return 'fail'
    b = n-1
    k=0
    while (b%2)==0:
        b = b/2
        k = k+1
    A = MyPower(a,b,n)
    if A == 1 or A == (n-1):
        return 'pass'
    for i in range(0,k):
        A = MyPower(A,2,n)
        if A == (n-1):
            return 'pass'
```

(code continues on the next page)

```
        if A == 1:
            return 'fail'
    return 'fail'
```

□

Exercise 2(e) The number 5461 passes the Fermat test with base $a = 2$, but fails the Miller-Rabin test. From this, we can deduce that the sequence of values produced by the Miller-Rabin test ends in 1, but does not contain -1. Therefore, the sequence gives us a square-root 128 of 1 modulo 5461 which is not ± 1. We have $128^2 \equiv 1 \mod 5461$. Rearranging, $(128 + 1)(128 - 1) \equiv 0 \mod 5461$, but 128 is not congruent to ± 1. Therefore, $\gcd(129, 5461)$ and $\gcd(127, 5461)$ give non-trivial factors of 5461. We find that $5461 = 43 \times 127$. □

Exercise 3(a) By Fermat's Little Theorem, we have that $(m')^2 \equiv c \mod p$. □

Exercise 3(b) We can compute $c_p \equiv c \mod p$ and $c_q \equiv c \mod q$. Using the first part of the question, we can compute the square roots m_p with $m_p^2 = c_p \mod p$ and $m_q^2 = c_q \mod q$. Finally, we can use the Chinese Remainder Theorem to compute $m \mod N$ from $m_p \mod p$ and $m_q \mod q$. □

Exercise 4(a) Using Euclid's Algorithm, we find integers $r_0, r_1, r_2, \ldots$ such that:

$$\begin{aligned} a &= q_0 b + r_0 \\ b &= q_1 r_0 + r_1 \\ r_0 &= q_2 r_1 + r_2 \\ &\vdots \\ r_{n-1} &= q_{n+1} r_n \end{aligned}$$

We substitute the expression for a into $\frac{a}{b}$ and rearrange to get

$$\frac{a}{b} = \frac{q_0 b + r_0}{b} = q_0 + \frac{r_0}{b} = q_0 + \frac{1}{\frac{b}{r_0}}$$

We can then substitute the expression for b and rearrange in a similar way to get

$$\frac{a}{b} = q_0 + \frac{1}{q_1 + \frac{1}{\frac{r_0}{r_1}}}$$

Repeating the same idea, we eventually arrive at

$$r = q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{q_3 + \frac{1}{q_4 + \ldots + \frac{1}{q_{n+1}}}}}}$$

□

Exercise 4(b) SAGE should display "[3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 2, ...]". □

Exercise 4(c) The third convergent to π is $355/113$, which approximates π to 6 decimal places. □

Exercise 4(e) The first convergent is $1/5$, which shows that $d = 5$. □