University College London
Department of Computer Science

Cryptanalysis Lab 2

J. P. Bootle

January 20, 2017

jonathan.bootle.14@ucl.ac.uk
Version 2.0

Cyclic Groups

Click on the "Ans" button to get a hint.

Shift-click on "Ans" buttons that have a green boundary to get a full solution. Click on the green square to go back to the questions.

Quiz

1. How many elements in $(\mathbb{Z}/11\mathbb{Z})^*$?

$=$ [ ] Ans

2. Find a single element that generates $(\mathbb{Z}/11\mathbb{Z})^*$.

$=$ [ ] Ans

3. What is the order of 5 in $(\mathbb{Z}/11\mathbb{Z})^*$?

$=$ [ ] Ans

Clear

Exercise 1.

(a) Let p be a prime such that $p = 2q + 1$, where q is also prime. We call p with this property a 'strong' prime or 'safe' prime. Let g

be a generator of $(\mathbb{Z}/p\mathbb{Z})^*$. How can we generate a group of order q?

Cyclic Groups in SAGE

Try out the following sequence of SAGE commands, and verify that the first 3 results match with your answers to the first 3 questions.

EXERCISE 2.

(a) euler_phi(11)

(b) primitive_root(11)

(c) To find the order of 5:

```
R = Integers(11)
a = R(5)
a.multiplicative_order()
```

(d) Compute (easy) discrete logarithms:

```
R = Integers(11)
a = R(5)
b = a*a*a*a
a.log(b)
```

(e) Compute modular square roots:

```
R = Integers(7)
a = R(3)
b = a*a
mod(2,7).sqrt()
```

The Fermat Factorisation Algorithm

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 3.

(a) Given that $1309 = 47^2 - 30^2$, what is the prime factorisation of 1309?

(b) Let N, a, b be odd, positive integers such that $N = ab$. Show that N can be expressed as the difference between two square numbers.

(c) The incomplete function 'Fermat' implements a factorisation algorithm. The function takes input N, and should output a, b such

that $N = ab$. Please fill in the question marks to obtain a complete implementation of the Fermat factorisation algorithm.

```
def fermat(N):
    n = ceil(sqrt(N))
    while ???:
        M = n*n-N
        m = floor(sqrt(M))
        if m == sqrt(M):
            return ???
        n = n+1
```

(d) Use your completed code to find the factors of $N = 1488391, 1467181, 1456043$. Can you see a connection between the running time of your code and the prime factors of N?

Polynomials in SAGE

EXERCISE 4.

(a) Try out the following sequence of SAGE commands.

```
ZP.<x> = ZZ[]
(x^5 + 3*x^2 - 2*x + 7) // (x + 1)
(x^5 + 3*x^2 - 2*x + 7).quo_rem(x + 1)
gcd(3*x^2 + 6*x - 9, 5*x^3 - 2*x + 2)
factor(3*x^5 + 5*x - 8)
(3*x^5 + 5*x - 8).factor_mod(3)
```

Elliptic Curves

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 5. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. Write $+$ for the operation of adding two points. Beware: $P + Q \neq (x_1 + x_2, y_1 + y_2)$!

(a) Watch the tutorial on elliptic curve point addition at https://www.youtube.com/watch?v=XmygBPb7DPM.

(b) Browse the internet to find the formulae for the coordinates of

$P + Q$ when $P \neq Q$. What about when $P = Q$? You can assume that $Q \neq (x_1, -y_1)$ since things are slightly different in this case.

(c) Let $E : y^2 = x^3 + 3x + 3$ be an elliptic curve, defined over $\mathbb{F}_7$. Two points on the curve are $P = (4, 3)$ and $Q = (3, 2)$. Verify that $2^*P = Q$ (remember that $2^*P = P + P$).

(d) Construct E, P, Q in SAGE using the following commands. Check your answer to the previous part by typing $2 * P$ (the answer will have three coordinates, for reasons to be explained in lectures, but ignore the last coordinate). What is $P + Q$?

```
p = 7
E = EllipticCurve( GF(p),[3,3] )
P = E(4,3)
Q = E(3,2)
```

(e) Type *E.cardinality*() to find out how many points lie on this elliptic curve.

(f) Type *E.gens*() to obtain a set of points which generate all points in the elliptic curve group.

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 6. 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.

Smooth Numbers

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 7. Smooth numbers are useful in index calculus attacks for factorising and computing discrete logarithms. A number n is B-smooth if all of the prime factors of n are $\leq B$. Let $\Psi(B, N)$ be the number of B-smooth numbers that are $\leq N$.

(a) Write a program to find $\Psi(B, N)/N$ for $(B, N) = (10, 10^{10})$, $(15, 10^7)$,$(100, 10^4)$.

Some tips: Try to write a program which efficiently generates the smooth numbers $\leq N$ from the primes $\leq B$, for example, by computing products of these primes and checking if they are smaller than N. This will be much faster than a program which factorises each number $\leq N$ and checks whether the prime factors are $\leq B$. If you want to be extremely efficient, try to think of a clever way to avoid storing all of the numbers, and alternatives to computing lots and lots of products.

(b) We have the approximation $\Psi(B,N) \approx \frac{1}{\pi(B)!}\prod_{p\le B}\frac{\log N}{\log p}$, where $\pi(B)$ is the number of primes $\le B$. Compare the approximate values of Ψ/N with the true values computed by your program. How close are these to the values you computed?

Solutions to Exercises

Exercise 1(a) The order of g is $\phi(p) = p - 1 = 2q$. We can compute $g^2 \mod p$, and this element will have order q, generating a subgroup of size q. □

Exercise 3(a) We have $1309 = (47 + 30)(47 - 30) = 77 \cdot 17$. □

Exercise 3(b) Write $N = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2$. Each bracketed expression is a whole number, because N is odd, so a, b are both odd, and therefore $a \pm b$ is even. □

Exercise 3(c) The following code implements the Fermat Factorisation algorithm.

```
def fermat(N):
    n = ceil(sqrt(N))
    while True:
        M = n*n-N
        m = floor(sqrt(M))
        if m == sqrt(M):
            return [n+m,n-m]
        n = n+1
```

□

Exercise 3(d) The Fermat factorisation method finds factors of N as $n+m$ and $n-m$, where $N = n^2 - m^2$. The value of $n+m$ is at least $\sqrt{N}$ and increases as n is incremented. Therefore, Fermat factorisation runs fastest on integers N which have factors close to $\sqrt{N}$. □

Exercise 5(b) If $P \neq Q$, we set $s = (y_1 - y_2)(x_1 - x_2)^{-1}$. If $P = Q$, we take $s = (3x_1^2 + a)(2y_1)^{-1}$. Then, $(x_3, y_3) = (x_1, y_1) \oplus (x_2, y_2)$, where $x_3 = s^2 - x_1 - x_2$, and $y_3 = s(x_1 - x_3) - y_1$.

These formulae come from the definition of addition on an elliptic curve that you saw in the video. This uses different points of intersection between straight lines and the curve. □

Exercise 5(c) Substituting the coordinates of P into the correct formula from the previous part shows that $2^*P = Q$. □

Exercise 5(d) You should find that $P + Q = (1, 0)$. □

Exercise 6(a) By Fermat's Little Theorem, we have that $(m')^2 \equiv c \mod p$. □

Exercise 6(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 7(a) The following code counts smooth numbers.

```
def CountSmooth(B,N):
    P = Primes()
    prime = 3
    prime_list = [2]
    while prime ¡= B:
        prime_list.append(prime)
        prime = next_prime(prime)
    smooth_numbers = [1]
    for number in smooth_numbers:
        for prime in prime_list:
            n = number*prime
            if not (n in smooth_numbers):
                if n ¡= N:
                    smooth_numbers.append(n)
    return len(smooth_numbers)-1
```

□

Solutions to Quizzes

Solution to Quiz: The number of elements in $(\mathbb{Z}/N\mathbb{Z})^*$ is $\phi(N)$, so in this case, the answer is $\phi(11) = 10$. ■

Solution to Quiz: If we compute the powers of 2 modulo 11, we get $2, 4, 8, 5, 10, 9, 7, 3, 6, 1$, so 2 is a generator. Alternatively, by Lagrange's Theorem, the order of an element divides the size of the group. The size of the group is 10, so the only possibilities for the order of an element are 1, 2, 5, and 10. A group generator should have order 10 to generate every group element, so to check that 2 is a generator, we just have to check that $2^2 \neq 1 \mod 11$, and $2^5 \neq 1 \mod 11$, implying that 2 has order 10. ■

Solution to Quiz: The smallest n such that $5^n = 1 \mod 11$ is $n = 5$.Alternatively, by Lagrange's Theorem, the order of an element divides the size of the group. The size of the group is 10, so the only possibilities for the order of an element are 1, 2, 5, and 10. Therefore, it is enough to check that $5^2 \neq 1 \mod 11$, and $5^5 = 1 \mod 11$. ■