ECE 5200 Exercise 4

**Date:** 4/21/20 Tuesday, week 13

**Due Date:** 5/5/20 Tuesday, week 15

**Total Points:**  58 points

## More on Elliptic Curve Cryptography

1. (12%) In Stalling’s book you have seen E 23(1,1) or the set of all integer solutions of y2 = x3 + x + 1 as in table 10.1 below (there are 27 points of this curve)

**Generate and present** tables of points of the format E p(1,1) for odd prime numbers p starting from 11 until 29.

Prime number: 11

Number of points: 13

Points: [(0, 1), (0, 10), (1, 5), (1, 6), (2, 0), (3, 3), (3, 8), (4, 5), (4, 6), (6, 5), (6, 6), (8, 2), (8, 9)]

Prime number: 13

Number of points: 17

Points: [(0, 1), (0, 12), (1, 4), (1, 9), (4, 2), (4, 11), (5, 1), (5, 12), (7, 0), (8, 1), (8, 12), (10, 6), (10, 7), (11, 2), (11, 11), (12, 5), (12, 8)]

Prime number: 17

Number of points: 17

Points: [(0, 1), (0, 16), (4, 1), (4, 16), (6, 6), (6, 11), (9, 5), (9, 12), (10, 5), (10, 12), (11, 0), (13, 1), (13, 16), (15, 5), (15, 12), (16, 4), (16, 13)]

Prime number: 19

Number of points: 20

Points: [(0, 1), (0, 18), (2, 7), (2, 12), (5, 6), (5, 13), (7, 3), (7, 16), (9, 6), (9, 13), (10, 2), (10, 17), (13, 8), (13, 11), (14, 2), (14, 17), (15, 3), (15, 16), (16, 3), (16, 16)]

Prime number: 23

Number of points: 27

Points: [(0, 1), (0, 22), (1, 7), (1, 16), (3, 10), (3, 13), (4, 0), (5, 4), (5, 19), (6, 4), (6, 19), (7, 11), (7, 12), (9, 7), (9, 16), (11, 3), (11, 20), (12, 4), (12, 19), (13, 7), (13, 16), (17, 3), (17, 20), (18, 3), (18, 20), (19, 5), (19, 18)]

Prime number: 29

Number of points: 35

Points: [(0, 1), (0, 28), (6, 7), (6, 22), (8, 12), (8, 17), (10, 5), (10, 24), (11, 3), (11, 26), (12, 1), (12, 28), (13, 6), (13, 23), (14, 2), (14, 27), (16, 13), (16, 16), (17, 1), (17, 28), (18, 14), (18, 15), (19, 8), (19, 21), (22, 12), (22, 17), (24, 4), (24, 25), (25, 7), (25, 22), (26, 0), (27, 7), (27, 22), (28, 12), (28, 17)]

**Diffie Hellman Key Exchange (36%)**

1. (18%) Diffie Hellman exchange is covered on page 314 (section 10.1, 7th edition) of Stallings with prime number q = 353,  is a primitive root of q, in this case  = 3 (this means  k = 1 (mod 353) for k = 352, and  m ≠ 1 (mod 353) for 1 <= m < 352).

In Stalling’s book, A(lice) and B(ob) select secret keys XA = 97 and XB = 233, respectively.

Then A computes YA = 3 97 mod 353 = 40 and

B computes YB = 3 233 mod 353 = 248.

Finally A computes K = (YB) XA mod 353 = 248 97 mod 353 = 160 that is the same as K = (YA) XB mod 353 = 40 233 mod 353 = 160 computed by B.

Now use the number q2 = 1997.

1. **Prove** that 1997 is a prime. In case 1997 is not a prime from your computation, **pick** the smallest integer n bigger than 1997 so that n is a prime (**verify** that n is a prime).

SQRT(1997) = 44.687, and every non-prime number must have a factor that is less than the square root other than 1

Proof: 

Since the only one of those integers divides 1997 evenly, namely 1, this tells us that 1997 is prime.

1. **Find** a primitive root of 1997. Note 3 may or may not be a primitive root of 1997.

Primitives of 1997: 

**Proof that 21 is a primitive element of 1997**

Number of Power elements for 21: 1996, i.e. {1,2,3,…,1996} when sorted

Power Elements of 21: 

(c/d) Using XA = 97 and XB = 233 as Stallings, calculate YA and YB for 1997. Finally calculate K for A and then K for B. Do they agree?

For Prime number 1997 and primitive 21 for private key a:97 and b:233, the Diffie-Hellman Key Exchange outputs: {'KpubA': 869, 'KpubB': 1522, 'Kab': 681, 'Kba': 681}

1. (18%) Repeat question 2. This time you pick a prime number p yourself of at least 6 digits. Add some randomness by adding to 1,000,000 (one million) your birth date d as a Julian day in the year. Your prime number p must be bigger than 1,000,000 + d.

Repeat all the calculations of Q2 (primitive root, YA, YB, and K).

Prime number chosen: 1,000,000 + 336 + 21 = 1,000,357 = p

And the prime factors of p-1 (1,000,356) are: 2, 3, 7, 11909

As such the proof that 2 is a primitive element of 1000357

2^(1000356/2) = 2^(500178) = 1000356 mod 1000357 != 1 mod 1000357

2^(1000356/3) = 2^(333452) = 32609 mod 1000357 != 1 mod 1000357

2^(1000356/7) = 2^(142908) = 333490 mod 1000357 != 1 mod 1000357

2^(1000356/11909) = 2^(84) = 157221 mod 1000357 != 1 mod 1000357

For Prime number 1000357 and primitive 2 for private key a:97 and b:233, the Diffie-Hellman Key Exchange outputs: {'KpubA': 494973, 'KpubB': 961803, 'Kab': 311818, 'Kba': 311818}

1. (10%) Primitive Roots

In chapter 2.8 Discrete Logarithms, page 75, table 2.7, you have learned the primitive roots of 19 as 2, 3, 5, 10, 13, 14, and 15.

Write a computer program to compute and display all the primitive roots (and also the number of primitive roots) for the prime numbers from 100 (one hundred) to 200 (two hundreds)

**Code:**

**def** **primeCheck**(number):  
 k=0  
 **for** i **in** range(2,number//2+1):  
 **if**(number%i==0):  
 k=k+1  
 **if**(k<=0):  
 **return** True  
 **else**:  
 **return** False  
   
**def** **powerElements**(a,p):  
 powers = []  
 **for** i **in** range (1,p):  
 power = pow(a,i,p)  
 **if** power **not** **in** powers:  
 powers.append(power)  
 **return** powers  
   
**def** **cyclicGroupOrder**(p):  
 order = []  
 **for** i **in** range (1,p):  
 powers = powerElements(i,p)  
 order.append(len(powers))  
 **return** order  
   
**def** **primitveElements**(p):  
 primitive = []  
 **for** i **in** range (1,p):  
 powers = powerElements(i,p)  
 **if**(len(powers) == (p-1)):  
 primitive.append(i)  
 **return** primitive  
   
**def** **e4p4**():  
 **for** p **in** range (100,200+1):  
 **if**(primeCheck(p)):  
 primitives = primitveElements(p)  
 print("Prime number: {0}" .format(p))  
 print("Number of Primitives: {0}" .format(len(primitives)))  
 print("Primitives: {0}" .format(primitives))  
 print()

**Output:**

Prime number: 101

Number of Primitives: 40

Primitives: [2, 3, 7, 8, 11, 12, 15, 18, 26, 27, 28, 29, 34, 35, 38, 40, 42, 46, 48, 50, 51, 53, 55, 59, 61, 63, 66, 67, 72, 73, 74, 75, 83, 86, 89, 90, 93, 94, 98, 99]

Prime number: 103

Number of Primitives: 32

Primitives: [5, 6, 11, 12, 20, 21, 35, 40, 43, 44, 45, 48, 51, 53, 54, 62, 65, 67, 70, 71, 74, 75, 77, 78, 84, 85, 86, 87, 88, 96, 99, 101]

Prime number: 107

Number of Primitives: 52

Primitives: [2, 5, 6, 7, 8, 15, 17, 18, 20, 21, 22, 24, 26, 28, 31, 32, 38, 43, 45, 46, 50, 51, 54, 55, 58, 59, 60, 63, 65, 66, 67, 68, 70, 71, 72, 73, 74, 77, 78, 80, 82, 84, 88, 91, 93, 94, 95, 96, 97, 98, 103, 104]

Prime number: 109

Number of Primitives: 36

Primitives: [6, 10, 11, 13, 14, 18, 24, 30, 37, 39, 40, 42, 44, 47, 50, 51, 52, 53, 56, 57, 58, 59, 62, 65, 67, 69, 70, 72, 79, 85, 91, 95, 96, 98, 99, 103]

Prime number: 113

Number of Primitives: 48

Primitives: [3, 5, 6, 10, 12, 17, 19, 20, 21, 23, 24, 27, 29, 33, 34, 37, 38, 39, 43, 45, 46, 47, 54, 55, 58, 59, 66, 67, 68, 70, 74, 75, 76, 79, 80, 84, 86, 89, 90, 92, 93, 94, 96, 101, 103, 107, 108, 110]

Prime number: 127

Number of Primitives: 36

Primitives: [3, 6, 7, 12, 14, 23, 29, 39, 43, 45, 46, 48, 53, 55, 56, 57, 58, 65, 67, 78, 83, 85, 86, 91, 92, 93, 96, 97, 101, 106, 109, 110, 112, 114, 116, 118]

Prime number: 131

Number of Primitives: 48

Primitives: [2, 6, 8, 10, 14, 17, 22, 23, 26, 29, 30, 31, 37, 40, 50, 54, 56, 57, 66, 67, 72, 76, 82, 83, 85, 87, 88, 90, 93, 95, 96, 97, 98, 103, 104, 106, 110, 111, 115, 116, 118, 119, 120, 122, 124, 126, 127, 128]

Prime number: 137

Number of Primitives: 64

Primitives: [3, 5, 6, 12, 13, 20, 21, 23, 24, 26, 27, 29, 31, 33, 35, 40, 42, 43, 45, 46, 47, 48, 51, 52, 53, 54, 55, 57, 58, 62, 66, 67, 70, 71, 75, 79, 80, 82, 83, 84, 85, 86, 89, 90, 91, 92, 94, 95, 97, 102, 104, 106, 108, 110, 111, 113, 114, 116, 117, 124, 125, 131, 132, 134]

Prime number: 139

Number of Primitives: 44

Primitives: [2, 3, 12, 15, 17, 18, 19, 21, 22, 26, 32, 40, 50, 53, 56, 58, 61, 68, 70, 72, 73, 85, 88, 90, 92, 93, 98, 101, 102, 104, 108, 109, 110, 111, 114, 115, 119, 123, 126,

128, 130, 132, 134, 135]

Prime number: 149

Number of Primitives: 72

Primitives: [2, 3, 8, 10, 11, 12, 13, 14, 15, 18, 21, 23, 27, 32, 34, 38, 40, 41, 43, 48, 50, 51, 52, 55, 56, 57, 58, 59, 60, 62, 65, 66, 70, 71, 72, 74, 75, 77, 78, 79, 83, 84, 87, 89, 90, 91, 92, 93, 94, 97, 98, 99, 101, 106, 108, 109, 111, 115, 117, 122, 126, 128, 131, 134, 135, 136, 137, 138, 139, 141, 146, 147]

Prime number: 151

Number of Primitives: 40

Primitives: [6, 7, 12, 13, 14, 15, 30, 35, 48, 51, 52, 54, 56, 61, 63, 71, 77, 82, 89, 93, 96, 102, 104, 106, 108, 109, 111, 112, 114, 115, 117, 120, 126, 129, 130, 133, 134, 140, 141, 146]

Prime number: 157

Number of Primitives: 48

Primitives: [5, 6, 15, 18, 20, 21, 24, 26, 34, 38, 43, 53, 55, 60, 61, 62, 63, 66, 69, 70, 72, 73, 74, 77, 80, 83, 84, 85, 87, 88, 91, 94, 95, 96, 97, 102, 104, 114, 119, 123, 131, 133, 136, 137, 139, 142, 151, 152]

Prime number: 163

Number of Primitives: 54

Primitives: [2, 3, 7, 11, 12, 18, 19, 20, 29, 32, 42, 44, 45, 50, 52, 63, 66, 67, 68, 70, 72, 73, 75, 76, 79, 80, 82, 89, 92, 94, 101, 103, 106, 107, 108, 109, 112, 114, 116, 117, 120, 122, 124, 128, 129, 130, 137, 139, 147, 148, 149, 153, 154, 159]

Prime number: 167

Number of Primitives: 82

Primitives: [5, 10, 13, 15, 17, 20, 23, 26, 30, 34, 35, 37, 39, 40, 41, 43, 45, 46, 51, 52, 53, 55, 59, 60, 67, 68, 69, 70, 71, 73, 74, 78, 79, 80, 82, 83, 86, 90, 91, 92, 95, 101, 102, 103, 104, 105, 106, 109, 110, 111, 113, 117, 118, 119, 120, 123, 125, 129, 131, 134, 135, 136, 138, 139, 140, 142, 143, 145, 146, 148, 149, 151, 153, 155, 156, 158, 159, 160, 161, 163, 164, 165]

Prime number: 173

Number of Primitives: 84

Primitives: [2, 3, 5, 7, 8, 11, 12, 17, 18, 19, 20, 26, 27, 28, 30, 32, 39, 42, 44, 45, 46, 48, 50, 53, 58, 59, 61, 62, 63, 65, 66, 68, 69, 70, 71, 72, 74, 75, 76, 79, 82, 86, 87, 91, 94, 97, 98, 99, 101, 102, 103, 104, 105, 107, 108, 110, 111, 112, 114, 115, 120, 123, 125, 127, 128, 129, 131, 134, 141, 143, 145, 146, 147, 153, 154, 155, 156, 161, 162, 165, 166, 168, 170, 171]

Prime number: 179

Number of Primitives: 88

Primitives: [2, 6, 7, 8, 10, 11, 18, 21, 23, 24, 26, 28, 30, 32, 33, 34, 35, 37, 38, 40, 41, 44, 50, 53, 54, 55, 58, 62, 63, 69, 71, 72, 73, 78, 79, 84, 86, 90, 91, 92, 94, 96, 97, 98, 99, 102, 103, 104, 105, 109, 111, 112, 113, 114, 115, 118, 119, 120, 122, 123, 127, 128, 130, 131, 132, 133, 134, 136, 137, 140, 143, 148, 150, 152, 154, 157, 159, 160, 162, 163, 164, 165, 166, 167, 170, 174, 175, 176]

Prime number: 181

Number of Primitives: 48

Primitives: [2, 10, 18, 21, 23, 24, 28K, 41, 47, 50, 53, 54, 57, 58, 63, 66, 69, 76, 77, 78, 83, 84, 85, 90, 91, 96, 97, 98, 103, 104, 105, 112, 115, 118, 123, 124, 127, 128, 131,

134, 140, 153, 157, 158, 160, 163, 171, 179]

Prime number: 191

Number of Primitives: 72

Primitives: [19, 21, 22, 28, 29, 33, 35, 42, 44, 47, 53, 56, 57, 58, 61, 62, 63, 71, 73, 74, 76, 83, 87, 88, 89, 91, 93, 94, 95, 99, 101, 105, 106, 110, 111, 112, 113, 114, 116, 119, 123, 124, 126, 127, 131, 132, 137, 140, 141, 143, 145, 146, 148, 151, 157, 164, 165, 167, 168, 171, 173, 174, 175, 176, 178, 179, 181, 182, 183, 187, 188, 189]

Prime number: 193

Number of Primitives: 64

Primitives: [5, 10, 15, 17, 19, 22, 26, 30, 34, 37, 38, 40, 41, 44, 45, 47, 51, 52, 53, 57, 58, 61, 66, 70, 73, 77, 78, 79, 80, 82, 90, 91, 102, 103, 111, 113, 114, 115, 116, 120, 123, 127, 132, 135, 136, 140, 141, 142, 146, 148, 149, 152, 153, 155, 156, 159, 163, 167, 171, 174, 176, 178, 183, 188]

Prime number: 197

Number of Primitives: 84

Primitives: [2, 3, 5, 8, 11, 12, 13, 17, 18, 21, 27, 30, 31, 32, 35, 38, 44, 45, 46, 48, 50, 52, 56, 57, 58, 66, 67, 71, 72, 73, 74, 75, 78, 79, 80, 82, 86, 89, 91, 94, 95, 98, 99, 102, 103, 106, 108, 111, 115, 117, 118, 119, 122, 123, 124, 125, 126, 130, 131, 139, 140, 141, 145, 147, 149, 151, 152, 153, 159, 162, 165, 166, 167, 170, 176, 179, 180, 184, 185, 186, 189, 192, 194, 195]

Prime number: 199

Number of Primitives: 60

Primitives: [3, 6, 15, 22, 30, 34, 38, 39, 41, 44, 48, 54, 68, 69, 71, 73, 75, 77, 84, 87, 95, 97, 99, 105, 108, 110, 113, 118, 119, 120, 127, 129, 133, 134, 142, 143, 146, 148, 149, 150, 152, 153, 154, 163, 164, 166, 167, 168, 170, 173, 176, 179, 183, 185, 186, 189, 190, 192, 195, 197]

Appendix (Code)

**def** **ECpoints**(a,b,p):  
 points = []  
 **for** x **in** range (0,p):  
 **for** y **in** range (0,p):  
 lhs = pow(y,2,p)  
 rhs = (pow(x,3,p) + a\*x + b) % p  
 **if**(lhs == rhs):  
 points.append((x,y))  
 **return** points  
  
**def** **powerElements**(a,p):  
 powers = []  
 **for** i **in** range (1,p):  
 power = pow(a,i,p)  
 **if** power **not** **in** powers:  
 powers.append(power)  
 **return** powers  
  
**def** **cyclicGroupOrder**(p):  
 order = []  
 **for** i **in** range (1,p):  
 powers = powerElements(i,p)  
 order.append(len(powers))  
 **return** order  
  
**def** **primitveElements**(p):  
 primitive = []  
 **for** i **in** range (1,p):  
 powers = powerElements(i,p)  
 **if**(len(powers) == (p-1)):  
 primitive.append(i)  
 **return** primitive  
   
**def** **DHP**(alpha,a,b,p):  
 ans = dict()  
 ans['KpubA'] = pow(alpha,a,p)  
 ans['KpubB'] = pow(alpha,b,p)  
  
 ans['Kab'] = pow(ans['KpubB'],a,p)  
 ans['Kba'] = pow(ans['KpubA'],b,p)  
  
 **return** ans  
  
**def** **primeCheck**(number):  
 k=0  
 **for** i **in** range(2,number//2+1):  
 **if**(number%i==0):  
 k=k+1  
 **if**(k<=0):  
 **return** True  
 **else**:  
 **return** False

**def** **primeFactors**(number):  
 factors = []  
 **for** i **in** range(2, number + 1):  
 **if**(number % i == 0):  
 prime = True  
 **for** j **in** range(2, (i//2 + 1)):  
 **if**(i % j == 0):  
 prime = False  
 **break**   
 **if** (prime):  
 factors.append(i)  
 **return** factors  
  
**def** **e4p13**():  
 **for** p **in** range (11,29+1):  
 **if**(primeCheck(p)):  
 points = ECpoints(1,1,p)  
 print("Prime number: {0}" .format(p))  
 print("Number of points: {0}" .format(len(points)))  
 print("Points: {0}" .format(points))  
 print()  
  
**def** **e4p2**():  
 a = 97  
 b = 233  
 p = 1997  
 primitive = 21  
 powers = powerElements(primitive,p)  
 print("Proof that {0} is a primitive element of {1}:\nNumber of Power elements for {0}: {2}\nPower Elements of {0}: {3}" .format(primitive,p,len(powers),powers))  
 print("For Prime number {0} and primitive {1} for private key a:{2} and b:{3}, the Diffie-Hellman Key Exchange outputs: {4}" .format(p,primitive,a,b,DHP(primitive,a,b,p)))  
  
**def** **e4p3**():  
 a = 97  
 b = 233  
 p = 1000000+336+21  
 primitive = 2  
 dhke = DHP(primitive,a,b,p)  
 pFactors = primeFactors(p-1)  
 check = []  
 **for** i **in** range(0,len(pFactors)):  
 check.append(pow(primitive,int((p-1)/pFactors[i]),p))  
 print("Proof that {0} is a primitive element of {1}" .format(primitive,p))  
 **for** i **in** range(0,len(pFactors)):  
 print("{0}^({1}/{2}) = {0}^({3}) = {4} mod {5} != 1 mod {5}" .format(primitive,p-1,pFactors[i],int((p-1)/pFactors[i]),check[i],p))  
 print("For Prime number {0} and primitive {1} for private key a:{2} and b:{3}, the Diffie-Hellman Key Exchange outputs: {4}" .format(p,primitive,a,b,dhke))  
  
**def** **e4p4**():  
 **for** p **in** range (100,200+1):  
 **if**(primeCheck(p)):  
 primitives = primitveElements(p)  
 print("Prime number: {0}" .format(p))  
 print("Number of Primitives: {0}" .format(len(primitives)))  
 print("Primitives: {0}" .format(primitives))  
 print()

part = "start"  
**while**(part != "exit"):  
 part = input("Which question would you like to display? [13, 2, 3, or 4]: ")  
 **if**(part == "13"):  
 e4p13()  
 **elif**(part == "2"):  
 e4p2()  
 **elif**(part == "3"):  
 e4p3()  
 **elif**(part == "4"):  
 e4p4()  
 **elif**(part == "all"):  
 e4p13()  
 e4p2()  
 e4p3()  
 e4p4()