**Homework for Section#2 (Approximate due date Aug 31st , 2020)**

1. Pick one prime number ***p*** greater than 5
   1. Generate the multiplication tables mod ***p***
   2. Find the inverses mod ***p*** for all numbers k ∈ :{1, 2, …, p-1}
   3. Validate Fermat: (using arithmetic of 2.1)
   4. Validate Wilson: (p-1)!
   5. Check that
2. Compute the Euler parameter
   1. Generate the multiplication tables mod 3***p***
   2. Find the inverses all numbers k ∈ : {*, , …,* }
   3. Verify Euler:
   4. Check: (3p-1)!
   5. Check (3p-1)! – (p-1)! ≡1 mod 3p

**Homework for section 9B**

Based on the circular group **ℱ**

***≡++1 mod P(x)***

Irreducible ***P()*= + +1**

Generator ***g = (0010)***

Primitive element: P = (, )

Verify several points:

Point doubling 1: (, ) + (, )

Point doubling 2: (, ) + (, )

Point addition 1: (, ) + (, )

Point addition 2: (, ) + (, )

**Homework for Section#3 (approximate due date Sept 7th, 2020)**

-Decipher the following cipher text, and uncover the key for Caesar:

**Alphabet:** abcdefghijklmnopqrstuvwxyz1234567;,./

Cipher text 🡺U742 ,wt P7pqxr ;xu7 awxrw 2tp3 dt74n,wx; ,.73ts x3,4 rx5wt7 u47 p r43rtp1ts 2t;;pvtn P1;4 ,.73ts x3,4 rxu7t ,wt rwxuu7t x3

u7t3rw awxrw 2tp3 3.2qt7n

- Use a 5-bit key to encrypt this message with Vigenere

Link: <https://cryptii.com/pipes/caesar-cipher>

**Homework for section#4 (approximate due date Sept 7th, 2020)**

- Find the 32-bit substitutions of the following 48-bit data streams with the S-Box of DES:

Stream #1 🡺111010 001010 011010 101001 101110 110101 010001 110010

Stream #2 🡺010101 110011 011011 101110 110101 101011 101010 111001

* Using DES key processor, find the first 4 sub-keys of:

Key#1 🡺0110110101100101011010010111010101010001101001101101000101010110

Key#2 🡺1101001001101011001000110101010101110110010011101011101010001101

* Using DES online tool, find the cipher text of the following data streams:

Stream#1 🡺1010011001001110010011010101110010100110101011001101101001001011

Stream#2 🡺0111000100111001001100010110011101100011011010011100100011101100

Example of resource: <https://www.tools4noobs.com/online_tools/encrypt/>

**Homework for section#5A (approximate due date Sept 14th, 2020)**

1-Arithmetic in extended Galois field with ;  ***= + + + + 1***

**Compute the following elements**:

1/21; 1/9f; 1/ab; 1/cd

2-Arithmetic in extended Galois Field with *;*  ***= + + 1***

Represented by polynomials:  *= + +*

Elements: *(****0****,* ***1****, ,* ***+1****, ,* ***+1****,* ***+*** *,* ***+ +1****)*

Find the 8 inverses:

**Compute the following:**

**+1 ⨸ + =?**

**+ ⨸ +1 =?**

**Optional:**

**+ ⊗ + =?**

**+ +1⊗+ =**?

**Homework for section#5B (approximate due date: Sept 28th, 2020)**

**Part 1 Encryption:** Find the stream {C0 C1 C2 C3} from the initial steam {A0 A5 A10 A15}

|  |  |  |  |
| --- | --- | --- | --- |
| Initial | Inverse | After affine | After shift  & Mix column |
|  |  |  |  |
| A0 = 0101 1010 | B’0 =? | B0 =? | C0 =? |
| A5 = 0101 1001 | B’5 =? | B5 =? | C1 =? |
| A10 = 1010 0110 | B’10 =? | B10 =? | C2 =? |
| A15 = 1101 0001 | B’15 =? | B15 =? | C3 =? |

**Part 2 Decryption:** Find the stream {A0 A5 A10 A15} from the initial steam {C0 C1 C2 C3}

|  |  |  |
| --- | --- | --- |
| Initial | After reverse shift & Mix column | After reverse byte substitution |
|  |  |  |
| C0 = 1001 1100 | B0 =? | A0 =? |
| C1 = 1100 0101 | B5 =? | A5 =? |
| C2 = 0111 0001 | B10 =? | A10 =? |
| C3 = 0010 1101 | B15 =? | A15 =? |

**Homework for Section#6A (due date September 28th, 2020)**

**Alice uses the two following random numbers:**

* 1. Random numbers for polarizer (0=+ ; 1=x) are:

10101100 11110110 01000110 11000111 10001110 01011110 00001011 10100110

00011001 11100001 01100001 11100001 10111001 00111010 01111111 11000101

10101110 10001110 11001011 10000100 01000010 10100010 11010110 01001011

01101100 10100100 11100000 10101010 00101011 00110101 11111101 10011110

b. Random numbers for data stream are:

01100111 00110001 10001101 10010011 11101000 01001111 10010011 01001000

00000001 00001011 10100111 01101001 00000101 01100101 10011101 00111000

10000010 11001110 00100000 10110110 10001100 10000011 00010111 11001100

11010000 00001001 10001111 00001011 10001000 01001010 01000100 01011111

**Bob random numbers for polarizer (0=+ ; 1=x) are:**

10100000 00000011 01111000 01010011 10001110 00000010 10000110 11001111

10100011 00101001 11010101 11111000 11001101 00011011 11110010 11111010

00001010 11110010 10100000 11001001 10100011 10010111 11111110 10000100

00100100 10000001 11000011 00100010 10010000 11011110 10100111 10010101

**QUESTION: Find two possible sequences of matching positions to send the following key:**

00110100 00111001 11101101 01110100 10010001 10101101 10111101 00110001

**Homework for section 6B (approximate due date: Oct 5th, 2020)**

**Question 1:**

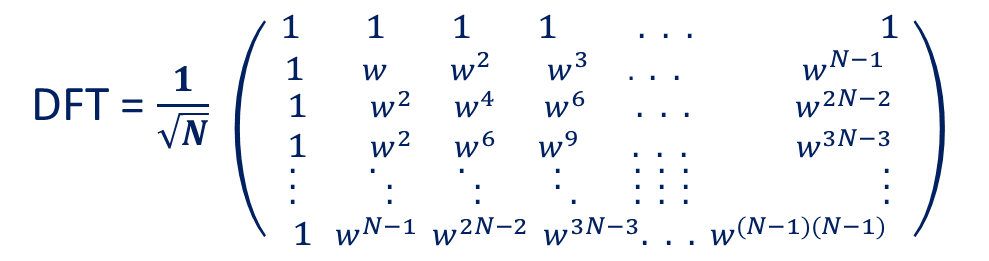
Pick two prime numbers p and q (between 5 and 37), N=pq;

Use Shor algorithm to find p and q from N:

* + 1. Pick a number ***a*** smaller than N
    2. Find the integer **r** verifying  **mod N**
    3. If r odd find a different ***a***
    4. Compute and
    5. If the gcd is not uncovering **p**, and **q**, pick a different ***a***

**Question 2**:

Find the Discrete Fourier Transform (DFT) matrix for N=2, then for N=4:



**=**

**Homework for section 7 (due date tbd)**

**Question 1: Use on-line SHA-2 calculator:**

a) Pick a portion of your resume

b) Hash it with SHA-2

c) Modify one character

d) Hash it with SHA-2

f) Verify the differences in the resulting message digests

**Question 2: Create a blockchain**

1. Segment parts of your resume in 3 blocks
2. Hash the first part: H1
3. Hash (H1 + Part2): H2
4. Hash (H2 + Part 3): H3
5. Modify one character to verify the resulting message digests

**Homework for section 8A - part 1 (due date tbd)**

Use EA to find: gcd(9135,8070) and gcd(11296,8976)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **i** | **ri-1** | **ri** | **qi** | **ri+1** |
| 1 | r0 = | r1 = | q1 = | r2 = |
| 2 | r1 = | r2 = | q2 = | r3 = |
| 3 | r2 = | r3 = | q3 = | r4 = |
| 4 | r3 = | r4 = | q4 = | r5 = |
| 5 | r4 = | r5 = | q5 = | r6 = |
| 6 | r5 = | r6 = | q6 = | r7 = |
| 7 | r6 = | r7 = | q7 = | r8 = |
| 8 | r7 = | r8 = | q8 = | r9 = |
| 9 | r8 = | r9 = | q9 = | r10 = |
| 10 | r9 = | r10 = | q10 = | r11= |

**Homework for section 8A - part 2: (Due date tbd)**

**F**ind **S** and **T** for 11296 and 8976

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **R** | **Q** | **S** | **T** | **EA**  = | **EEA: S**  = – | **EEA: T**  = – |
| **0** | = | - | = **1** | = 0 | R0 = R1 + R2 |  |  |
| **1** | = | = | = **0** | = **1** | R1 = R2 + R3 |  |  |
| **2** | = | = | = | = | R2 = R3 + R4 | S2 = S0 – S1 | T2 = T0 – T1 |
| **3** | = | = | = | = | R3 = R4 + R5 | S3 = S1 – S2 | T3 = T1 – T2 |
| **4** | = | = | = | = | R4 = R5 + R6 | S4 = S2 – S3 | T4 = T2 – T3 |
| **5** | = | = | = | = | R5 = R6 + R7 | S5 = S3 – S4 | T5 = T3 – T4 |
| **6** | = | = | = | = | R6 = R7 + R8 | S6 = S4 – S5 | T6 = T4 – T5 |
| **7** | = | = | = | = | R7 = R8 + R9 | S7 = S5 – S6 | T7 = T5 – T6 |
| **8** | = | = | = | = | R8 = R9 + R10 | S8 = S6 – S7 | T8 = T6 – T7 |

**Homework for section 8B (due date tbd)**

Fast Exponentiation 2273

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Homework – for section 9A**

Calculate several points on the circular group by using the formula of point addition/doubling.

(without using the table presented in class)

***≡ 2 +2 mod 17***

Primitive element: **P = (5, 1)**

Point doubling 1: **(9, 1)** + **(9, 1)**

Point doubling 2: **(0, 6)** + **(0, 6)**

Point addition 1: **(10, 11)** + **(16, 13)**

Point addition 2: **(6,14)**+ **(5, 16)**

**Homework – for section 9B**

Based on the circular group G**ℱ**

***≡++1 mod P(x)***

Irreducible ***P()*= + +1**

Generator ***g = (0010)***

Primitive element: P = (, )

Verify several points:

Point doubling 1: (, ) + (, )

Point doubling 2: (, ) + (, )

Point addition 1: (, ) + (, )

Point addition 2: (, ) + (, )

**Homework for section 10-A**

(this is the last HW for INF 638 2020)

Find an example of DSA signature, and verification using Elliptic Curves

* The elliptic curve is ≡++2 mod 17
* ***q***=19
* The primitive element is: **A** = (, )
* The private key is **d** = 5
* The hash of the message to sign is: = 8

**Method:**

* + Find Public Key **B**
  + Pick ephemeral key ***KE***
  + Find ***r***
  + Compute ***S***
* ***Send (r, s)***
  + Compute: *w ≡ mod q*
  + Compute: *≡ w. mod q*
  + Compute: *≡ w. r mod q*
  + Compute: ***P****= .****A*** *+* ***B***🡺

**Practice for section 10-B**

(no need to submit this as HW)

Find an example of DSA signature, and verification using Elliptic Curves with GF24

* The elliptic curve is ≡++1 mod P(x)
* P()= + +1
* The generator is: **g** = (0010) ***q***=16
* The primitive element is: **A** = (, )
* The private key is **d** = 5
* The hash of the message to sign is: = 9

**Method:**

* + Find Public Key **B**
  + Pick ephemeral key ***KE***
  + Find ***r***
  + Compute ***s***
* ***Send (r,s)***
  + Compute: *w ≡ mod q*
  + Compute: *≡ w. mod q*
  + Compute: *≡ w. r mod q*
  + Compute: ***P****= .****A*** *+* ***B***🡺