1.Perform an Experiment for port scanning with nmap

Nmap is a network scanner utility used for port mapping ,host discovery and vulnerability scanning. Most of its functions are based on using IP packet analysis to detect and identify remote hosts,operating systems and services

Step 1: Port scan for port 21

Command: nmap -p 21scanme.org

```
C:\Users\allek>  nmap -p 21 scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:13 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.100s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT   STATE  SERVICE
21/tcp closed ftp

Nmap done: 1 IP address (1 host up) scanned in 1.90 seconds
```

Step 2:port scan port range

Command: $ nmap -p 21-100 scanme.org

```
C:\Users\allek>nmap -p 21-100 scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:19 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.28s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 77 closed tcp ports (reset)
PORT   STATE    SERVICE
22/tcp open     ssh
25/tcp filtered smtp
80/tcp open     http

Nmap done: 1 IP address (1 host up) scanned in 4.55 seconds
```

step 3: Port scan for multiple TCP and UDP ports

Command: $ nmap -p U:53, T:21-25,80 scanme.org

```
C:\Users\allek>nmap -p U:53, T:21-25,80 scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:25 India Standard Time
WARNING: Your ports include "U:" but you haven't specified UDP scan with -sU.
WARNING: a TCP scan type was requested, but no tcp ports were specified.  Skipping this scan type.
Failed to resolve "T:21-25,80".
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.043s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds
```

step 4: Port scan for all ports

Command: $ nmap -p- example.com

```
C:\Users\allek>nmap -p- example.com
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:27 India Standard Time
Stats: 0:03:42 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 89.09% done; ETC: 22:31 (0:00:27 remaining)
Stats: 0:03:43 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 89.37% done; ETC: 22:31 (0:00:26 remaining)
Stats: 0:03:44 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 89.62% done; ETC: 22:31 (0:00:26 remaining)
Stats: 0:03:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 91.45% done; ETC: 22:31 (0:00:21 remaining)
```

Step 5: Port scan for service name

Command: # nmap -p http, https scanme.org

```
C:\Users\allek>nmap -p http, https scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:32 India Standard Time
Failed to resolve "https".
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.093s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT     STATE  SERVICE
80/tcp   open   http
8008/tcp closed http

Nmap done: 1 IP address (1 host up) scanned in 3.58 seconds
```

Step 6: Fast port scan (100)

Command: $ nmap -F scanme.org

```
C:\Users\allek>nmap -F scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:33 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.31s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 94 closed tcp ports (reset)
PORT    STATE    SERVICE
22/tcp  open     ssh
25/tcp  filtered smtp
80/tcp  open     http
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 4.92 seconds
```

SCAN TECHNIQUES:

Step 1: TCP SYN port scan

Command: $ nmap -sS scanme.org

```
C:\Users\allek>nmap -sS scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:35 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.33s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 992 closed tcp ports (reset)
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
9929/tcp  open     nping-echo
31337/tcp open     Elite
```

Step 2: TCP Connect port scan (without root privileges)

Command: $ nmap -sT scanme.org

```
C:\Users\allek>nmap -sT scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:36 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.035s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
All 1000 scanned ports on scanme.org (45.33.32.156) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 38.57 seconds
```

Step 3: TCP ACK port scan

Command: $ nmap -sA scanme.org

```
C:\Users\allek>nmap -sA scanme.org
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:39 India Standard Time
Nmap scan report for scanme.org (45.33.32.156)
Host is up (0.043s latency).
Other addresses for scanme.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 996 unfiltered tcp ports (reset)
PORT    STATE    SERVICE
25/tcp  filtered smtp
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds
```

Step 4: TCP window port scan

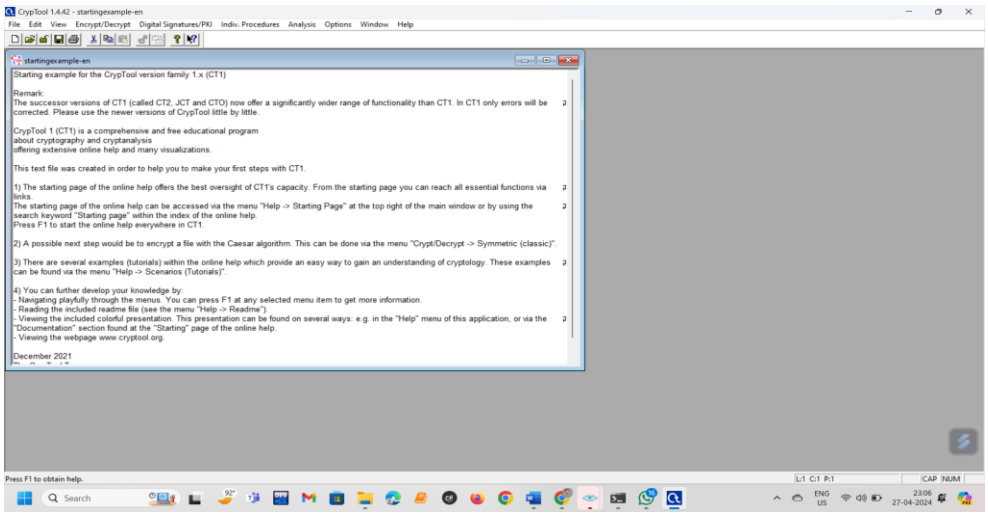
Command: $ namp -w scanme.org

```
C:\Users\allek>nmap -w scanme.orG
Starting Nmap 7.95 ( https://nmap.org ) at 2024-04-27 22:40 India Standard Time
Stats: 0:00:41 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 98.90% done; ETC: 22:41 (0:00:00 remaining)
Nmap scan report for scanme.orG (45.33.32.156)
Host is up (0.37s latency).
Other addresses for scanme.orG (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 992 closed tcp ports (reset)
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 44.10 seconds
```

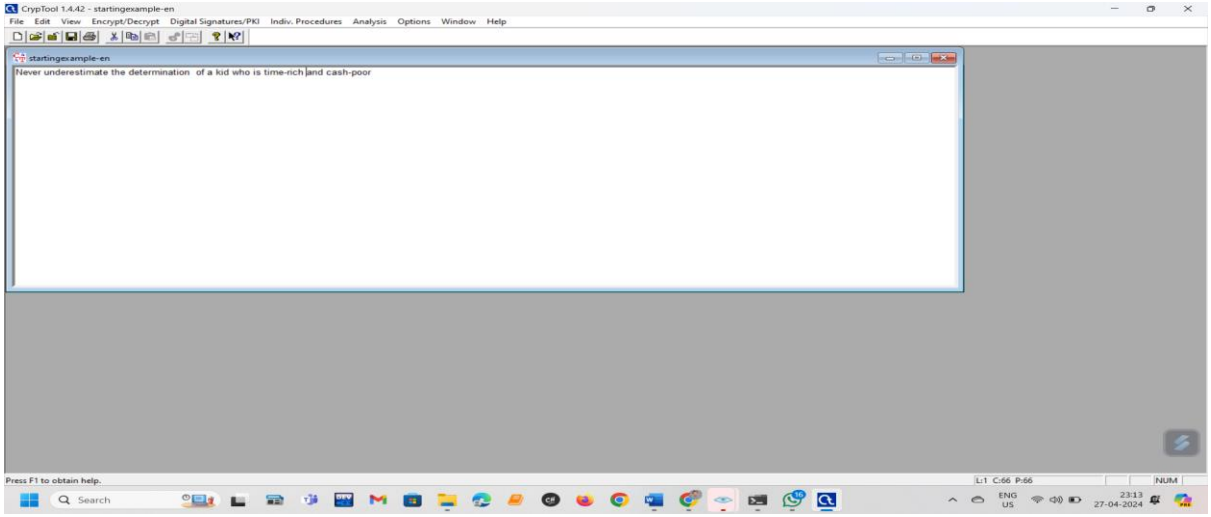
2. Instal a jcrpt tool(or any other eqvivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures studied in theory Network security and management

Step 1: Download and install Cryptool. Download it from https://www.cryptool.org/en/ct1

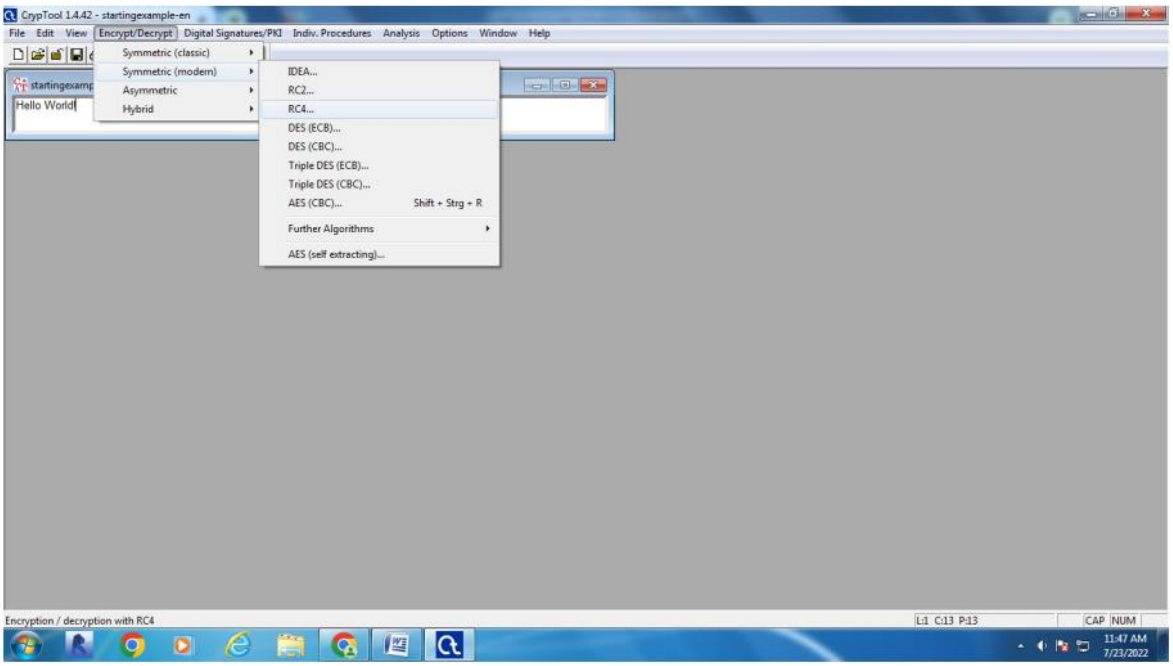
Step 2: Open Cryptool and replace the text Encrypt the following text. Never underetimate the determination of a kid who is time-rich and cash-poor Encryption key: 00 00 00



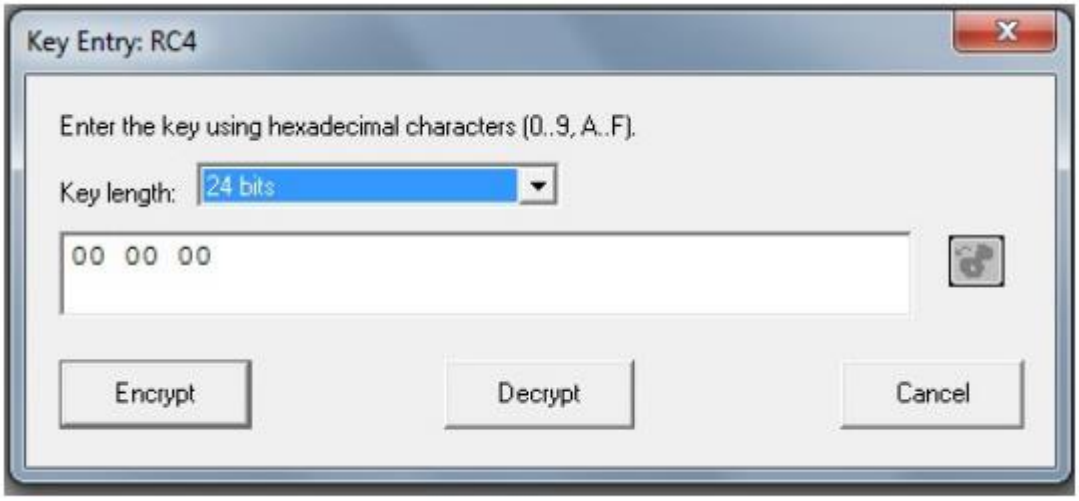
Replace the text with Never underestimate the determintion of a kid who is time-rich and cash-poor



Step 3: Encrypt the text Click on Encrypt/Decrypt button--> Symmetric (modern) -->RC4



Next, the following window will appear

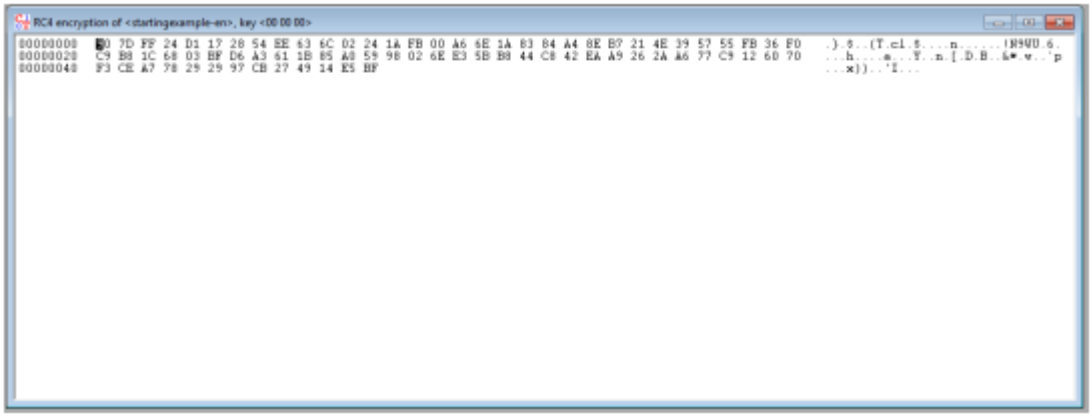


Step 4: Select encrytion key • Select 24 bits as the encrypti on key

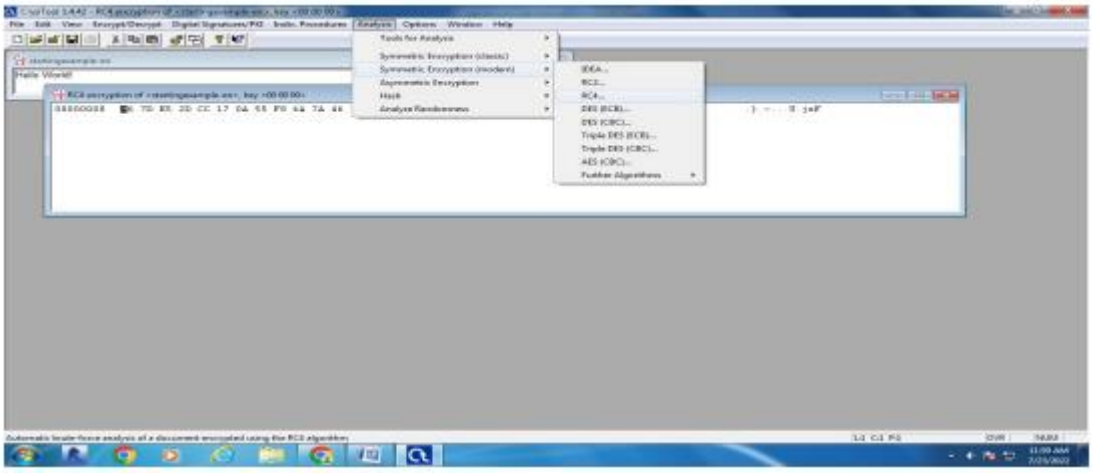
• Set the value to 00 00 00

• Click on Encrypt button

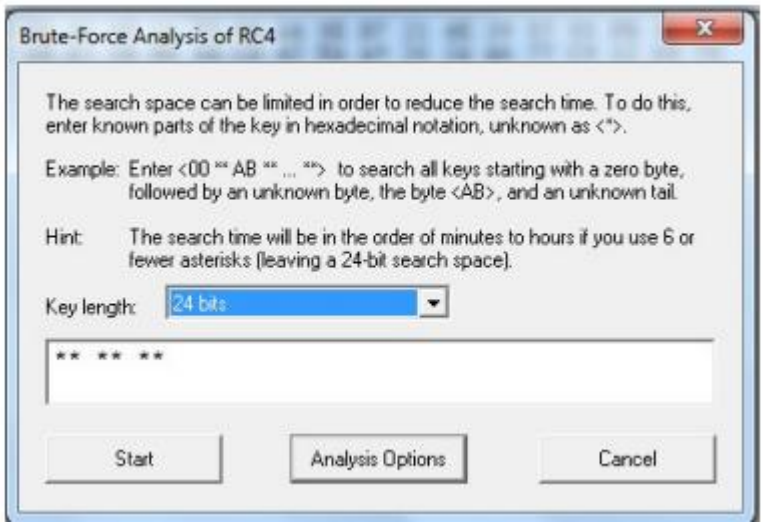
• We will get the following stream cipher



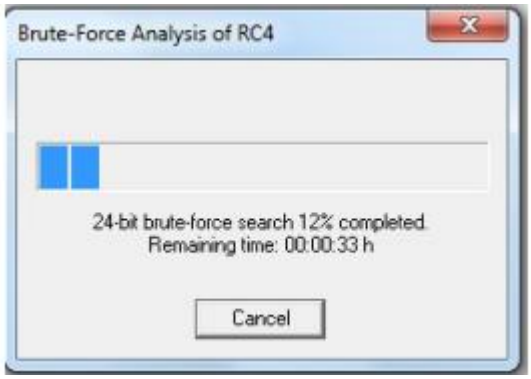
Next, attack the stream cipher Step 5: Click on Analysis menu -->Symmetric Encryption (Modern) --> RC4



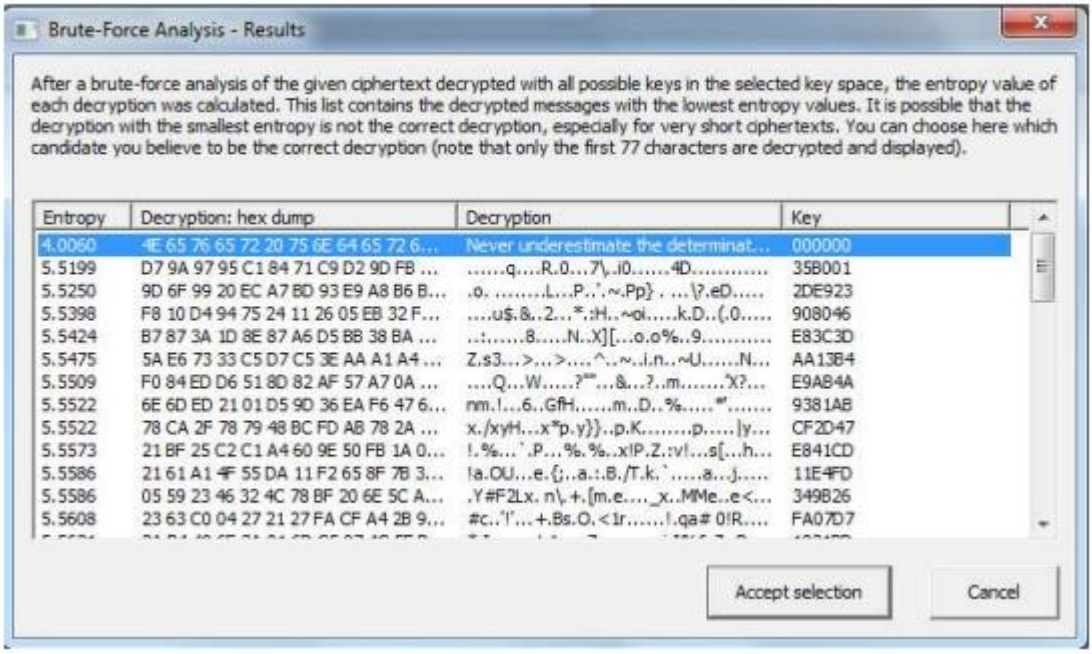
We will get the following window



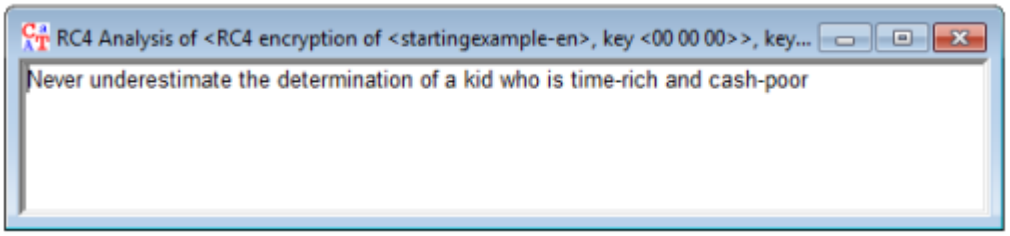
Remember the assumption made is the secret key is 24 bits. So make sure we select 24 bits as the key length. • Click on the Start button. We will get the following window



Note: the time taken to complete the Brute-Force Analysis attack depends on the processing capacity of the machine been used and the key length. The longer the key length, the longer it takes to complete the attack. Step 6: Analyze the results • When the analysis is complete, We will get the following results.



• Note: A lower Entropy number means it is the most likely correct result. It is possible a higher than the lowest found Entropy value could be the correct result. • Select the line that makes the most sense then click on Accept selection button when done. Step 7: When we click on Accept select button it shows the decrypted text / cracked text.



3. Write a program to perform encryption and decryption using the following substitution ciphers.

4. Caeser cipher

```
#include<bits/stdc++.h>
#define MAX_LENGTH 100
void encrypt(char s[], int k)
{
   int i = 0;
   for (i = 0; s[i] != '\0'; i++)
   {
      if (s[i] >= 'A' && s[i] <= 'Z')
      {
         s[i] = ((s[i] + k - 'A') % 26) + 'A';
      }
      else if (s[i] >= 'a' && s[i] <= 'z')
      {
         s[i] = ((s[i] + k - 'a') % 26) + 'a';
      }
      else
         s[i] = s[i];
   }}
void decrypt(char s[], int k)
{
   int i = 0;
   for (i = 0; s[i] != '\0'; i++)
   {
      if (s[i] >= 'A' && s[i] <= 'Z')
      {
         s[i] = ((s[i] - k - 'A') % 26) + 'A';
```

```
        }
        else if (s[i] >= 'a' && s[i] <= 'z')
        {
            s[i] = ((s[i] - k - 'a') % 26) + 'a';
        }
        else
            s[i] = s[i];
    }}
int main()
{
    char s[MAX_LENGTH];
    cout << "Enter the original message :\n";
    cin >> s;
    int k;
    cout << "Enter the value of k :\n";
    cin >> k;
    encrypt(s, k);
    cout << "------Encryption-----\n";
    cout << "The Cipher message is : ";
    cout << s << "\n";
    decrypt(s, k);
    cout << "------Decryption-----\n";
    cout << "The original message is : ";
    cout << s; }
```

Output:

```
Enter the orignal message :
ABCDEFJHJHKNabcdefghhjik
Enter the value of k :
2
------Encryption-----
The Chiper message is : CDEFGHLJLJMPcdefghijjlkm
------Decryption-----
The orignal message is : ABCDEFJHJHKNabcdefghhjik
--------------------------------
```

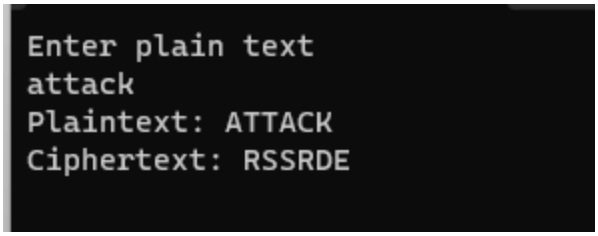
5. Play fair cipher

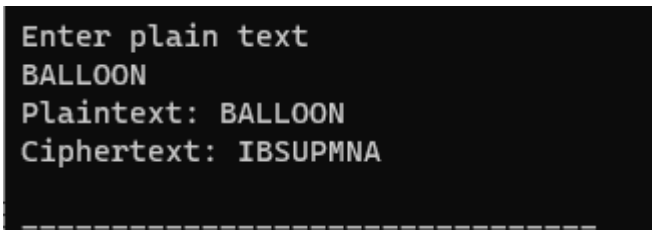
```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;
string preparePlaintext(string plaintext) {
   string result = "";
   for (int i = 0; i < plaintext.length(); ++i) {
      char c = plaintext[i];
      if (isalpha(c))
         result += toupper(c);
   }
   return result;
}
string generateKeyTable(string key) {
   string keyTable = "";
   bool used[26] = {false};

   for (int i = 0; i < key.length(); ++i) {
      char c = key[i];
      if (c == 'J')
         c = 'I';
      if (!used[c - 'A']) {
         keyTable += c;
         used[c - 'A'] = true;
      }
   }
   for (char c = 'A'; c <= 'Z'; ++c) {
      if (c != 'J' && !used[c - 'A']) {
```

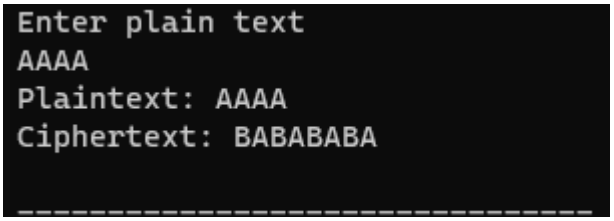
```
            keyTable += c;
            used[c - 'A'] = true;
        }
    }
    return keyTable;
}
string encrypt(string plaintext, string keyTable) {
    string ciphertext = "";
    for (int i = 0; i < plaintext.length(); i += 2) {
        char a = plaintext[i];
        char b = (i + 1 < plaintext.length()) ? plaintext[i + 1] : 'X';



        if (a == b) {
            b = 'X';
            i--;
        }
        int row1, col1, row2, col2;
        row1 = keyTable.find(a) / 5;
        col1 = keyTable.find(a) % 5;
        row2 = keyTable.find(b) / 5;
        col2 = keyTable.find(b) % 5;
        if (row1 == row2) {
            ciphertext += keyTable[row1 * 5 + (col1 + 1) % 5];
            ciphertext += keyTable[row2 * 5 + (col2 + 1) % 5];
        } else if (col1 == col2) {
            ciphertext += keyTable[((row1 + 1) % 5) * 5 + col1];
            ciphertext += keyTable[((row2 + 1) % 5) * 5 + col2];
        } else {
            ciphertext += keyTable[row1 * 5 + col2];
            ciphertext += keyTable[row2 * 5 + col1];
```

```
        }
    }
    return ciphertext;
}
int main() {
    string key = "MONARCHY";
    string plaintext;
    cout<<"Enter plain text \n";
    cin>>plaintext;
    plaintext = preparePlaintext(plaintext);
    string keyTable = generateKeyTable(key);
    string ciphertext = encrypt(plaintext, keyTable);
    cout << "Plaintext: " << plaintext << endl;
    cout << "Ciphertext: " << ciphertext << endl;
    return 0;
}
```

Output:







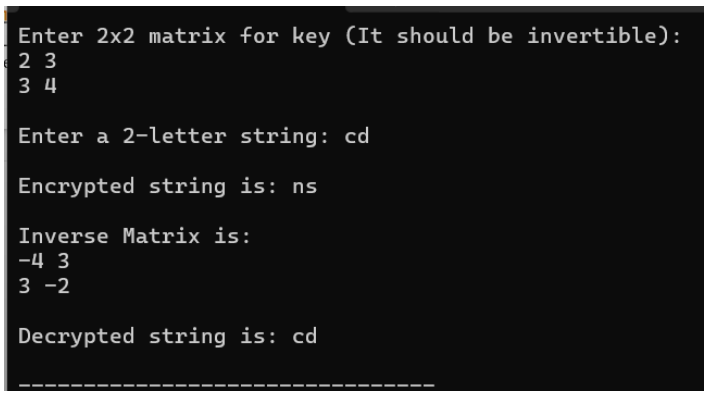
6. Hill Cipher

```
#include<iostream>
#include<string>
using namespace std;
float encrypt[2][1], decrypt[2][1], a[2][2], b[2][2], mes[2][1], c[2][2];
void encryption(); //encrypts the message
void decryption(); //decrypts the message
void getKeyMessage(); //gets key and message from user
void inverse(); //finds inverse of key matrix
int main() {
   getKeyMessage();
   encryption();
   decryption();
}
void encryption() {
   int i, j, k;
   for(i = 0; i < 2; i++)
      for(j = 0; j < 1; j++)
         for(k = 0; k < 2; k++)
            encrypt[i][j] = encrypt[i][j] + a[i][k] * mes[k][j];
   cout<<"\nEncrypted string is: ";
   for(i = 0; i < 2; i++)
      cout<<(char)(((int)encrypt[i][0] % 26) + 97);
}
void decryption() {
   int i, j, k;
   inverse();
   for(i = 0; i < 2; i++)
      for(j = 0; j < 1; j++)
```

```
            for(k = 0; k < 2; k++)
                decrypt[i][j] = decrypt[i][j] + b[i][k] * encrypt[k][j];
    cout<<"\nDecrypted string is: ";
    for(i = 0; i < 2; i++)
        cout<<(char)(((int)decrypt[i][0] % 26) + 97);
    cout<<"\n";
}
void getKeyMessage() {
    int i, j;
    string msg;
    cout<<"Enter 2x2 matrix for key (It should be invertible):\n";
    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++) {
            cin>>a[i][j];
            c[i][j] = a[i][j];
        }
    cout<<"\nEnter a 2-letter string: ";
    cin>>msg;
    for(i = 0; i < 2; i++)
        mes[i][0] = msg[i] - 97;
}
void inverse() {
    int i, j, k;
    float p, q;
    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++) {
            if(i == j)
                b[i][j] = 1;
            else
                b[i][j] = 0;
```

```
    }
  for(k = 0; k < 2; k++) {
    for(i = 0; i < 2; i++) {
      p = c[i][k];
      q = c[k][k];
      for(j = 0; j < 2; j++) {
        if(i != k) {
          c[i][j] = c[i][j] * q - p * c[k][j];
          b[i][j] = b[i][j] * q - p * b[k][j];
        }}}}
  for(i = 0; i < 2; i++)
    for(j = 0; j < 2; j++)
      b[i][j] = b[i][j] / c[i][i];
  cout<<"\n\nInverse Matrix is:\n";
  for(i = 0; i < 2; i++) {
    for(j = 0; j < 2; j++)
      cout<<b[i][j]<<" ";
    cout<<"\n";
  }
}
```

Output:



7. Write a program to implement the DES algorithm.

```
#include<iostream>
#include<string>
#include<bits/stdc++.h>
using namespace std;

string sbox(string s,int table[4][4])
{
map<string,int> mp;
mp["00"]=0;
mp["10"]=2;
mp["01"]=1;
mp["11"]=3;
string a="";
a+=s[0];
a+=s[3];
string b="";
b+=s[1];
b+=s[2];
int row=mp[a];
int col=mp[b];
vector<string> v;
v.push_back("00");
v.push_back("01");
v.push_back("10");
v.push_back("11");
return v[table[row][col]];
```

```
}

string fk(string l,string r,string k)
{
int e[]={4,1,2,3,2,3,4,1};
string eout="";
for(int i=0;i<8;i++)
{
char a=r[e[i]-1];
if(a==k[i]) eout+="0";
else eout+="1";
}

cout<<"this will be the input going to be feeded to s boxes:"<<eout<<endl;

string frst=eout.substr(0,4);
string sec=eout.substr(4,4);
int table1[][4]={
{1,0,3,2},
      { 3,2,1,0},
     {0,2,1,3},
     {3,1,3,2}
  };

  int table2[][4]={
  {0,1,2,3},
  { 2,0,1,3 },
  {3,0,1,0},
  { 2,1,0,3 }
  };
```

```
frst=sbox(frst,table1);
sec=sbox(sec,table2);

//int p4[]={2,4,3,1};
string p4 ="";
p4+=frst[1];
p4+=sec[1];
p4+=sec[0];
p4+=frst[0];

string fklout="";
for(int i=0;i<4;i++)
{
if(p4[i]==l[i]) fklout+="0";
else fklout+="1";
}
return fklout;

}



string encrypt(string pt, string k1,string k2)
{
int ip[]={2,6,3,1,4,8,5,7};
string ptip="";
for(int i=0;i<8;i++)
   {
   ptip+=pt[ip[i]-1];
}
```

```
cout<<"After IP text will be as:"<<ptip<<endl;

string frst=ptip.substr(0,4);
string sec=ptip.substr(4,4);
frst=fk(frst,sec,k1);
sec=fk(sec,frst,k2);

string s="";
s+=sec;
s+=frst;
int ipinv[]={4,1,3,5,7,2,8,6};
string ct="";
for(int i=0;i<8;i++)
   {
   ct+=s[ipinv[i]-1];
}
return ct;

}


int main()
{
   string pt="10010111";
   string key="1010000010";
   int ipinv[]={4,1,3,5,7,2,8,6};
   int p10[]={3,5,2,7,4,10,1,9,8,6};
   int p8[]={6,3,7,4,8,5,10,9};
   string p10key="";
   for(int i=0;i<10;i++)
```

```
    {
    p10key+=key[p10[i]-1];
}
cout<<"key after p10 is:"<<p10key<<endl;

//left circular shift of 1bit for each part of 5bits
string frst=p10key.substr(0,5);
string sec=p10key.substr(5,5);
p10key.clear();
frst+=frst[0];
sec+=sec[0];
frst.erase(0,1);
sec.erase(0,1);
p10key+=frst;
p10key+=sec;

string p8key1="";
for(int i=0;i<8;i++)
    {
    p8key1+=p10key[p8[i]-1];
}
cout<<"key after frst p8 i.e K1 is:"<<p8key1<<endl;
frst+=frst[0];
frst+=frst[1];
sec+=sec[0];
sec+=sec[1];
frst.erase(0,2);
sec.erase(0,2);
p10key.clear();
p10key+=frst;
```

```
p10key+=sec;

string p8key2="";
for(int i=0;i<8;i++)
   {
   p8key2+=p10key[p8[i]-1];
}
cout<<"key after frst p8 i.e K2 is:"<<p8key2<<endl;

cout<<endl<<endl;
string ct=encrypt(pt,p8key1,p8key2);

cout<<"cipher text is:"<<ct<<endl;

cout<<endl<<endl;

cout<<"plain text after decryption is:"<<encrypt(ct,p8key2,p8key1)<<endl;

return 0;
}
```

```
key after p10 is:1000001100
key after frst p8 i.e K1 is:10100100
key after frst p8 i.e K2 is:01000011


After IP text will be as:01011101
this will be the input going to be feeded to s boxes:01001111
this will be the input going to be feeded to s boxes:00010110
cipher text is:00111000


After IP text will be as:00101010
this will be the input going to be feeded to s boxes:00010110
this will be the input going to be feeded to s boxes:01001111
plain text after decryption is:10010111
```

8. Write a program to implement RSA algorithm.

```
#include <bits/stdc++.h>

using namespace std;

int gcd(int a, int b){
	if(b == 0) return a;
	return gcd(b, a % b);
}

double modInverse(int e, int phiN){
	int d = 0;
	while(d < phiN){
		if((e * d) % phiN == 1)
			return d;
		d++;
	}
	return 1;
}

int main(){
	int p, q;
	cout << "enter p, q prime value " ;
	cin >> p >> q;
	int n = p * q;
	int phiN = (p-1) * (q-1);
	double e = 2;
	while(e < phiN){
		if(gcd(e, phiN) == 1)
			break;
```

```
            else
                e++;
        }
        int d = modInverse(e, phiN);
        int plaintext;
        int cipher = 1;
        cout << "\nEnter plaintext ";
        cin >> plaintext;
        for(int i=0; i<e; i++){
            cipher = (cipher * plaintext) % n;
        }
    cout << "cipher text is " << cipher << endl;

        int decrypt = 1;
        for(int i =0; i<d; i++){
            decrypt = (decrypt * cipher) %n;
        }
        cout << "plain text is " << decrypt << endl;
        return 0;
}
```

```
enter p, q prime value 3 7

Enter plaintext 12
cipher text is 3
plain text is 12
```

9. Calculate the message digest of a text using the SHA-1 algorithm.

```
#include <bits/stdc++.h>
using namespace std;
tydef unsigned long int uint32;
typedef unsigned long long int uint64_t;
uint32 hexCharToInt(char hexChar) {
   return hexChar >= '0' && hexChar <= '9' ? hexChar - '0' : hexChar - 'A' + 10;
}
uint32 hexToBinary(const string &hexString) {
   uint32 binaryValue = 0;
   for (int i=0;hexString[i]!='\0';i++) {
      binaryValue = (binaryValue << 4) | hexCharToInt(hexString[i]);
   }
   return binaryValue;
}
string binaryToHex(uint32 binaryValue) {
   stringstream ss;
   ss << hex << setw(8) << setfill('0') << binaryValue;
   return ss.str();
}
uint32 rotateLeft(uint32 x, uint32 n) {
   return (x << n) | (x >> (32 - n));
}
uint32 f(uint32 t, uint32 b, uint32 c, uint32 d) {
   if (t < 20) return (b & c) | ((~b) & d);
   else if (t < 40) return b ^ c ^ d;
   else if (t < 60) return (b & c) | (b & d) | (c & d);
   else return b ^ c ^ d;
}
```

```
uint32 K(uint32 t) {
    if (t < 20) return 0x5A827999;
    else if (t < 40) return 0x6ED9EBA1;
    else if (t < 60) return 0x8F1BBCDC;
    else return 0xCA62C1D6;
}
void processBlock(uint32 *W, uint32 *H) {
    uint32 a = H[0];
    uint32 b = H[1];
    uint32 c = H[2];
    uint32 d = H[3];
    uint32 e = H[4];
    for (uint32 t = 0; t < 80; t++) {
        if (t >= 16)
            W[t] = rotateLeft(W[t-3] ^ W[t-8] ^ W[t-14] ^ W[t-16], 1);
        uint32 TEMP = rotateLeft(a, 5) + f(t, b, c, d) + e + W[t] + K(t);
        e = d;
        d = c;
        c = rotateLeft(b, 30);
        b = a;
        a = TEMP;
    }
    H[0] += a;
    H[1] += b;
    H[2] += c;
    H[3] += d;
    H[4] += e;
}

int main() {
```

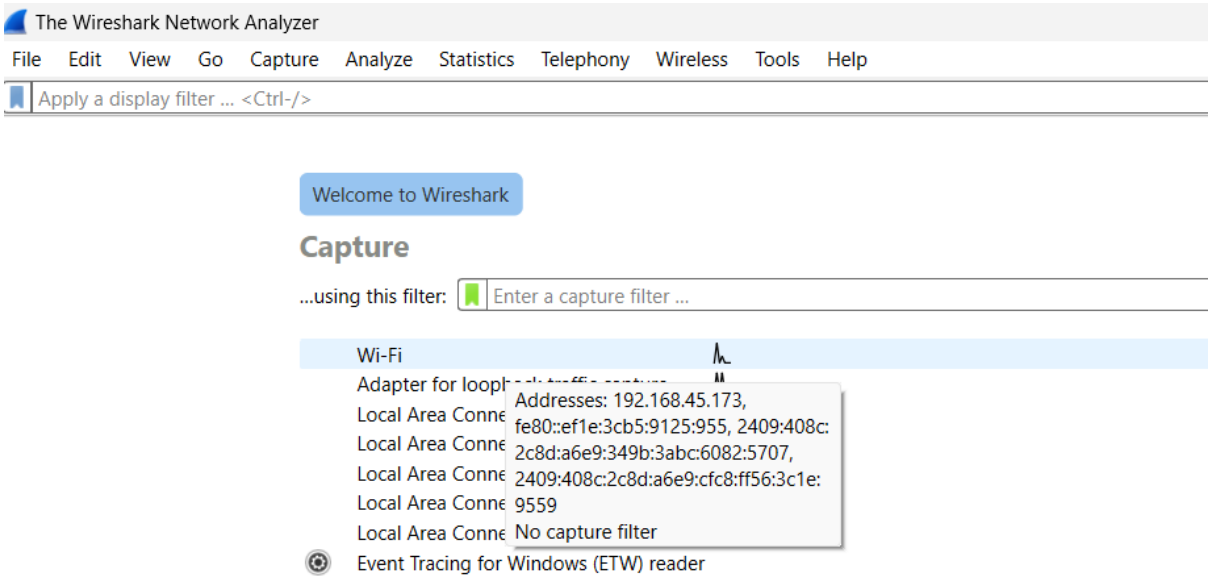
```
    string input;
    cout << "Enter binary string: ";
    cin >> input;
    uint64_t input_length = input.length();
    input += '1';
    while (input.length() % 512 != 448) {
        input += '0';
    }

    string input_length_bin = bitset<64>(input_length).to_string();
    input += input_length_bin;

    uint32 H[5] = {0x67452301, 0xEFCDAB89, 0x98BADCFE, 0x10325476, 0xC3D2E1F0};
    uint32 W[80];
    memset(W, 0, sizeof(W));

    for (size_t i = 0; i < input.length(); i += 512) {
        for (size_t j = 0; j < 16; j++) {
            W[j] = hexToBinary(input.substr(i + j * 32, 32));
        }
        processBlock(W, H);
    }
for (int i = 0; i < 5; i++) {
        cout << binaryToHex(H[i]) << " ";
    }  return 0;
}
```

Output:

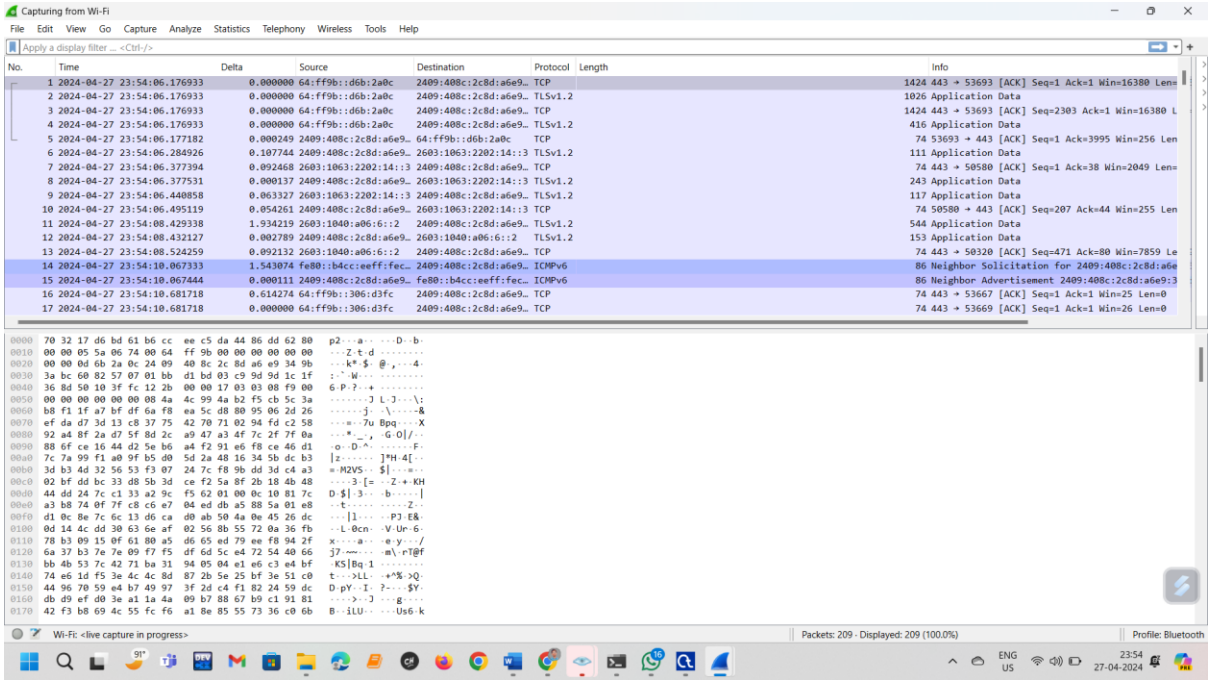
```
Enter binary string: allekarthik
187c53cd dea6deac 15b49a57 f7916102 876084ce
--------------------------------
```

10. Working with sniffers for monitoring network communication (Wireshark).

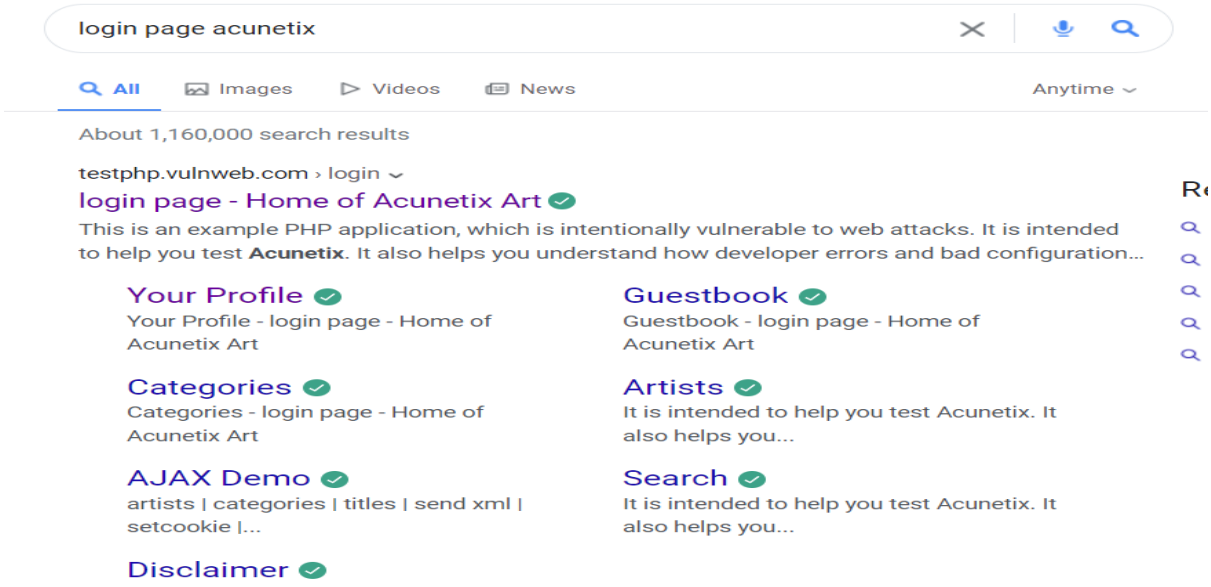
Step 1: Open Wireshark, select wifi Interface and click on it to capture packets.



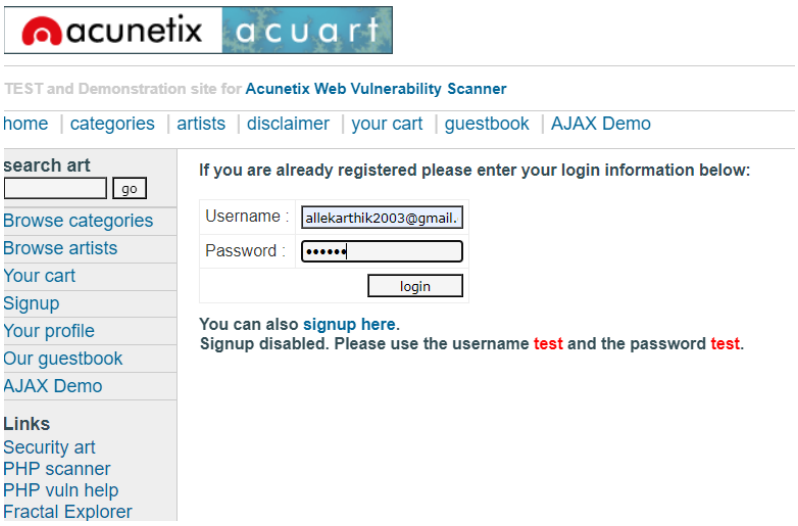
The below are packets under wifi interface .



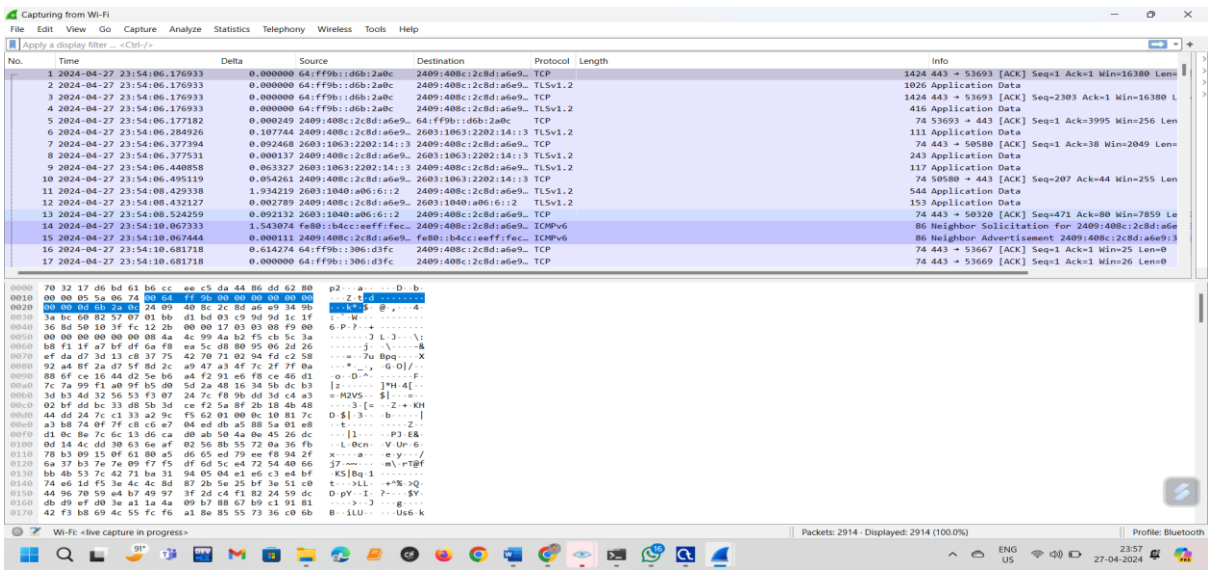
Then go to any browser like chrome then click any login website



Open the first website and enter your login details

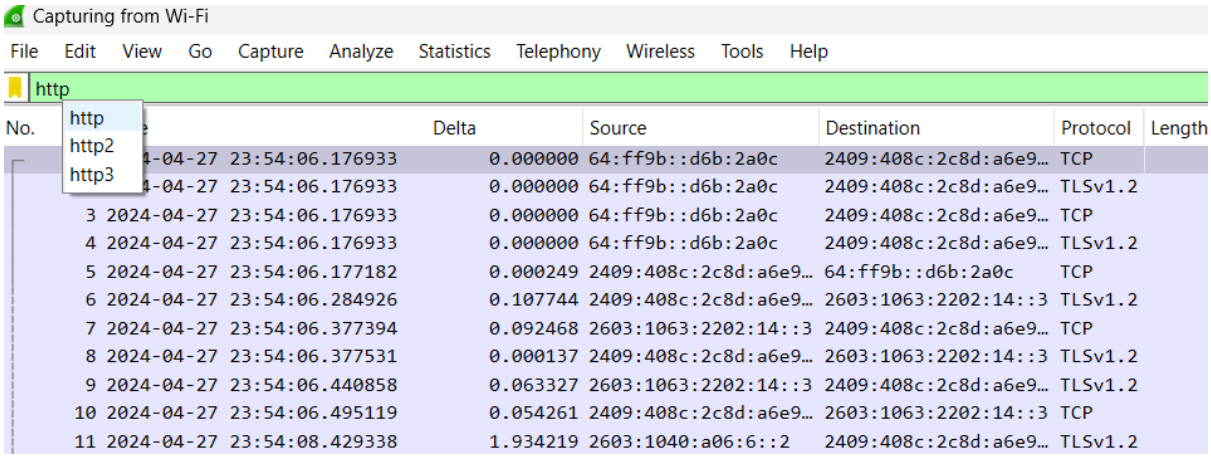


Then open wireshark the packets are approximately above 2000



22015A0507

Click http in filter tab and click enter



It will display the all the http packets