CSE 3501

INFORMATION SECURITY ANALYSIS & AUDIT

Midterm Lab

Group B

L9+L10 | PLBG04

FALL SEMESTER 2021-22

by

SHARADINDU ADHIKARI

19BCE2105

Group B

Question 1

1. **Aim:** To implement the RSA algorithm.

2. **Objective:** After some research and studying RSA briefly, I've decided to go forward with implementing it in C++.

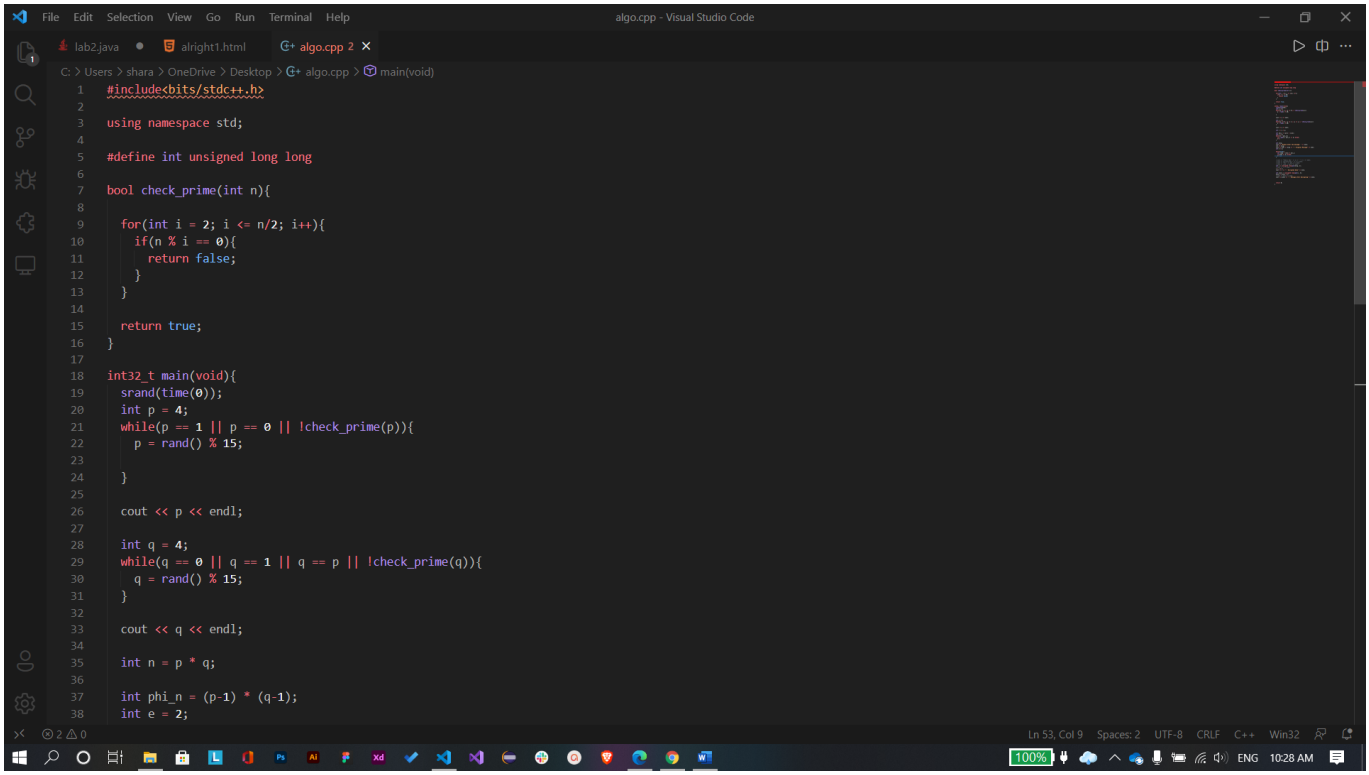
3. **Procedure:**
   We know right, RSA Algorithm is an example for Public Key Encryption algorithm; so here we are supposed to find two keys:
   1) Public Key which is used at encryption, & 2) Private Key which is used at decryption.

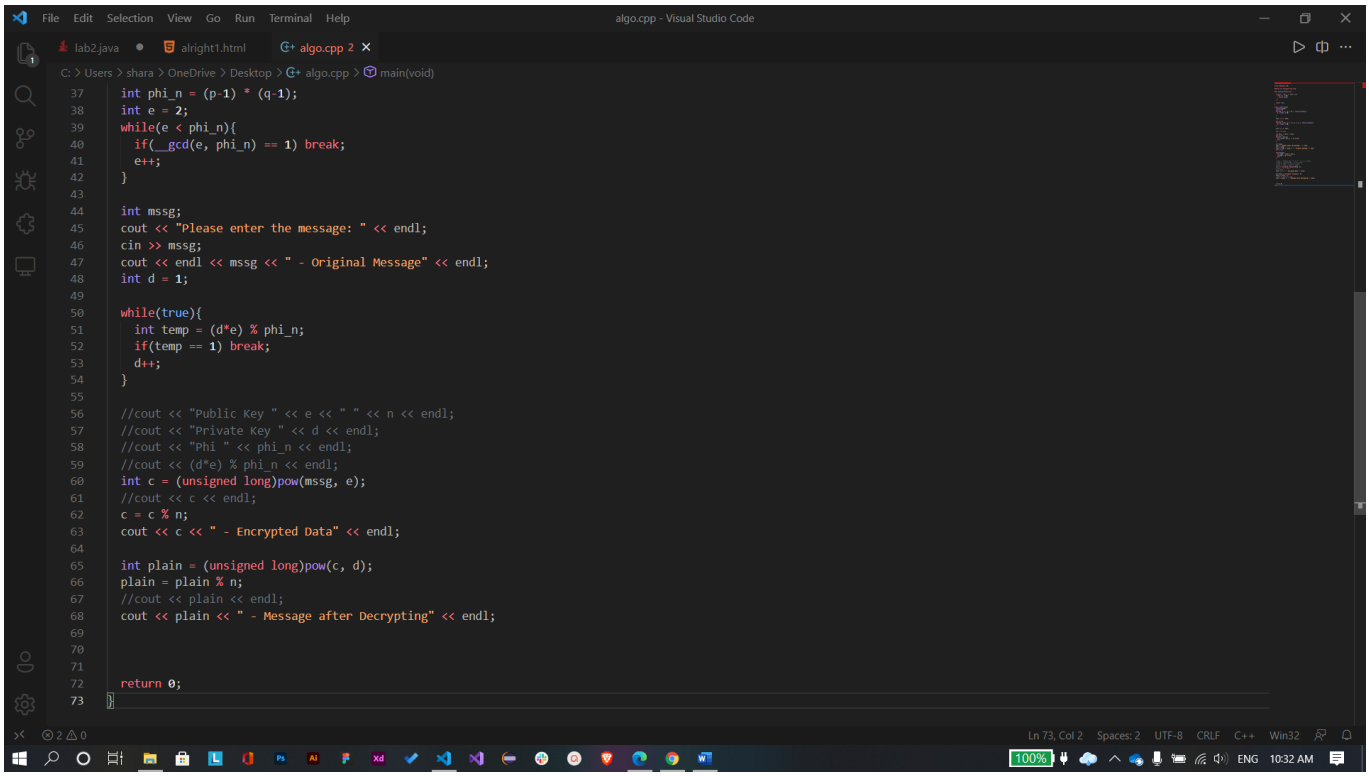
   Step 1: Select two large Primes P, Q
   Step 2: Calculate n=P*Q & O(n) = (P-1)*(Q-1)
   Step 3: Assume e and d (Public and Private Key).
   Step 4: Encrypt the Plain Text using Public Key e.
   Step 5: Decrypt the Cipher Text using Private Key d.

4. **Code:**

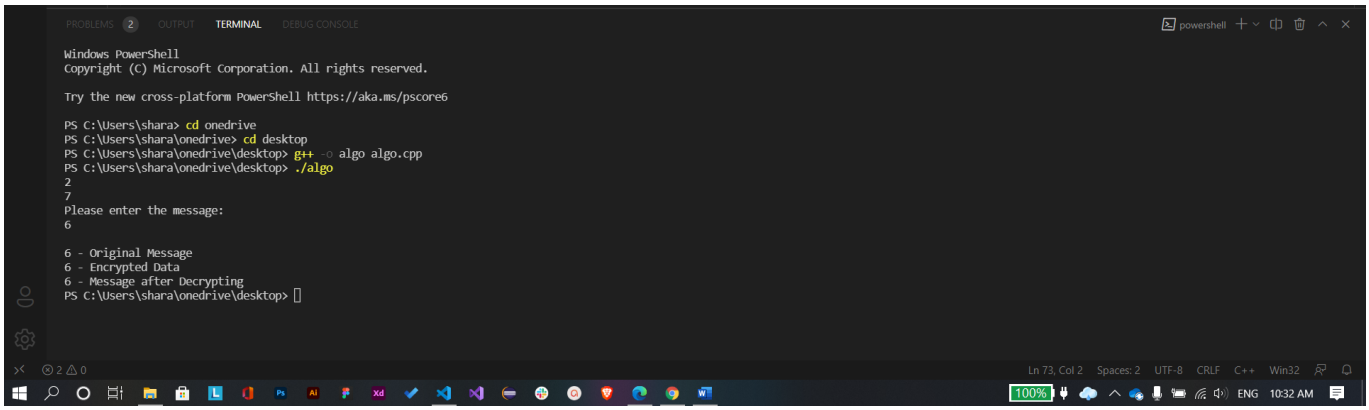
```cpp
#include<bits/stdc++.h>
using namespace std;
#define int unsigned long long

bool check_prime(int n){

  for(int i = 2; i <= n/2; i++){
    if(n % i == 0){
      return false;
    }
  }
  return true;
}

int32_t main(void){
  srand(time(0));
  int p = 4;
  while(p == 1 || p == 0 || !check_prime(p)){
    p = rand() % 15;
  }

  cout << p << endl;
  int q = 4;
  while(q == 0 || q == 1 || q == p || !check_prime(q)){
    q = rand() % 15;
  }

  cout << q << endl;
  int n = p * q;
  int phi_n = (p-1) * (q-1);
  int e = 2;
```

```
    while(e < phi_n){
      if(__gcd(e, phi_n) == 1) break;
      e++;
    }

    int mssg;
    cout << "Please enter the message: " << endl;
    cin >> mssg;
    cout << endl << mssg << " - Original Message" << endl;
    int d = 1;

    while(true){
      int temp = (d*e) % phi_n;
      if(temp == 1) break;
      d++;
    }

    //cout << "Public Key " << e << " " << n << endl;
    //cout << "Private Key " << d << endl;
    //cout << "Phi " << phi_n << endl;
    //cout << (d*e) % phi_n << endl;

  int c = (unsigned long)pow(mssg, e);
    //cout << c << endl;
    c = c % n;
    cout << c << " - Encrypted Data" << endl;

    int plain = (unsigned long)pow(c, d);
    plain = plain % n;
    //cout << plain << endl;
    cout << plain << " - Message after Decrypting" << endl;
    return 0;
  }
```





5. **Results screenshot:**



6. **Conclusion:** While establishing RSA key pairs, usage keys and general-purpose keys are generally integrated. In usage RSA keys, two key pairs are used for encryption and signatures. I've tried to demonstrate the same using C++. Hope you liked it.

Question 2

1. **Aim:** To all UDP related captures using packet Sniffer in Wireshark.

2. **Objective:** To demonstrate Capture and Display filters in Wireshark, while being connected to WiFi.

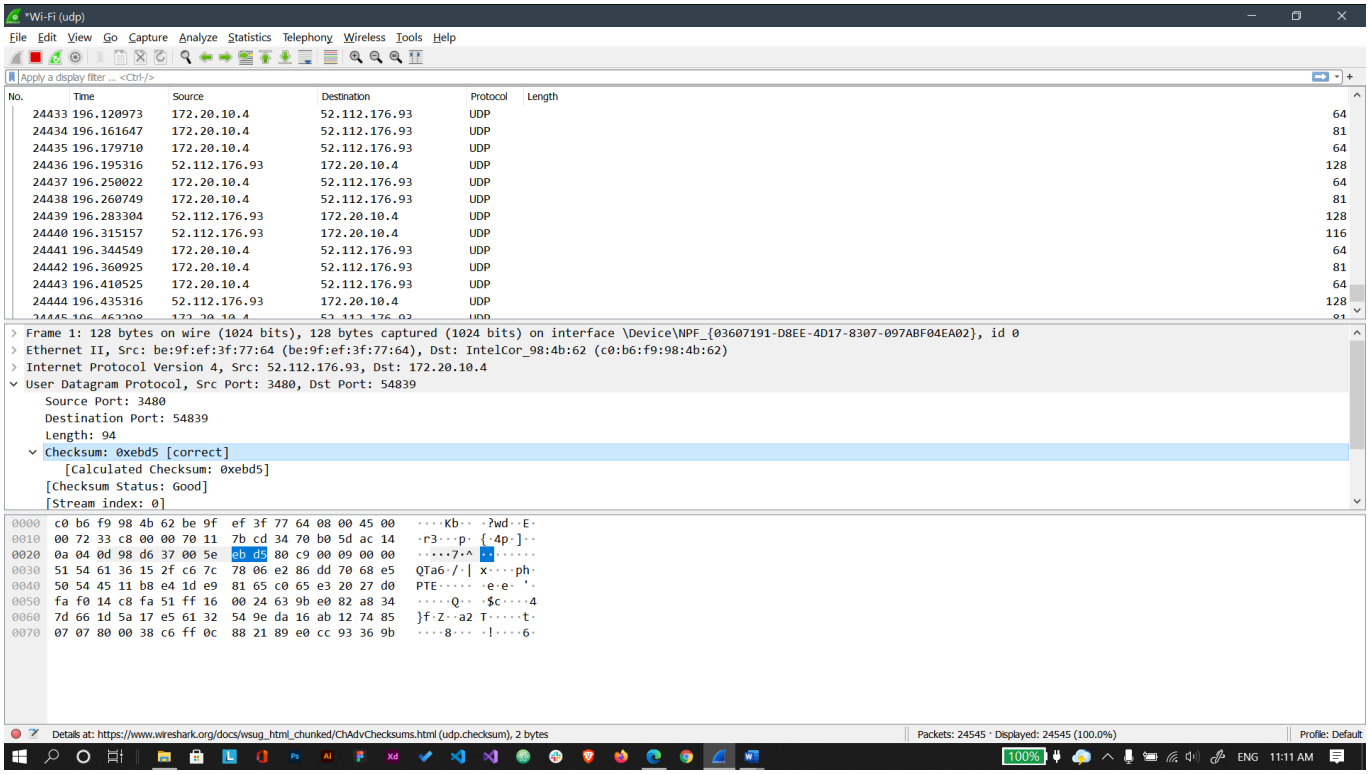
3. **Procedure:**

    1. Start up the Wireshark program (select an interface and press start to capture packets).
    2. Start up our favourite browser (Edge)
    3. In the browser, go to any homepage
    4. After browser has displayed the webpage, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets capture.
    5. Apply all the filters.
    6. Take Implementation screenshots of the all the stuff asked in the question.

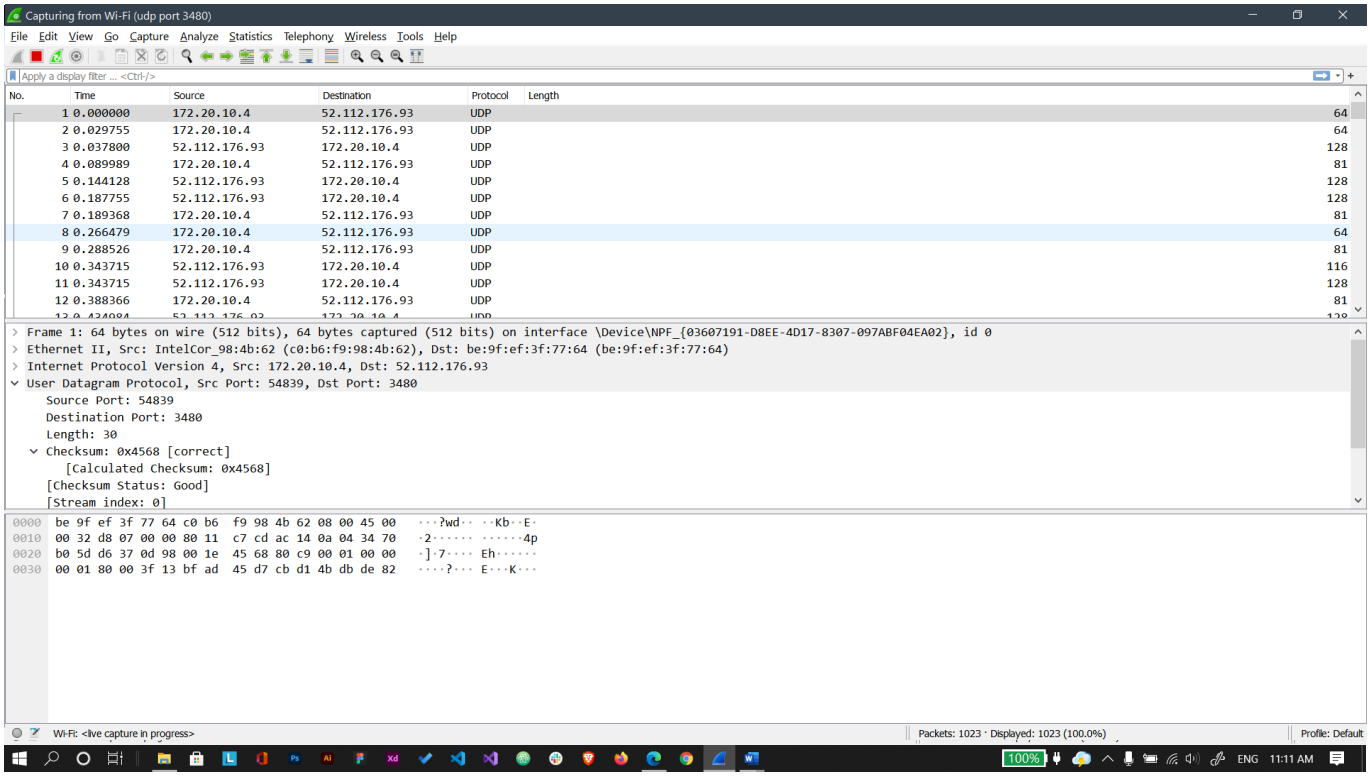
4. **Implementation Screenshots:**

Capture Filter:

UDP:

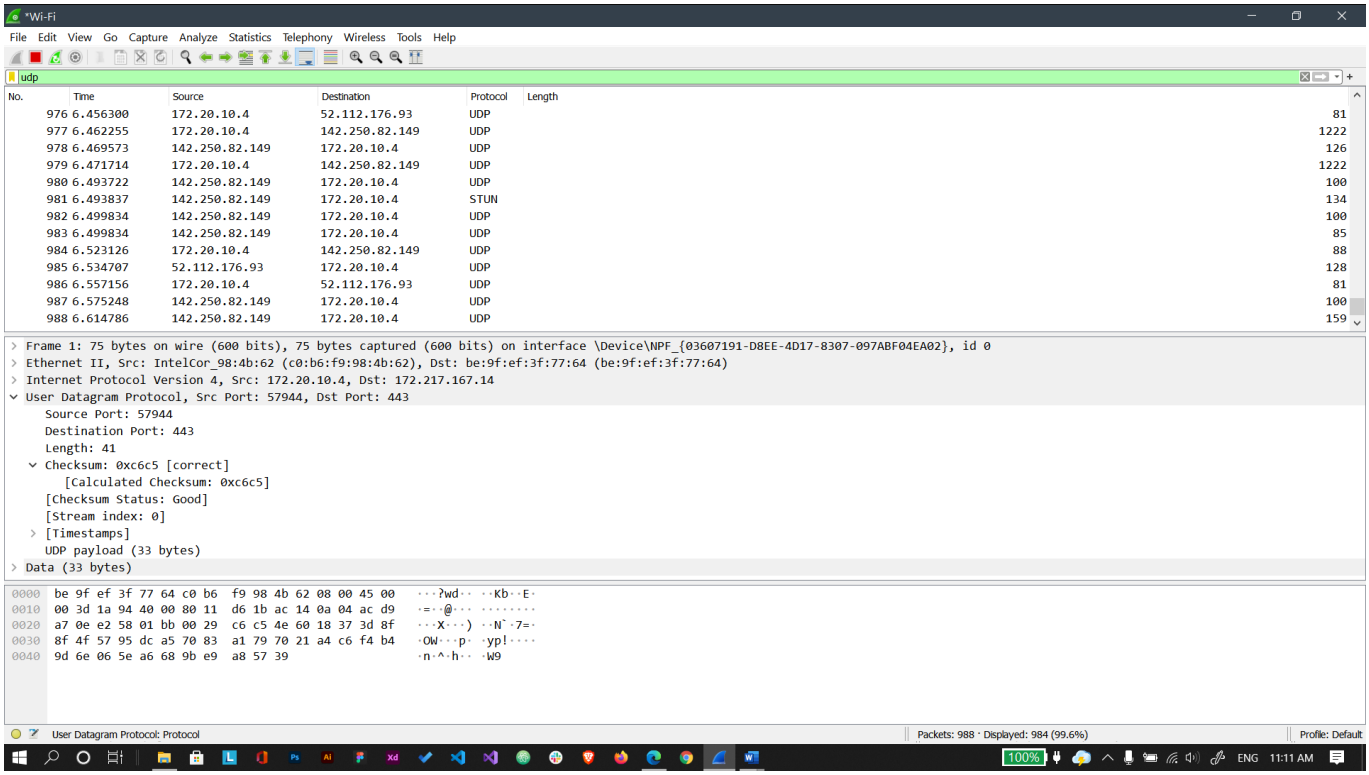


UDP Port 3480:

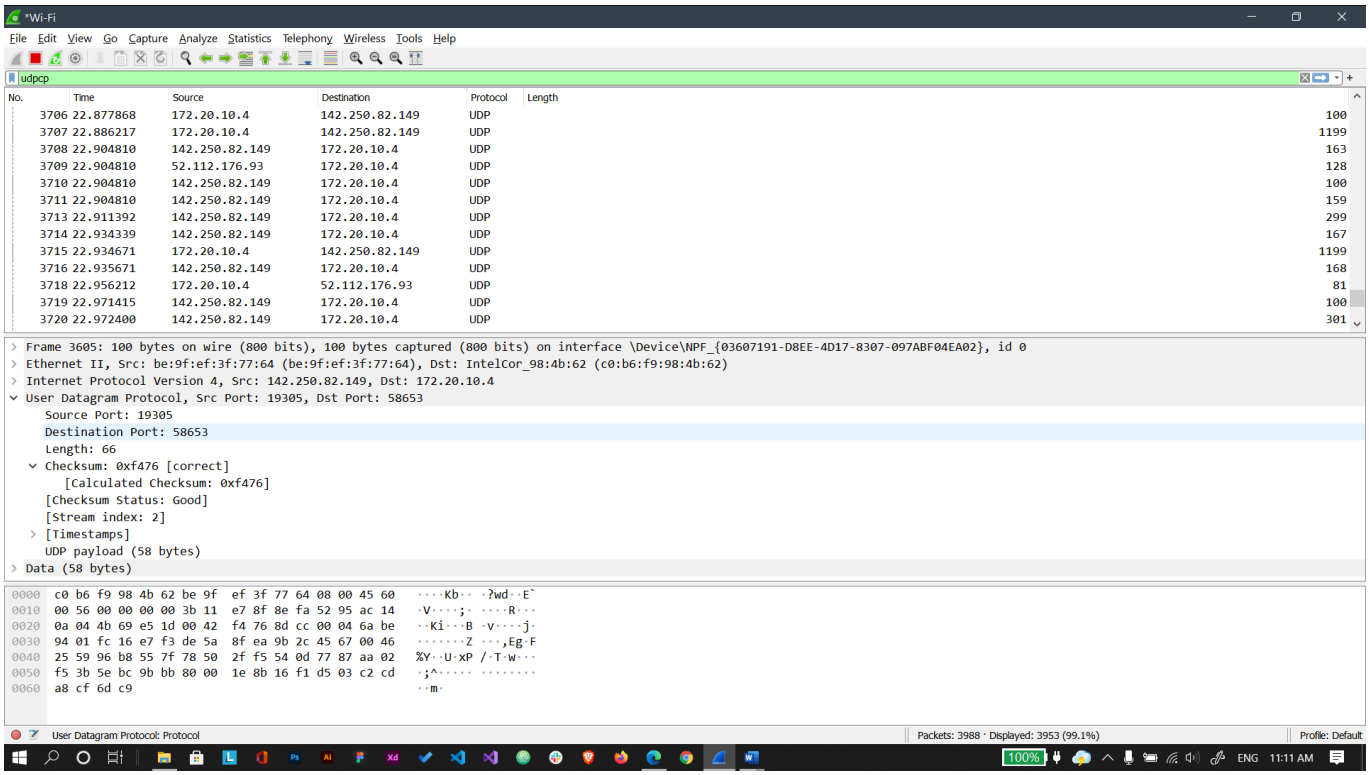


Display filters:

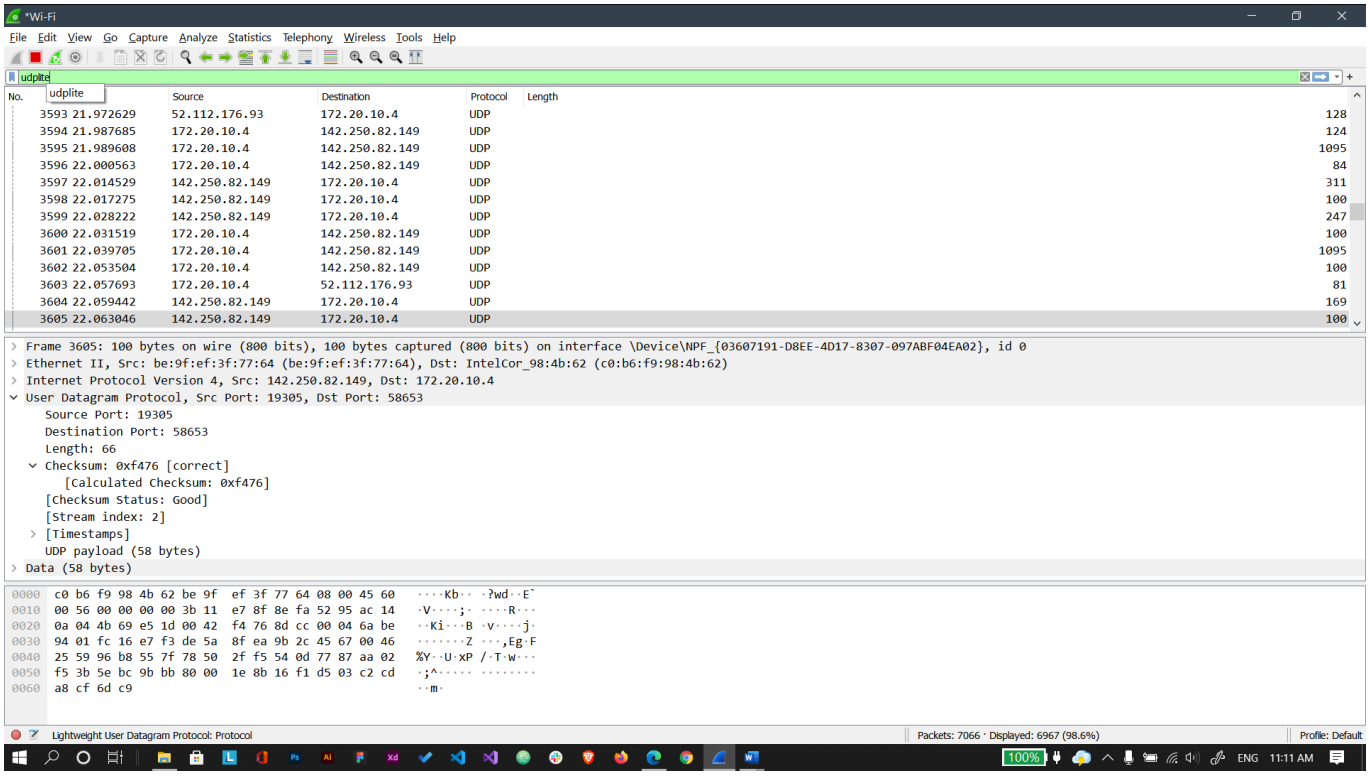
UDP:



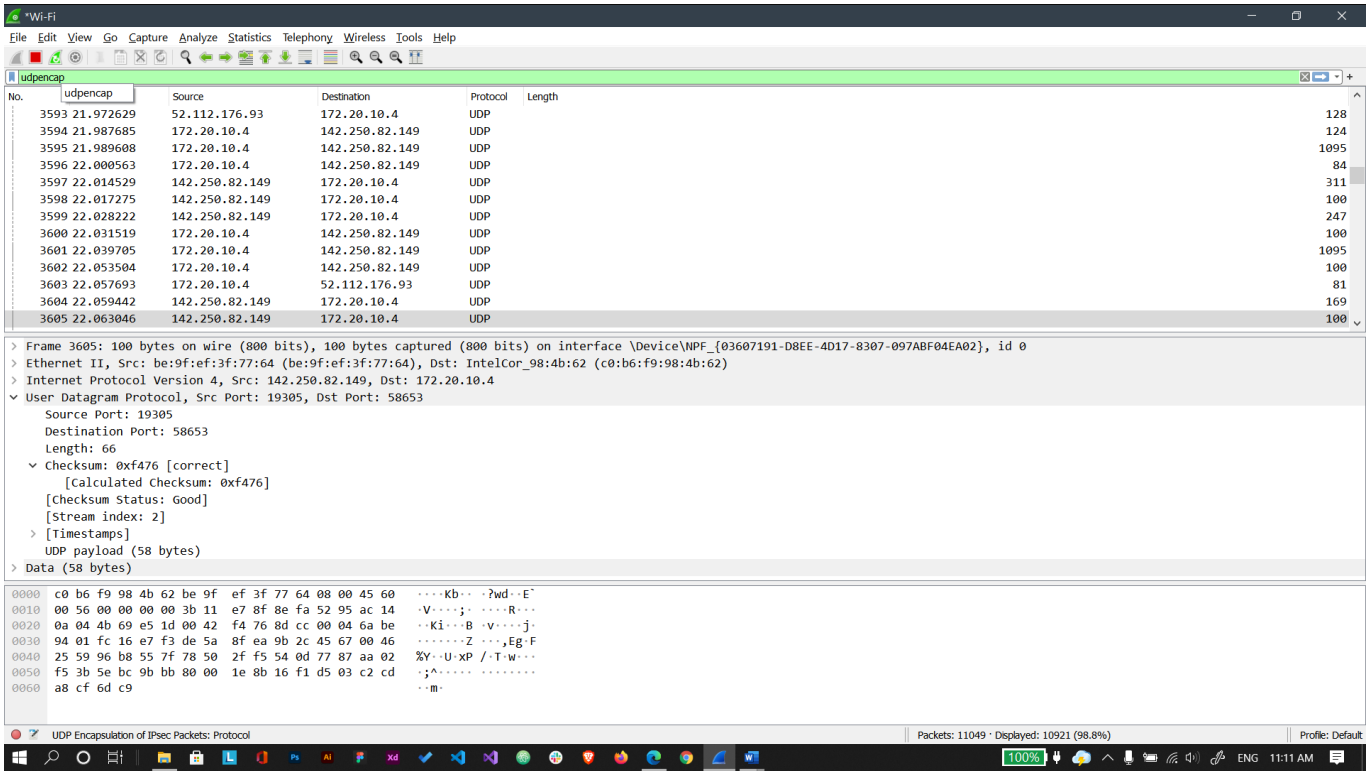
UDPCP:



UDPLite:



UDPencap:



5. **Conclusion:** As I've demonstrated here using Wireshark, UDP packets can be filtered greatly using the various filters available.
