NETWORK SECURITY & CRYPTOGRAPHY

Assignment 2

Submitted by:

Pranav Choudhary

2018184

- ## About Assignment

  Extend the client and server applications with confidential message exchange created in assignment 1 to provide authentication, integrity and key sharing among both the client and server. Implement the RSA algorithm from scratch that will be used for secret key encryption and digital signature

- ## Built with

  The python language is used to build this project

- ## Project Structure

  The project folder contains 5 files

  1. constants.py
  2. client.py
  3. server.py
  4. encryption.py
  5. decryption.py
  6. rsa.py
  7. output_client.png
  8. output_server.png

  Function of each file:

  - **constants.py:**

    This file contains all constants which are used in process of encryption and decryption. For example, S-box;

  - **client.py:**

    This file contains code for client side. In this file user input is encrypted and sent to server using socket programming.

  - **server.py:**

    This file contains code for server side. In this file encrypted cipher side is received from client which is then decrypted and plain-text is generated.

  - **encryption.py**

    This file contains code for encryption of plain-text.

- **decryption.py**

  This file contains code for decryption cipher-text.

- **rsa.py**

  This file contains implementation of RSA algorithm.

- **output_client.png**

  This is image output file of client.

- **Output_server.png**

  This is image of server output.

Function Detailed description

In encryption.py:

1. **encrypt**

   This function take plaintext as parameter and then encrypt it and return cipher-text.

2. **addRoundKey**

   This function add round key in gf(16). It take 2 number as parameter and return their sum.

3. **shiftRows**

   Responsible for shift-rows of state matrix. It take state matrix as parameter and return resultant state matrix.

4. **substituteNibbles**

   This function is responsible for nibble substitution using S-box. Each nubble in input is used in encryption s-box to generate output nibble.

5. **mixColumns**

This function is responsible for transformation of state matrix. In this function state matrix is passed as parameter which then us multiplied with constant matrix. And resultant matrix is returned.

6. **keyExpansion**

   This function is used to create three 16 bit round keys from one 16-bit cipher key.

7. **intToState**

   This function converts 2-byte integer into 4X4 state matrix

8. **stateToInt**

   This function converts 4X4 state matrix into 2-byte integer

In decryption.py:

1. **decrypt**

   This function take plaintext as parameter and then encrypt it and return cipher-text.

2. **addRoundKey**

   This function add round key in gf(16). It take 2 number as parameter and return their sum.

3. **shiftRows**

   Responsible for shift-rows of state matrix. It take state matrix as parameter and return resultant state matrix.

4. **substituteNibbles**

   This function is responsible for nibble substitution using S-box. Each nubble in input is used in encryption s-box to generate output nibble.

5. **inverseMixColumns**

   This function is responsible for inverse transformation of state matrix.

6. **keyExpansion**

This function is used to create three 16 bit round keys from one 16-bit cipher key.

7. **intToState**

   This function converts 2-byte integer into 4X4 state matrix

8. **stateToInt**

   This function converts 4X4 state matrix into 2-byte integer

In server.py:

1. **start**
   This function accepts cipher-text which is send by client and then decrypt is and print the plain-text.

In rsa.py:

1. **egcd**
   This function calculates the modular inverse from e and phi

2. **gcd**
   This function calculates the gcd of two numbers

3. **is_prime**
   This function checks if number is prime

4. **generate_keyPairs**

   This function Generate key pairs for public and private key

5. **rsa_encrypt**

   This function is implementation of encryption by RSA algorithm

6. **rsa_decrypt**
   This function is implementation of decryption by RSA algorithm

- **How to run?**

  First open project in any python-ide, for example, VS-code.
  Then first run the server.py.
  After running server.py in another terminal run client.py.