NUMBER THEORY AND CRYPTOGRAPHY PROJECT

SECURE CHAT

USING RSA ENCRYPTION AND LUCAS-LEHMER TEST

DEVELOPED BY

RAHUL AGRAWAL (09CO71)

FEATURES

- A Secure TCP-IP based Client – Server chat application.
- Security is ensured by RSA encryption-decryption.
- Support for concurrent chat with multiple clients and the server.
- Encryption is ensured between any two participants i.e. one client and a server, so any other client would not be able to decrypt their messages. This ensures the security parameter.
- Each user (client) and server have their own set of public and private keys.

RSA ENCRYPTION

- Rivest-Shamir-Adleman created a public-key cryptography algorithm which uses prime factorization problem to ensure security.

  - Define $n \equiv pq$; for **p** and **q** primes.
  - Also define a private key **d** and a public key **e** such that

    $$\mathbf{(d, \phi(n)) = 1}$$

    $$\mathbf{de \equiv 1 (mod \phi(n))}$$

    Where $\phi(n)$ is the totient function, (a,b) denotes the GCD, and $a \equiv b \pmod{n}$ is a congruence.
  - Let the message be converted to a number **M**. The sender then makes **n** and **e** public and sends $\mathbf{c \equiv m^e \pmod{n}}$.
  - To decode, the receiver (whose private key is **b**) computes

    $$\mathbf{m \equiv c^d \pmod{n}}$$

LUCAS-LEHMER TEST

- Let **n** be a positive integer. If there exists an integer $1 < a < n$ such that

$$\mathbf{a^{n-1} \equiv 1 \pmod{n}}$$

and for every prime factor **q** of **n − 1**

$$\mathbf{a^{(n-1)/q} \not\equiv 1 \pmod{n}}$$

then n is prime. If no such number a exists, then n is either 1 or composite.

- The reason for the correctness of this claim is as follows:
  - If the first equality holds for **a**, we can deduce that a and n are co-prime. If **a** also survives the second step, then the order of **a** in the group **(Z/nZ)*** is equal to **n−1**, which means that the order of that group is **n−1** (because the order of every element of a group divides the order of the group), implying that **n is prime**.
  - Conversely, if n is prime, then there exists a primitive root modulo n, or generator of the group (Z/nZ)*. Such a generator has order $|(Z/nZ)^*| = n-1$ and both equalities will hold for any such primitive root.

FLOW OF CONTROL

- **Server End**
    1. Enter Server Port.
    2. Enter **p** and **q**; check if primes.
    3. Compute **n** and **PHI(n)**.
    4. Enter private key, **d** and compute **e.**
    5. When Client joins, exchange public keys with client automatically.
    6. Enter Message; encrypt using public key of client.
    7. Send cipher-text to the client.
    8. Receive Cipher-text from Client and decrypt it using server's private key, **(d, n)**

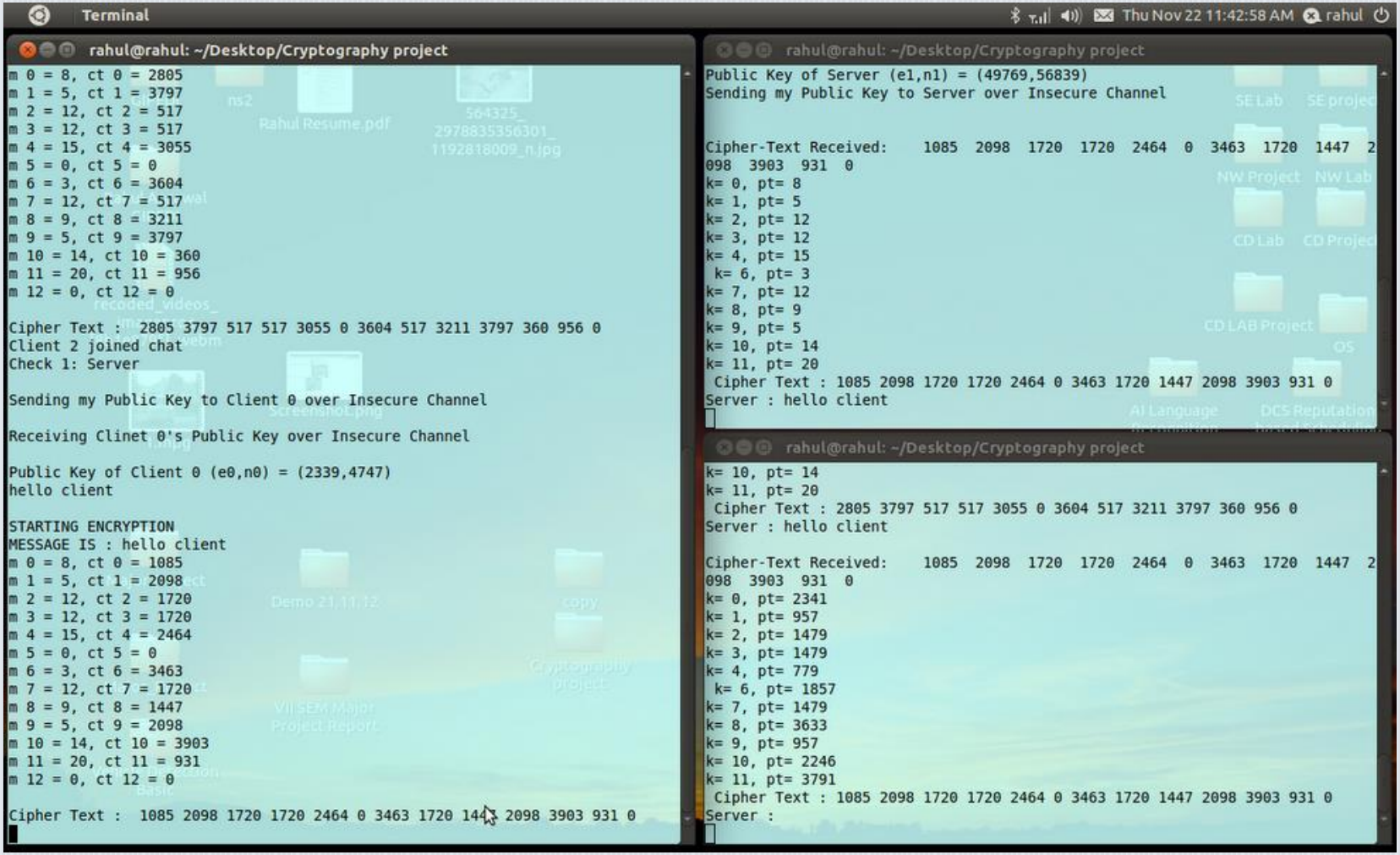
FLOW OF CONTROL

- **Client End**
    1. Enter Server Port and IP address.
    2. Enter **p'** and **q'**; check if primes.
    3. Compute **n'** and **PHI(n')**.
    4. Enter private key, **d'** and compute **e'.**
    5. Exchange public keys with server automatically.
    6. Receive Cipher-text from Server and decrypt it using client's private key, **(d', n')**
    7. Enter Message; encrypt using public key of server.
    8. Send cipher-text to the server.

How is Security Ensured?

- Let us assume a case where some attacker is trying to access the channel and gets the cipher-text.
- To crack the cipher-text, he must get private key of the party for whom the message is meant.
- To get **d**, he must know **e, n and PHI(n)**
- Since there is no way that he can get **PHI(n)** from **(e,n)** and the problem of trial and error is a hard problem, security is ensured.
- Also, if the number of digits in each block and the value of p and q are large , decryption without proper private key takes a large amount of time.

RESULTS


Terminal
Thu Nov 22 11:42:58 AM rahul
rahul@rahul: ~/Desktop/Cryptography project
m 0 = 8, ct 0 = 2805
m 1 = 5, ct 1 = 3797
m 2 = 12, ct 2 = 517
m 3 = 12, ct 3 = 517
m 4 = 15, ct 4 = 3055
m 5 = 0, ct 5 = 0
m 6 = 3, ct 6 = 3604
m 7 = 12, ct 7 = 517
m 8 = 9, ct 8 = 3211
m 9 = 5, ct 9 = 3797
m 10 = 14, ct 10 = 360
m 11 = 20, ct 11 = 956
m 12 = 0, ct 12 = 0
Cipher Text : 2805 3797 517 517 3055 0 3604 517 3211 3797 360 956 0
Client 2 joined chat
Check 1: Server
Sending my Public Key to Client 0 over Insecure Channel
Receiving Clinet 0's Public Key over Insecure Channel
Public Key of Client 0 (e0,n0) = (2339,4747)
hello client
STARTING ENCRYPTION
MESSAGE IS : hello client
m 0 = 8, ct 0 = 1085
m 1 = 5, ct 1 = 2098
m 2 = 12, ct 2 = 1720
m 3 = 12, ct 3 = 1720
m 4 = 15, ct 4 = 2464
m 5 = 0, ct 5 = 0
m 6 = 3, ct 6 = 3463
m 7 = 12, ct 7 = 1720
m 8 = 9, ct 8 = 1447
m 9 = 5, ct 9 = 2098
m 10 = 14, ct 10 = 3903
m 11 = 20, ct 11 = 931
m 12 = 0, ct 12 = 0
Cipher Text : 1085 2098 1720 1720 2464 0 3463 1720 1447 2098 3903 931 0
rahul@rahul: ~/Desktop/Cryptography project
Public Key of Server (e1,n1) = (49769,56839)
Sending my Public Key to Server over Insecure Channel
Cipher-Text Received: 1085 2098 1720 1720 2464 0 3463 1720 1447 2098 3903 931 0
k= 0, pt= 8
k= 1, pt= 5
k= 2, pt= 12
k= 3, pt= 12
k= 4, pt= 15
k= 6, pt= 3
k= 7, pt= 12
k= 8, pt= 9
k= 9, pt= 5
k= 10, pt= 14
k= 11, pt= 20
Cipher Text : 1085 2098 1720 1720 2464 0 3463 1720 1447 2098 3903 931 0
Server : hello client
rahul@rahul: ~/Desktop/Cryptography project
k= 10, pt= 14
k= 11, pt= 20
Cipher Text : 2805 3797 517 517 3055 0 3604 517 3211 3797 360 956 0
Server : hello client
Cipher-Text Received: 1085 2098 1720 1720 2464 0 3463 1720 1447 2098 3903 931 0
k= 0, pt= 2341
k= 1, pt= 957
k= 2, pt= 1479
k= 3, pt= 1479
k= 4, pt= 779
k= 6, pt= 1857
k= 7, pt= 1479
k= 8, pt= 3633
k= 9, pt= 957
k= 10, pt= 2246
k= 11, pt= 3791
Cipher Text : 1085 2098 1720 1720 2464 0 3463 1720 1447 2098 3903 931 0
Server :