IITb ENCRYPTO-MESSENGER

**Team: 522**

**T.A: Pallav Vasa**

**Team Members:**

**Divyam Bansal 110050086**

**Ishan S. Bhadauria 110040085**

**Prithviraj.M.Billa 110050065**

**Divyank Agarwal 110260015**

**Deepak Chaudhary 110040016**

**Deepak prajapathi 115280015**

Tables of contents:

1.0 problem statement

1.1 project description.

1.2 Main parts of the project.

1.2.1 Data Transfer through sockets.

1.2.2 Encryption and Decryption.

1.2.3 Generation of a Shared secret Key. (DIFFIE HELLMAN KEY EXCHANGE)

1.3 Systems Requirements.

1.4 basic mechanisms

1.5 utility

1.6 Basic goal of project

1.7 References.

1.8 Project division.

1.9 future ideas to improve the project

1.10 user manual

1.11 status of completion

1.12 project progress

1.13 functions included

1.14 conclusion

1.0 problem statement

To ensure***secure*** ***data transmission*** between two system.

1.1 Project description:

Our project targets at transmission of data in encrypted form. It exchange data in encrypted form and can only be decrypted by the user who knows the correct security key.

Our Project IITB ENCRYTPO-MESSENGER is a chat box in which you can share messages and text files safe and securely on LAN. Messages and files of limited size are shared through IPV4 (Internet Protocol version 4) connection through 64-bit encryption (but length of prime no. produced is small currently) which is based on block cipher Method (block cipher chaining). Using this our data can be securely transmitted without being hacked.

1.2 Main parts of the project:

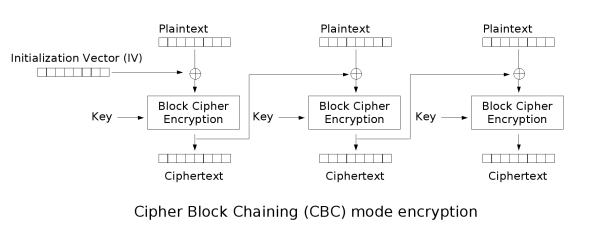
1.2.1 Data transfer

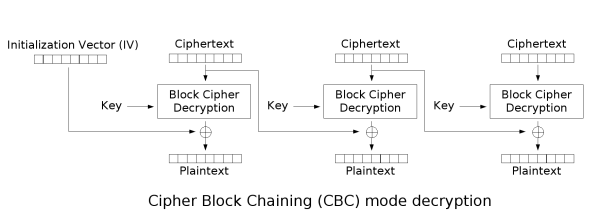
Scope for sending and receiving the data in c++ programing language is through sockets (*an endpoint for communication*). We are using IPV 4(Internet Protocol Version 4) in our project. We have include three libraries namely sys/types.h , sys/socket.h, netinet/in.h in the project. And using some predefined classes and function in those libraries in our project for example: socket, sockaddr\_in etc.We have included the IPV4 protocol so that in future we can connect to host kowing its IP address.

Threading is not included in our project :(

1.2.2 Encryption and decryption

Our encryption is based on the cipher-block chaining .It is called Private Key cryptography (symmetric Key algorithm) same key used for encryption and decryption. In the **cipher-block chaining** (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point.

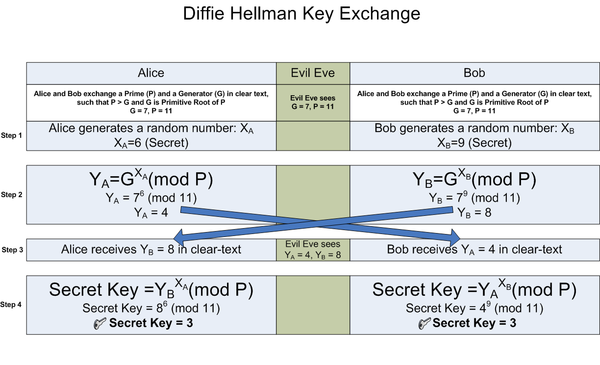




http://en.wikipedia.org/wiki/Block\_**cipher**\_modes\_of\_operation

1.2.3 Generation of a shared secret key (Diffie Hellman key exchange)

Diffie Hellman key exchange method deals about generation of safeprime and generator (which is primitive root modulo p) .A prime (q) is a safe prime if (q-1)/2 is also a prime. (Why only safe prime?).As using safe prime makes it difficult to decrypt the messages without knowing secret key. It involves lot to know about number theory.



<http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Alice** | | | Knows | doesn't know | | p = 23 | b = ? | | base g = 5 |  | | a = 6 |  | | A = 56 mod 23 = 8 |  | | B = 5b mod 23 = 19 |  | | s = 196 mod 23 = 2 |  | | s = 8b mod 23 = 2 |  | | s = 196 mod 23 = 8b mod 23 |  | | s = 2 |  | | |  |  | | --- | --- | | **Bob** | | | Knows | doesn't know | | p = 23 | a = ? | | base g = 5 |  | | b = 15 |  | | B = 515 mod 23 = 19 |  | | A = 5a mod 23 = 8 |  | | s = 815 mod 23 = 2 |  | | s = 19a mod 23 = 2 |  | | s = 815 mod 23 = 19a mod 23 |  | | s = 2 |  | | |  |  | | --- | --- | | **Eve** | | | knows | doesn't know | | p = 23 | a = ? | | base g = 5 | b = ? | |  | s = ? | | A = 5a mod 23 = 8 |  | | B = 5b mod 23 = 19 |  | | s = 19a mod 23 |  | | s = 8b mod 23 |  | | s = 19a mod 23 = 8b mod 23 |  | |

<http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>

HERE EVE IS THE HACKER WHO IS TRYING TO EXTRACT THE KEY OUT OF THEIR CONVERSATION.BUT UNABLE TO DO THAT.

1.3 system requirements.

The requirement is the code we have written in the program is compatible in Ubuntu, but some minor changes have to done to compile in other O.S.

The both server and client are connected over LAN provided the socket is using IPV 4 protocol.

1.4 Basic mechanism

First of all connection is established between two systems using standard library functions.Now a secret key is generated by diffie hellman mechanism. The method of key generation is such that only the connected systems will have the secret key which is quite random. This key involves in data encryption using cipher block chaining. The original information is transformed to the encrypted content using the secret key, then with the help of secret key at the receiving end it is transformed again into readable form.

1.5 utility

The key helps other individuals to convert the data into encrypted form and the secret key of the intended user can only decrypt it. Cryptography is used to hide crucial information. This mechanism helps in preventing the leakage of important information from the network. The use of this method is beneficial for almost all professional departments. It can also help the investigation agencies to keep their data secure.

In our project in each session a different secret key is produced which is quite random.

1.6 Basic goal of project

The main goals of this are authentication, integrity, confidentiality & reliability. The purpose of confidentiality is to provide security to the data from the attack of any foreign agent. It is in this step when there is need to use secret key to encrypt data. The authentication is helpful in providing sufficient evidence of the true owner of the data.

1.7 references:

For sockets Wikipedia,C++ tutorials.

For encryption and decryption:

<http://en.wikipedia.org/wiki/Block_cipher>

For Diffie Hellman key exchange process:

<http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>

<http://www.ietf.org/rfc/rfc4419.txt>

[www.cplusplus.com](http://www.cplusplus.com) for other commands in our project.

For encryption and decryption

http://en.wikipedia.org/wiki/Block\_**cipher**\_modes\_of\_operation

1.8 project division:

**Ishan , Divyam** ----- Data transfer through sockets, Integration of the project, presentation on terminal.

**Deepak chaudhary,Deepak prajapati**---- Encryption and decryption.

**Prithvraj, Divyank** --------- Generating the security key using Diffie Hellamn key exchange process.

1.9 future ideas to improve the project:

* We can implement it as 64 bit encryption or 128 bit encryption (which Gmail is using).
* We have limited the generator to 2 for now , but we can write code for generator for every specific safe prime( At present we have written the code but due to lack of time we are not implementing it)
* We can involve threading in our project which enables both reading mode and writing mode simultaneously. Using which user can receive and send message on different thread. As right now turns come alternately i.e. once user have sent the message he has to weight for receiving the message.
* We can include graphics which will make my messenger colourful.
* Currently we are using hostname to connect but we can use IP address to connect which will be a better.
* To make a better user friendly interface.
* To use better and more secure encryption and decryption program.

1.10 user manual

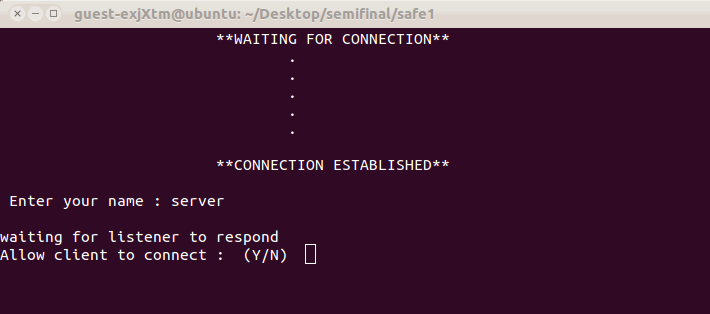
1. First Server needs to run his program.

2. Then client will run his program with arguments (./filename hostname). Here hostname is essential to connect with server.

(here localhost is the host name of server).

3. Then both are required to enter their name. And server will be asked whether to accept the request of client depending on which there conversation begin.

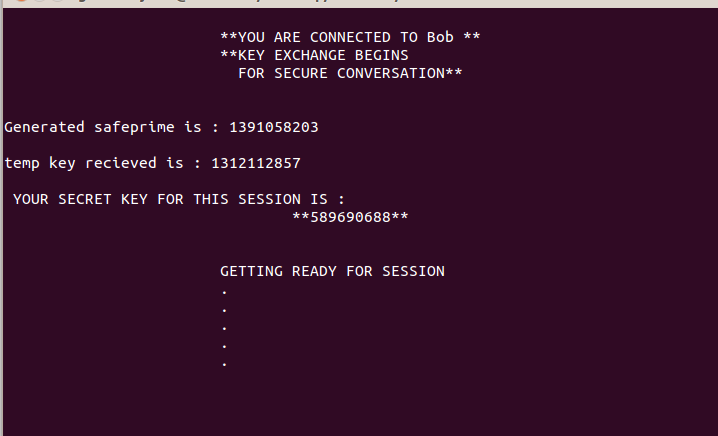
Display on server window



Display on client window

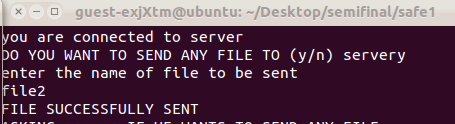
4. After that key will be exchanged automatically.

Same display on client and server window.

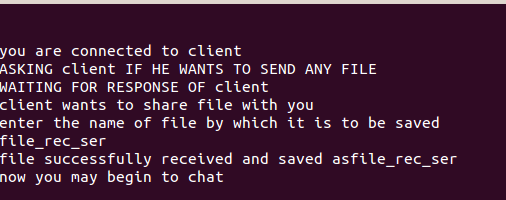


5.Client will be asked if he wants to send any file to server, in case he says yes he will be asked to input the file name which is in the same directory and consequently server will be asked to give a name to file in which received file is to be saved.

Display on client window



Display on server window



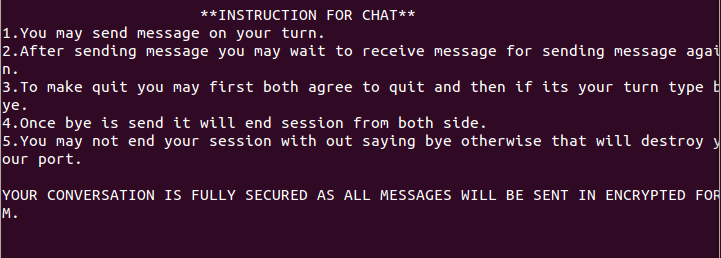
6. After this server will be asked if he wants to send any file to client and same process proceed as mentioned above.

Display on server window



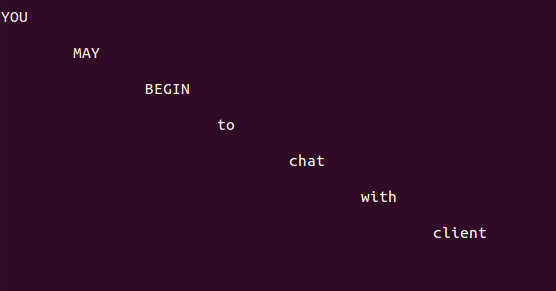
7. After this file transfer some instruction will be displayed.

Similar display on client and server window



8. This will be displayed for 20 secs.And after that chat will begin.

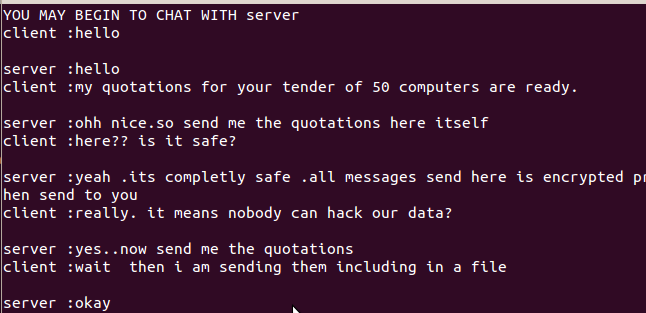
Display on server window



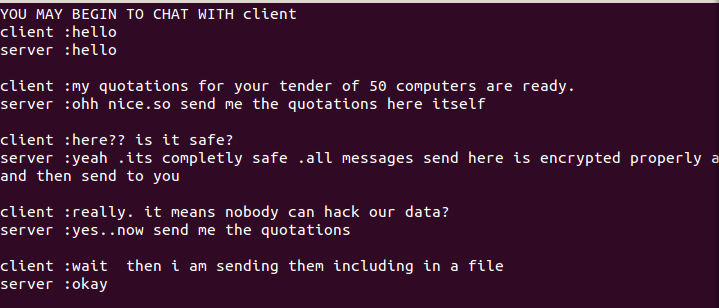
A similar display on client window.

9. And finally both are allowed to transmit message securely.

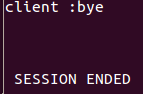
Display on client window



Display onserver window



10. Chat can be ended by saying “bye”.

  Display onserver and client window

1.11 status of completion

Our project is ***completed***. It could have been further modified by including a generator producing program and a big prime no. Generator but due to lack of time it could not be included. Code for above mentioned programs is ready.

But on other hand we can easily exchange file and messages by current version.

1.12 project progress

First we thought to make a game like Dave or Sudoku. But we were still thinking to make some new and much relevant project. Than while discussing with our TA we came to know about cryptography, RSA and several other things related to secure communication. In present scenario as it has become essential for many institutes to exchange important information securely and to prevent it from being getting hacked by anyone else. Thus seeing its relevance we all began discussing and gave modifications to the task and finally with consent of our TA decided to make project on**” exchanging messages in encrypted form” through network.** Also we all were fascinated by the fact that we were going to make a chatting application based on encryption.

After finalizing the topic there was no clue on how to start as the knowledge level was not up to the optimum level. So first of all we searched for different theoretical aspects of programming and other things required to develop algorithm. Also our TA **Pallav Vasa** guided our energy in proper direction and helped us to learn some new and ultimate concepts related to programming and debugging.

Our project task was decided on 30th September.

**Our first meeting**

**Venue: H3 comp room**

**2nd October @10pm-2am.** Meeting attended: By all members

Task: We divided the work among ourselves with help of our TA. Then TA made everybody his task clear. We searched for the things in the Google and got several new things.

**Our second meeting:**

**Venue: osl**

**Day and date: Friday 7th October. 8:30-10:30pm (lab timings)**

Attended by all

Task: discussed more about the topic and the problems being faced. Like in transmitting data we were facing problem for the socket connections which we discussed with our TA. Also other member discussed there problem with us.

**Our third meeting:**

**Venue: osl**

**Day and date: Sunday 9th Oct, 2pm-4pm**

Attended by all

Task: made some algorithm and solved the problem being faced in their execution. Googled several new things related to transmission and learnt about new library function.

**Our fourth meeting:**

**Venue: h3 comp room**

**Day and date: 12th Oct, Wednesday 9pm-10:30pm**

Attended by all

Task: Divided the task further and set their deadlines. Discussed about the changes being made to the algorithm of encryption.it was made much safer as in new method whole message was divided in parts and each part was encrypted by the previous encrypted path. Also developed algorithm to calculate a secure key based on number theory. Discussed what should be the nature of the generator and prime number to be used for the calculation of key. Like prime number was supposed to be a safe prime which is a prime p such that (p-1)/2 is also prime. If such number is chosen than p-1 is very difficult to be factorized thus making the key more secure where p is a very large prime number. Generator was supposed to be like that any number raised to its power will give all possible remainder when divided by p.

After this meet divyam and ishan had some more discussion with pallav till 1:30am. We discussed about the encryption method and discovered that in applying xor operator (which works bit by bit) between an integer value (32 bits) and a character (8bits) the initial 24 bit of integer are truncated and only remaining 8 bits participate in XOR. Also if we have defined a variable to be char and then if we take its input from user (by cin) and if we enter a two digit integer then it accepts only first digit as input treat it as character rather than integer. After this all we discussed about sockets and connections and came to know that by gethostbyname function it can only connect to a system in same wing.

**Our fifth meet:**

**Venue: osl**

**Day and date: 14th October, Friday, 8:30-10:30 pm.**

Task: Further discussed about the problems being faced by the member. By this time transmission was occurring between computers but problem being faced was that after sending one message we have to wait for one message from server otherwise we could not chat further. Also made other member clears how to get a bit expansion of a number as it was required for encryption. Also saw whether team was doing work on an appropriate pace and asked other to immediately interact whenever they face problem.

**Our sixth meet:**

**Venue: osl**

**Day and date: 15th October, Saturday, 2-4 pm.**

Attended by Ishan and Divyam

Task: worked on coding of data transmission system and creating sockets and other essentials in communication between systems. Also tried to solve problem of alternate sending and receiving in the algorithm. But unable to do so…efforts still going on.

**Our 7th meet:**

**Venue: h3 comproom**

**Day and date: 19th October, Wednesday, 10pm-12pm**

Task: as everybody was having their quiz in this week so work was slow. Prithvi prepared a code for calculating secure key but there were problem in that as it was not an efficient code. Then we discussed the changes that could be made to that came up with a new short and efficient code. At the end the code was ready but again some problem was there and then we learned about debugging our program using (-ggdb –wall) on terminal and we were successful in locating the mistake and at last we were with a working code.

**Our 8th meet:**

**Venue: H3 comproom**

**Day and date: 2nd November, Wednesday, 10-11pm**

Task: After returning from the Diwali break to remind everybody again about our on-going project meet was called. Final allocation of our jobs was done and we all decided to complete remaining algorithms on upcoming Friday and Saturday. And planned about completion of project.

**Our 9th meet:**

**Venue:OSL**

**Day and date:4th November, Friday, 8:30-11pm**

Task: Everybody was busy in doing their project completion.

With ishan and divyam achieving data transmission without any encryption or decryption but with few problems too. Both Deepak improved their algorithm on encryption and decryption.

**Our 10th meet:**

**Venue: physics department lab**

**Day and Date: 5th November, Saturday, 2-6:30pm**

Task: All team members met and finalized all the programs. The program for calculating safeprime was made more efficient. Problem of proper closing of the socket was completed to an extent. At last all the individual program was ready.

**Our 11th meet:**

**Venue:h2 comproom**

**Day and date:6th November, Sunday, 2-5pm, 5:30-8pm and 10-11:45pm**

**Attended by:Ishan and Divyam**

Task: To integrate whole project.Several problems were faced in compiling whole project.Integration included giving proper name to variables, defining functions, commenting. Process included many Human errors which were resolved by each other’s co-operation. As our program dealt with large no.s there was problem on data type defining. We integrated whole program on divyam’s laptop which had 64 bit OS but when we run the same program in comproom it gave segmentation error. Problem was resolved after using appropriate data types.

Meanwhile on Monday, Tuesday indentation , commenting was done. Also the problem of quiting the chat by saying bye was solved. But problem faced was in sending long messages only some part was transmitted. Also in Encryption whole key wasn’t used due to difference in byte size of character and integer.

**Our 12th meet :**

**Venue: H3 comproom**

**Day and date: 9th November, Wednesday, 9-11:30pm**

Task: To give the flavour of each other’s part to every member. And discussing how the terminal should like i.e. presentation. Active participation by all members in discussing how should terminal look like.

After this meet work on terminal presentation was started by divyam and ishan. Learnt many new commands like clearing the terminal and delaying the time.

After that started doing work and it almost took 4-5 hours to achieve that.

**Our 13th meet:**

**Venue : H2 comproom**

**Day and date:10th November, Thursday, 7-8pm**

Task :Debugging of the last task of presentation was being done by divyam ,divyank&ishan.

On Thursday , decided to be able to send a file also.

So divyam took this task for transferring file. Again it took almost 5 -6 hours in achieving this task and debugging of few things.

**Our 14th meet:**

**Venue:Osl**

**Day and Date: 11th November,Friday, 8:30-10pm, 2-4am**

Task: Ran our program successfully. Did peer review which almost completely in 20mins.

Did documentation.

1.13 functions included

Keygen – function which calculates (g^x) modp. Where g, x and p are passed as parameters. These are large numbers but function calculates it efficiently with taking minimal time.

Encrypt- function which encrypts a message using cipher block chaining. In this the char array is passed and a secret\_key of the session is passed.

Decrypt- function which decrypts a message using cipher block chaining. In this the encrypted char array is passed and a secret\_key of the session is passed.

Safeprime.h- it is a header file which generates a random safeprime.

Sleep- It is a predefined function in “unstid.h” which is used to delay the time. so in places where time delay occurs we have used the sleep command and not due to slow algorithm.

System(“clear”)- it is ysed to clear the terminal screen.

// some reference is taken from the google for the following transmission code

sys/types.h-This header file contains definitions of a number of data types used in system calls. These types are used in the next two include files.

sys/socket.h-The header file socket.h includes a number of definitions of structures needed for sockets.

netinet/in.h-The header file in.h contains constants and structures needed for internet domain addresses.

Struct sockaddr\_in –predefined structure having foolowing members

{Short sin\_family; /\*chosen to be AF\_INET\*/

U\_short sin\_port;

Struct in\_addr sin\_addr;

Char sin\_zero[8]; /\*not used , must be zero\*/}

Struct in\_addr

Contains one field a unsigned long s\_addr.

Socket()- creates a new socket and accept address domain, type of socket and protocol as parameters.

 bzero()- sets all values in a buffer to zero. It takes two arguments, the first is a pointer to the buffer and the second is the size of the buffer.

htons() -which converts a port number in host byte order to a port number in network byte order.

 bind()-  binds a socket to an address.

Listen()- allows the process to listen on the socket for connections

 accept() causes the process to block until a client connects to the server.

Write() – allows user to send message after connection.

Read() – allows user to receive the message.

1.14 conclusion

The journey of this cs101 project has been enthusiastically learning and has contributed much to our intellectual and understanding ability towards computer programming.

Our team would like to thank our T.A PALLAV VASA to make the above project possible.

HOPING NEW VERSION OF PROJECT WILL BE USED AS STANDARD MESSENGER IN IITB SOON.