Project 2: Mental Poker

Submitted By: Anshul Jain (ajain89), Rahul Jain (rjain29)

It is a virtual poker that is played between two players over a network. It uses SRA protocol that makes the game reliable. The encryption/Decryption algorithm to be used need to be commutative.

Socket Communication

Sockets are interfaces that can plug into each other over a network so that the programs connected can communicate. Internet sockets constitute a mechanism for delivering incoming data packets to the appropriate application process or thread, based on a combination of local and remote IP addresses and port numbers.

We are using TCP sockets as it is more reliable than UDP sockets.

The socket primitives for TCP:

Primitive	Meaning
SOCKET	Create a new communication end point.
BIND	Attach a local address to the socket.
LISTEN	Announce willingness to accept the connections: give queue size.
ACCEPT	Block the caller until a connection attempt arrives.
CONNECT	Actively attempt to establish a connection.
SEND	Send some data over the connection.
RECEIVE	Receive some data from the connection.
CLOSE	Release the connection.

We are using the localhost IP address (127.0.0.1) which can be changed to any IP address provided the Server (Bob) and Client (Alice) are on the same network.

Encryption of Cards

1. Bob (server) announce willingness to accept the connections.
2. Alice (client) establishes connection with Bob.
3. Bob selects a prime number ‘n’ and sends it to Alice.
4. Based on the value of ‘n’, Bob selects his set of public and private keys using the following :
   Public key => Select key such that GCD(key,n-1)=1.
   Private key=> Modular inverse of Public key i.e. (public key * private key) mod (n-1)=1.
5. Using the above algorithm and the value of ‘n’, Alice also selects her set of public and private keys.
6. Bob takes a deck of 52 cards (numbered from 2 to 53) and shuffles them. After encrypting the cards with his private keys, he sends the cards to Alice.

Card selection and exchange using the encryption protocol

1. Alice selects 10 random cards from the deck. She takes the first 5 cards and encrypts them with her public key and sends them to Bob (Alice’s cards).
2. She sends the next 5 cards as well to Bob (Bob’s cards).
3. Bob decrypts all the 10 cards using his private key and sends Alice’s cards back to Alice.
4. Alice now decrypts her 5 cards using her private key.

To see who won the game, both announces their private keys. The following steps take place:

1. Alice announces her private key to Bob.
2. Using Alice's private key, Bob decrypts Alice's cards and compares with his cards.
3. Bob announces his private key to Alice and Alice also decrypts Bob's cards to compare with her cards.
4. Both then announces the results.

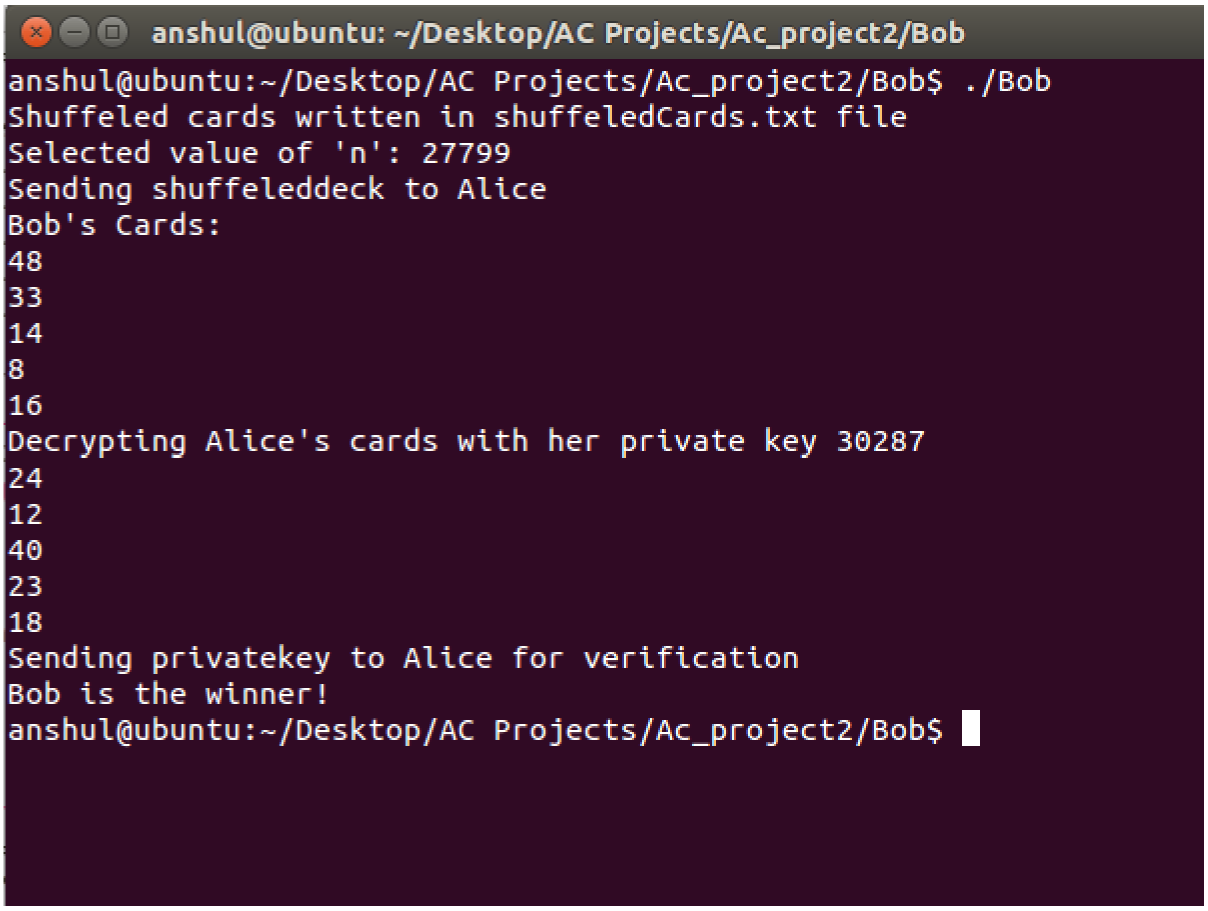
Verification of non-cheating once all cards are played

We can easily see that no cheating occurred in the above algorithm. Alice selects 10 random cards from 52 cards encrypted by Bob. Since Bob knows what the face value of the shuffled cards is, he is not allowed to select the cards and so Alice selects her as well as Bob's cards. To further confirm that no cheating occurred, we are printing the value of the cards shuffled by Bob in a text file (shuffledCards.txt on the server/Bob's folder) and the indices selected by Alice in a text file (SelectedCards.txt on the client/Alice's folder). By comparing these files and the data displayed, we can be sure that no cheating occurred.

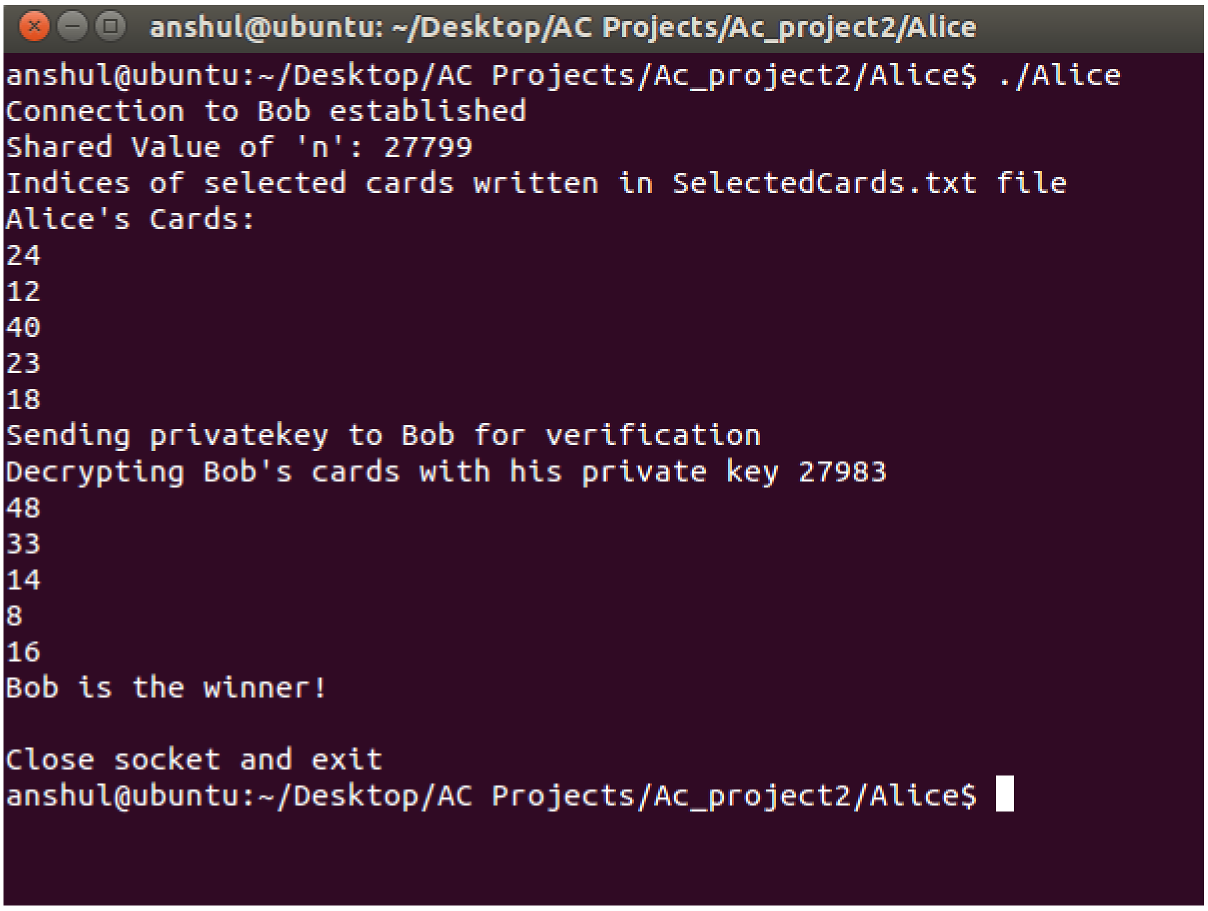
To determine the winner, we are using the following algorithm:

1. Take the sum of both Bob's and Alice's cards and compare them. Player having the higher sum is the winner.
2. If the sum for both Alice's and Bob's cards are same, we take the product of the cards to determine the winner.

Following are the screenshots for game played between Alice and Bob.



```
anshul@ubuntu:~/Desktop/AC Projects/Ac_project2/Bob$ ./Bob
Shuffeled cards written in shuffeledCards.txt file
Selected value of 'n': 27799
Sending shuffeleddeck to Alice
Bob's Cards:
48
33
14
8
16
Decrypting Alice's cards with her private key 30287
24
12
40
23
18
Sending privatekey to Alice for verification
Bob is the winner!
anshul@ubuntu:~/Desktop/AC Projects/Ac_project2/Bob$ 
```



anshul@ubuntu: ~/Desktop/AC Projects/Ac_project2/Alice

```
anshul@ubuntu:~/Desktop/AC Projects/Ac_project2/Alice$ ./Alice
Connection to Bob established
Shared Value of 'n': 27799
Indices of selected cards written in SelectedCards.txt file
Alice's Cards:
24
12
40
23
18
Sending privatekey to Bob for verification
Decrypting Bob's cards with his private key 27983
48
33
14
8
16
Bob is the winner!

Close socket and exit
anshul@ubuntu:~/Desktop/AC Projects/Ac_project2/Alice$ 
```

<u>**Shuffled Cards by Bob:**</u>

9	25	4	46	45	15	23	24	12
27	17	39	11	48	5	51	42	38
22	21	37	33	14	28	40	16	2
53	18	6	34	49	19	43	47	10
36	50	26	44	32	30	35	7	3
52	41	31	29	8	13	20		

<u>**Selected Cards Indices (0 to 51) by Alice:**</u>

Indices of Alice's' cards

7, 8, 24, 6, 28

Indices of Bob's' cards

13, 21, 22, 49, 25