

Project Two Report

Paul Jayme | 300222932

Task One:

EchoClient.java:

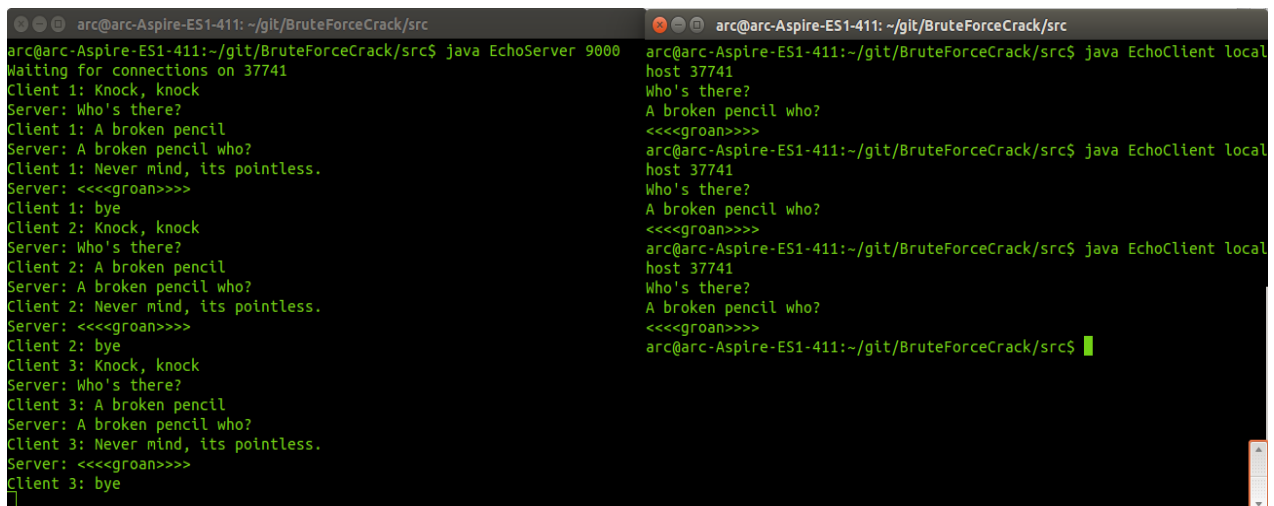
```
// send a sequence of messages and print the replies
out.println("Knock, knock");
System.out.println(in.readLine());
out.println("A broken pencil");
System.out.println(in.readLine());
out.println("Never mind, its pointless.");
System.out.println(in.readLine());
out.println("bye");
```

- When an EchoClient is launched it establishes its communication channels then sends a sequence of messages to the EchoServer. Here the code has been modified to a different knock-knock joke. The server must then recognize what the client is saying so it also has to be modified from that end.

Connection.java:

```
if (line.equals("Knock, knock")) {
    response = "Who's there?";
} else if (line.equals("A broken pencil.")) {
    response = "A broken pencil who?";
} else if (line.equals("Never mind, its pointless.")) {
    response = "<<<<groan>>>>";
}
```

- When a new thread is created by the EchoServer class it uses the Connection class as its main Thread class (since Connection extends Thread). Hence when "Connection.start()" is called, the program executes the "run()" method of the Connection class which launches the knock-knock joke interaction between the client and the server.



The image shows two terminal windows side-by-side. The left window shows the EchoServer running on port 9000, and the right window shows the EchoClient running locally. The EchoClient sends a sequence of messages: "Knock, knock", "A broken pencil", "Never mind, its pointless.", and "bye". The EchoServer responds with "Who's there?", "A broken pencil who?", "<<<<groan>>>>", and "bye".

```
arc@arc-Aspire-ES1-411: ~/git/BruteForceCrack/src
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$ java EchoServer 9000
Waiting for connections on 37741
Client 1: Knock, knock
Server: Who's there?
Client 1: A broken pencil
Server: A broken pencil who?
Client 1: Never mind, its pointless.
Server: <<<<groan>>>>
Client 1: bye
Client 2: Knock, knock
Server: Who's there?
Client 2: A broken pencil
Server: A broken pencil who?
Client 2: Never mind, its pointless.
Server: <<<<groan>>>>
Client 2: bye
Client 3: Knock, knock
Server: Who's there?
Client 3: A broken pencil
Server: A broken pencil who?
Client 3: Never mind, its pointless.
Server: <<<<groan>>>>
Client 3: bye

arc@arc-Aspire-ES1-411: ~/git/BruteForceCrack/src
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$ java EchoClient local
host 37741
Who's there?
A broken pencil who?
<<<<groan>>>>
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$ java EchoClient local
host 37741
Who's there?
A broken pencil who?
<<<<groan>>>>
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$ java EchoClient local
host 37741
Who's there?
A broken pencil who?
<<<<groan>>>>
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$
```

Above is a screenshot of the system functioning correctly - the left terminal represents the "Manager" / Server who is waiting for connections on port 37741. The right terminal represents the multiple clients that have connected to the Server. The moment a client connects to the server the full interaction is displayed on the terminal of the Server. Ultimately, the sequence of running the system starts with launching the server followed by clients connecting to the server.

Search.java:

```
int keySize = Integer.parseInt(args[1]);
byte[] key = Blowfish.asByteArray(bi, keySize);
byte[] ciphertext = Blowfish.fromBase64(args[2]);

// Go into a loop where we try a range of keys starting at the given one
String plaintext = null;
// Search from the key that will give us our desired ciphertext
for (int i=0; i< keySize; i++) {
```

- Search was initially searching based on a fixed number of keys (100 keys) – "i < 100".
- The conditional statement defined in the for-loop has now been changed to handle a dynamic / user-defined number of keys – "keySize" which is entered as the arguments of the program prior to its launch.

```
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$ java Search 3185209680 4 :+UHC
88LxQEgKq6BmdGo31UtE5HqTlmlZssAZMXqSXXXT7NJLc52Fng==
Plaintext found!
May good flourish; Kia hua ko te pai
key is (hex) BDDA7150 3185209680
arc@arc-Aspire-ES1-411:~/git/BruteForceCrack/src$
```

Above is a snapshot of the Search function working accurately – it finds the correct key based on a user-defined key-size of 4.

Task Two: Control Flow & Requirements

Key Manager:

1. The EchoServer is launched with three arguments – initial-key, key size and cipher text. The server launches on a randomized port and begins listening for incoming connections.

The server/manager is responsible for allocating key spaces for each client (i.e. the range in which each client will search the keys for) and managing the connections of each of these clients.

Upon launch, the user is prompted to enter the total number of clients – this is used to determine the “chunk-size” each client will get. For instance if we had 4 clients with a key size of 4 then there can be 4 billion possible key-variations, so the manager will look to distribute the workload equally across all 4 clients – approx. 1 billion possible key-variations for each client.

Once the server/manager has computed the accurate chunk-size then it creates the “messages” to be sent to the clients. The messages contain three key data-elements which are the “current-key”, “chunk-size” and “ciphertext”. When the client receives the message it begins executing its search.

2. Once the client has finished executing – the Manager will expect to receive results and check to see if the key has been found.
3. When the key has been found, the Manager prints success and the time it took to find the key. Consequently, if the key is not found the Manager prints a failure message.
4. The Manager shuts down once the key has been found or when it has exhausted its key space.

Client:

1. The Client simply requests for more work each time it has finished executing its task/search.
2. When a client connects for the first time it immediately pings the Server a message stating “Requesting Work..” – the Server then recognizes this and composes a message containing the search parameters and sends this to the client.
3. Once the client has received this message, it proceeds to execute its search
4. Once finished it reports back to the Manager and if the search is unsuccessful then it requests for more work but if the search returned successful then it kills the processes.

Requirements:

1. Clients are only connected to one socket which is the ServerSocket therefore it only knows the location of the Server and is completely unaware of the location of the other Clients. The architecture is setup in such a way that the two programs – Server and Client are independent processes. Once the Client has received its designated task from the Server then it independently processes this without relying on other clients.
2. Once a client has joined/connected, the Manager composes a message containing the range of keys that the client will search in along with the cipher text. The architecture guarantees that the client will first return the results of its search before sending a message to the manager that it is ready to request for more work.

3. The clients are built in such a way that once it has finished executing its search it will continuously ping the server to request for more work. The manager is able to compute and decide if there is more work left to do and in what particular key-range should this client be searching in.

4. Messages are only exchanged in two scenarios:
 - a. Manager sends the search parameters to the clients
 - b. Clients sends results back to the Manager

When a Manager or Client is doing its independent computations i.e. computing the search-range or executing the actual key-search then it is not interacting with any of the communication channels. The programs only use the channels when it has finished performing its computations and has a substantial message to send across the network.

5. When the key is found the Manager prints the key along with the time it took to find the key. Consequently, if the key is not found then it prints a failure message – stating that it has exhausted the key space and also the time it took to do the full search. When any of these scenarios occur the Manager shuts down and all client connections are terminated.

Task Three: Development & Testing

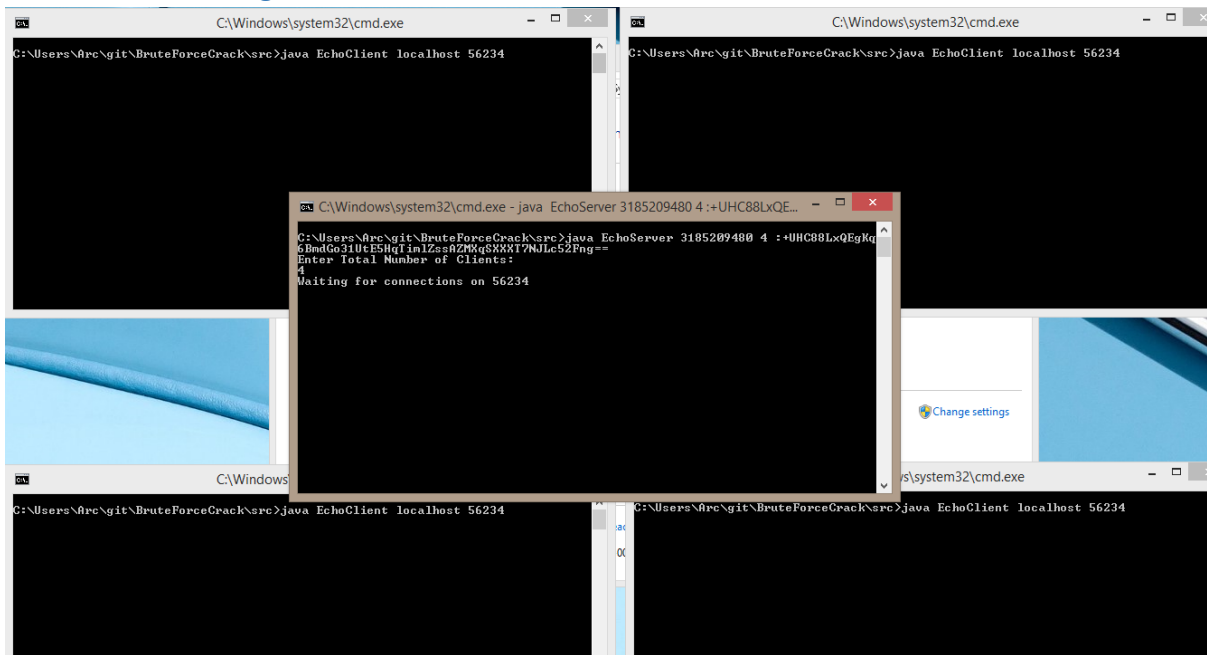
Testing Plan:

1. Launch a new terminal and run the EchoServer with the arguments:
 - a. Initial-Key: 3185209180
 - b. Key-Size: 4
 - c. CipherText: +UHC88LxQEgKq6BmdGo31UtE5HqTimlZssAZMXqSXXXt7NJLc52Fng==
2. Enter "4" as the total number of clients to be executed in parallel.
3. The Server should now display the port number it will be listening to.
4. Launch a new terminal and run the EchoClient with the arguments:
 - a. Localhost
 - b. Port number given by server

Note: The chunksize of each client is computed by the Server based on the total number of clients it is expecting hence why it is not used as an argument for launching the client.

5. Repeat step four based on the total number of clients required
6. The Manager computes the appropriate key-ranges each client should search in. I.e. client 1 = 0-100, client 2 = 101-200, etc. along with the current-key and cipher-text. Such parameters will be displayed onto the terminal and sent to each client.
7. Once the search has been completed by the client then it should return the results back to the server. The result will indicate if the key has been found or not, if it hasn't then the client will continuously ping the server to request for more work. In which case the server will send a message containing new search parameters if there is more work to be done.
8. If the key is found then the key is displayed along with the elapsed time. If the key is not found then the server displays a key-not found message along with the total time it took to exhaust the key-space. When any of the two scenarios occur, the system shuts down.

Evidence of Testing:



```
C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234

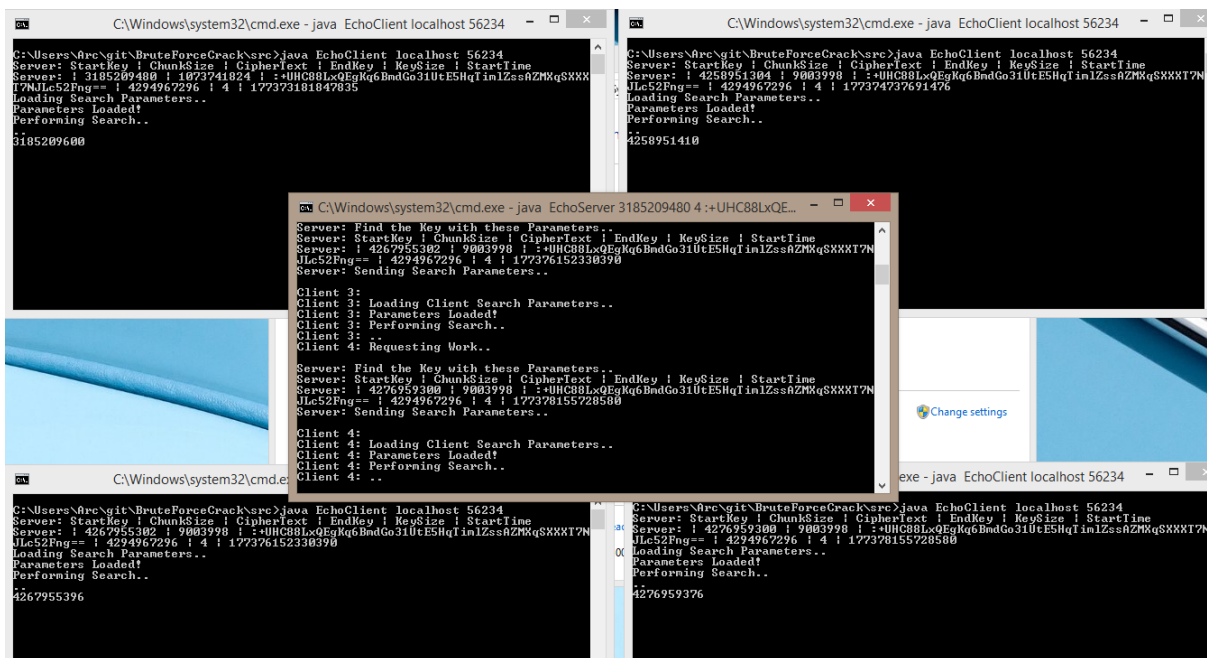
C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234

C:\Windows\system32\cmd.exe - java EchoServer 3185209480 4 :+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng==
C:\Users\Arc\git\BruteForceCrack\src>java EchoServer 3185209480 4 :+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng==
Enter Total Number of Clients:
4
Waiting for connections on 56234

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
```

Server listening for connections, four terminals setup for clients to connect.



```
C:\Windows\system32\cmd.exe - java EchoClient localhost 56234
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 3185209480 ! 1073741824 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177373181847835
Loading Search Parameters..
Parameters Loaded!
Performing Search..
3185209600

C:\Windows\system32\cmd.exe - java EchoClient localhost 56234
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 4258951304 ! 9003998 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177374737691476
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4258951410

C:\Windows\system32\cmd.exe - java EchoServer 3185209480 4 :+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng==
C:\Users\Arc\git\BruteForceCrack\src>java EchoServer 3185209480 4 :+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng==
Server: Find the Key with these Parameters..
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 4267955302 ! 9003998 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177376152330390
Server: Sending Search Parameters..
Client 3:
Client 3: Loading Client Search Parameters..
Client 3: Parameters Loaded!
Client 3: Performing Search..
Client 3: ..
Client 4: Requesting Work..
Server: Find the Key with these Parameters..
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 4276959300 ! 9003998 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177378155728580
Server: Sending Search Parameters..
Client 4:
Client 4: Loading Client Search Parameters..
Client 4: Parameters Loaded!
Client 4: Performing Search..
Client 4: ..

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 4267955302 ! 9003998 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177376152330390
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4267955396

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: ! 4276959300 ! 9003998 ! !+UHC88LxEgKq6BndGo310tESHqIinLZssAZMqSXXXI7NjLc52Fng== ! 4294967296 ! 4 ! 177378155728580
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4276959376
```

Search in progress, each client has received its respective search parameters and is now executing the search.

```
C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: 1 3185209460 ! 1073741824 ! :+UHC88LxQEqKq6BmdGo31UtESHqIinLZssAZMKqSXXX
T7MLc52Fng== ! 4294967296 ! 4 ! 1773731847835
Loading Search Parameters..
Parameters Loaded!
Performing Search..
Search Finished!
Key Found!
Plaintext found!
May good flourish; Kia hua ko te pai
Key is (hex) BDD07150 3185209600
Client Search Time: 21607785563ns

C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: 1 4258951304 ! 9003998 ! :+UHC88LxQEqKq6BmdGo31UtESHqIinLZssAZMKqSXXXIT7N
Jlc52Fng== ! 4294967296 ! 4 ! 177374737691476
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4258951519
C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
Client 3: Parameters Loaded!
Client 3: Performing Search..
Client 3:
Client 4: Requesting Work..

Server: Find the Key with these Parameters..
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: 1 4276959300 ! 9003998 ! :+UHC88LxQEqKq6BmdGo31UtESHqIinLZssAZMKqSXXXIT7N
Jlc52Fng== ! 4294967296 ! 4 ! 177378155728500
Server: Sending Search Parameters..

Client 4:
Client 4: Loading Client Search Parameters..
Client 4: Parameters Loaded!
Client 4: Performing Search..
Client 4:
Client 4: Search Finished!
Key Found!
Server: Plaintext: May good flourish; Kia hua ko te pai
Server: key is (hex) BDD07150 3185209600
Server: Client Time: 21607785563ns
Server: KEY HAS BEEN FOUND - MANAGER SHUT DOWN!
Server: Total Search Time: 21607078445ns

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>

C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: 1 4267955302 ! 9003998 ! :+UHC88LxQEqKq6BmdGo31UtESHqIinLZssAZMKqSXXXIT7N
Jlc52Fng== ! 4294967296 ! 4 ! 177376152330390
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4267955615
C:\Users\Arc\git\BruteForceCrack\src>

C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56234
Server: StartKey ! ChunkSize ! CipherText ! EndKey ! KeySize ! StartTime
Server: 1 4276959300 ! 9003998 ! :+UHC88LxQEqKq6BmdGo31UtESHqIinLZssAZMKqSXXXIT7N
Jlc52Fng== ! 4294967296 ! 4 ! 177378155728500
Loading Search Parameters..
Parameters Loaded!
Performing Search..
4276959603
C:\Users\Arc\git\BruteForceCrack\src>
```

Key Found! Client reports a successful result to the server which prints a success message along with the elapsed time of the search. The manager then shuts down.

```
C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254

C:\Windows\system32\cmd.exe - java EchoServer 0 1 :+UHC88LxQEqKq6BmdG...
C:\Users\Arc\git\BruteForceCrack\src>java EchoServer 0 1 :+UHC88LxQEqKq6BmdGo31U
tESHqIinLZssAZMKqSXXXIT7NJlc52Fng==
Enter total Number of Clients:
4
Waiting for connections on 56254

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254

C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254
```

Forcing the search to fail - creating an instance where we enter inaccurate search parameters which will cause the search to fail.

```
C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254
Server: StartKey : ChunkSize : CipherText : EndKey : KeySize : StartTime
Server: 1 0 1 64 : :+UHC88LxQEGKq6BndGo31UeESHqIinLZssAZMKqSXXXI7NjLc52Fng== : 2
256 : 1 : 177738918157994
Loading Search Parameters..
Parameters Loaded!
Performing Search..
..
Search Finished!
No key found!
java.net.SocketException: Connection reset
C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254
Server: StartKey : ChunkSize : CipherText : EndKey : KeySize : StartTime
Server: 1 64 1 64 : :+UHC88LxQEGKq6BndGo31UeESHqIinLZssAZMKqSXXXI7NjLc52Fng== :
256 : 1 : 177748927118492
Loading Search Parameters..
Parameters Loaded!
Performing Search..
..
Search Finished!
No key found!
java.net.SocketException: Connection reset
C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
Client 4:
Client 4: Loading Client Search Parameters..
Client 4: Parameters Loaded!
Client 4: Performing Search..
Client 4: ..
Client 4: Search Finished!
Search Ended @ Key:564
Client 4: Search Finished!
Search Ended @ Key:128
Client 4: Search Finished!
Search Ended @ Key:192
Client 4: Search Finished!
Search Ended @ Key:256
Server: Key Space has been exhausted!
Server: Searched all possible solutions! - No work left to be done!
Total Search Time: 7173677435ms
Key was not found! -- Manager will shut down!
C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254
Server: StartKey : ChunkSize : CipherText : EndKey : KeySize : StartTime
Server: 1 128 1 64 : :+UHC88LxQEGKq6BndGo31UeESHqIinLZssAZMKqSXXXI7NjLc52Fng== :
256 : 1 : 177741278839952
Loading Search Parameters..
Parameters Loaded!
Performing Search..
..
Search Finished!
No key found!
java.net.SocketException: Connection reset
C:\Users\Arc\git\BruteForceCrack\src>

C:\Windows\system32\cmd.exe
C:\Users\Arc\git\BruteForceCrack\src>java EchoClient localhost 56254
Server: StartKey : ChunkSize : CipherText : EndKey : KeySize : StartTime
Server: 1 192 1 64 : :+UHC88LxQEGKq6BndGo31UeESHqIinLZssAZMKqSXXXI7NjLc52Fng== :
256 : 1 : 177742312415624
Loading Search Parameters..
Parameters Loaded!
Performing Search..
..
Search Finished!
No key found!
java.net.SocketException: Connection reset
C:\Users\Arc\git\BruteForceCrack\src>
```

Search has failed – no key found, manager shuts down.