# Documentation – Safety and Security

- RSA Algorithm

https://hackernoon.com/how-does-rsa-work-f44918df914b

- References for our program

The algorithm is made by receiving hex numbers, so just keep in mind that decryption and encryption function have as argument hex numbers converted in integers.

The private key is generated with multiplicative inverse algorithm:

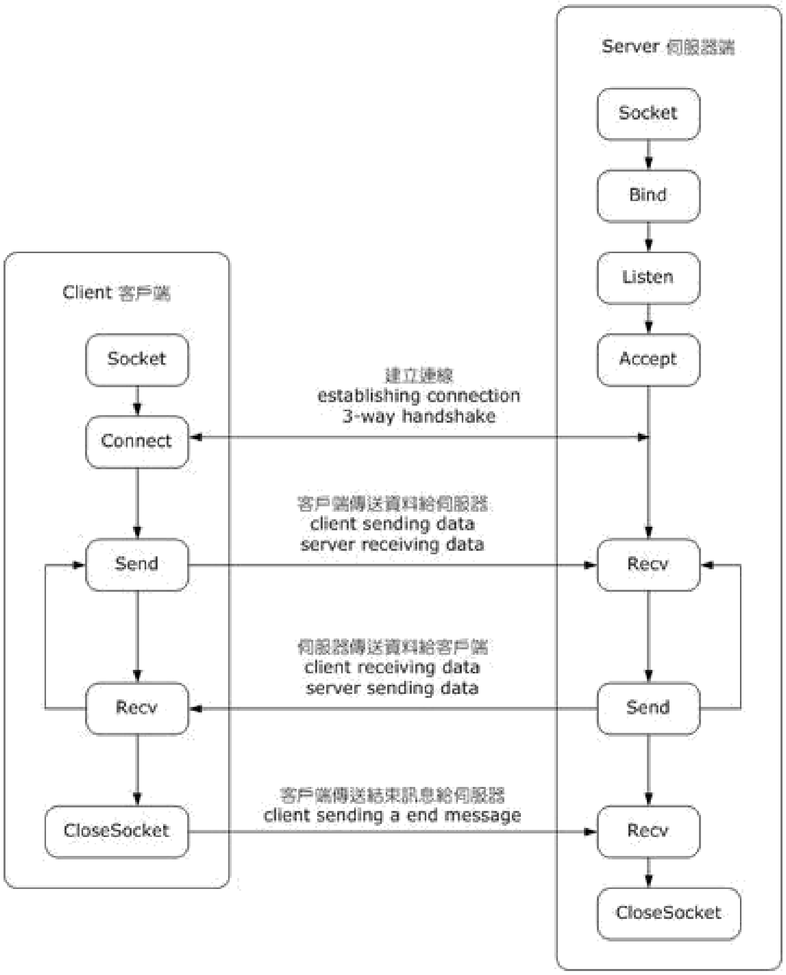
http://www-math.ucdenver.edu/~wcherowi/courses/m5410/exeucalg.html

- TCP Sockets

As you'll see, we'll create a socket object using socket.socket() and specify the socket type as socket.SOCK_STREAM. When you do that, the default protocol that's used in the Transmission Control Protocol (TCP). This is a good default and probably what you want.

Why should you use TCP? The Transmission Control Protocol (TCP):

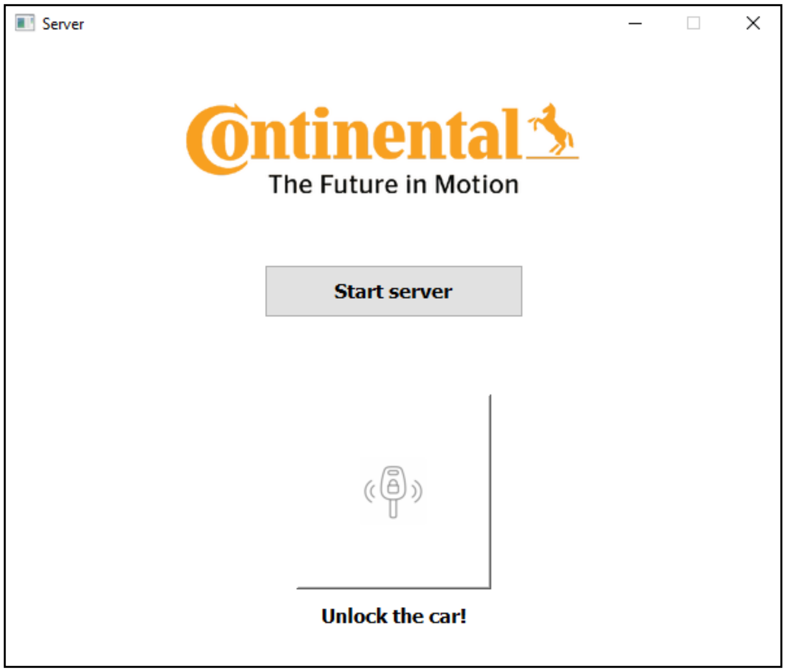
- **Is reliable:** packets dropped in the network are detected and retransmitted by the sender.
- **Has in-order data delivery:** data is read by your application in the order it was written by the sender.



TCP Socket Flow

- <u>Graphical interfaces</u>

  - Server interface

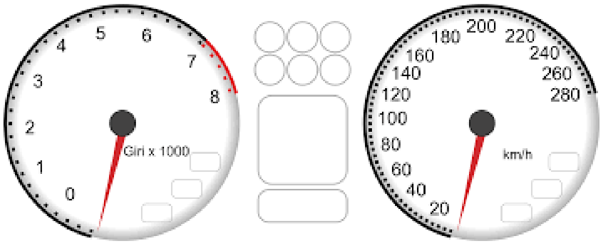


Start server starts the server and wait until the client is connected. When the client is connected a message Client connected will be displayed under the button.

The key button is initially disabled because there is no connection established, when a client is connected it becomes available.

When the key button is pressed, *"the car is unlocked"* and the dashboard is displayed on the screen.

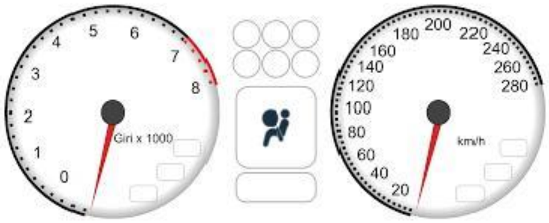
The dashboard is a warning zone, where specific errors will be displayed.



.

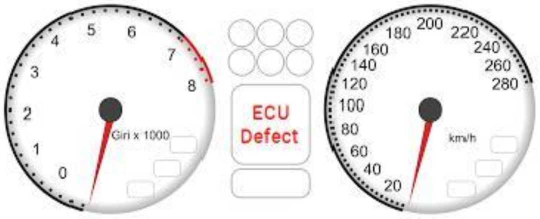
## Possible errors:

Airbag on – this appears when the client send a hex number that has the following format : LOW = 0x01, HIGH = ~LOW.

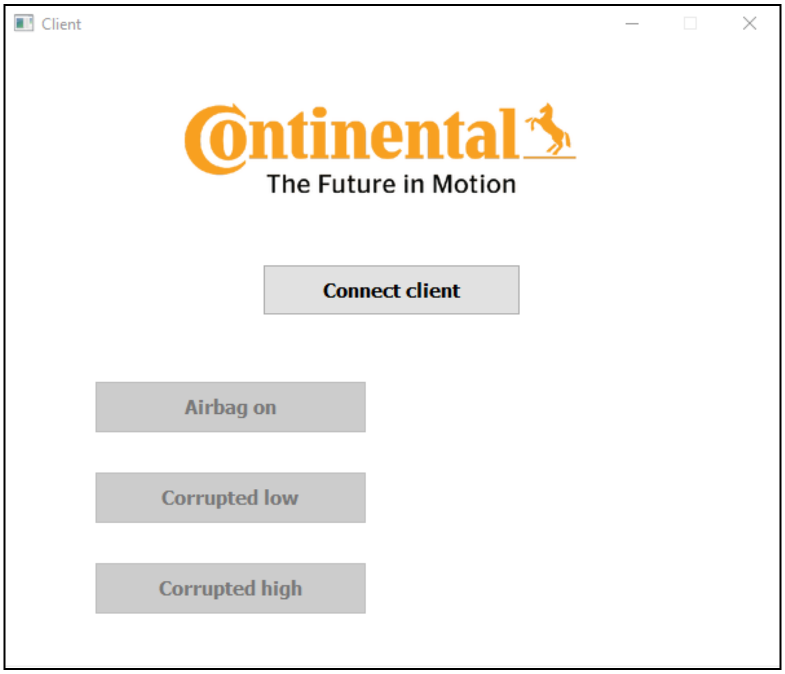


Low Corruption(LC)/High Corruption(HC) - this appears when the client send a hex number that doesn't have :

LOW != 0x01 (LC) and HIGH != ~LOW (HC)

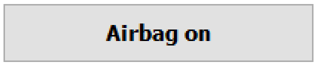


- Client Interface



Initially, the application has the buttons disabled, those become enabled after the key button is pressed on server interface.

Connect client establish the connection with the server, after the server responds a text Connected succesfully will be displayed.


Airbag on send messages with the following format:

LOW = 0x01 and HIGH = ~LOW. After the server execute the command a message Airbag on will be displayed.

Corrupted low send messages with the following format:
LOW = 0x57. After the server execute the command a Corrupted low message will be displayed.

Corrupted high send messages with the following format:

HIGH != ~LOW. After the server execute the command a Corrupted high message will be displayed.

- How the program works. Airbag ON simulation

A server is created and a client is connected to that server. The server is *"unlocking"* the car and the application can start. The client starts to send hex numbers to the server while it is waiting.

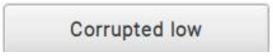
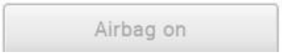
Airbag is started when the hex number has the following format:

- ➢ LOW = 0x01
- ➢ HIGH = ~LOW

When the is pressed, the client send a hex number (0xfe01). The number is encrypted using RSA Algorithm and after that sent to the server. The server receives the number, decrypt it, check if LOW = 0x01, if so check if HIGH = ~LOW. If all of those conditions are accomplished then the server starts the Airbag and sends a message back to the client.

When the number does not have the specific format, two possible warnings could appear:

- LOW != 0x01 – Low Corruption
- HIGH != ~LOW – High Corruption

If  / is pressed then  become disabled.

References :

https://commons.wikimedia.org/wiki/File:InternetSocketBasicDiagram_zhtw.png

https://hackernoon.com/how-does-rsa-work-f44918df914b

http://www-math.ucdenver.edu/~wcherowi/courses/m5410/exeucalg.html

https://realpython.com/python-sockets