

ELEC-C7420 Basic Principles in Networking

Part-II Security

Assignment I: Introduction to basic cryptography using Arduino

Hardware Required

- Arduino MKR WiFi 1010.

Circuit

- No special circuitry is required for this assignment.



Arduino MKR WiFi 1010

Details

1. Create a simple encryption sketch (Programs in Arduino IDE is called sketch). You are suggested to use Python but you can use any programming language or Arduino language (Set of C/C++ function).
2. The sketch tests communication between your terminal program and the Arduino board.
3. If you send a plain text message to the Arduino board (Preferably "Hello World") you get back an encrypted message.
4. If you send the encrypted message back to Arduino, you get the original plain text message back.
5. All you need to do is create a sketch that does any kinds of encryption to plain texts → compile and upload this sketch to the Arduino board → go to the serial monitor → type "Hello World" → show the encrypted text is there → type the encrypted text → see if you get "Hello World" back.

6. The Arduino Serial Monitor should work perfectly for this test. The putty terminal program also works fine. You can download putty from [here](#).
7. You can use any kinds of encryption schema you like, even use any kinds of encryption library (eg. crypto/base64, rot13 etc).

Please submit a written report.

Here is suggested template for the report-

- Section 1: Goals of the experiment (What is the purpose and the motivation behind the experiment), Connection diagram.
- Section 2 : Experimental Setup (Details of the Experiment step by step)
- Section 3 : Results & Conclusion (Please include snaps for each steps that contains proof of the successful experimentation)
- Section 4 : Annex (Please paste the sketch you created in this section)

Assessment Criteria (Total 8 Points):

- Creating successful compilation of sketch and upload without an error (2 Points)
- Successfully encrypt plain texts (3 Points)
- Successfully decrypt the encrypted text (3 Points)

Inspiration :

Below is a sample encryption program that you can take inspiration from (Full video tutorial can be found [here](#)). Please note that, the below sample code contains complex encryption method using rotate 13. The tasks given in this assignment are much simpler than this.

Testing procedure:

- Connect your USB to serial cable to the Arduino.
- Compile and upload this sketch to the Arduino board.
- Go to tools → serial monitor.
- You will see the Arduino is sending you the letter 'A'.
- Type in a plain text message and send it.
- The Arduino will reply with an encrypted message.

```
int inByte = 0;           // serial input and output character
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);    // initialize the serial port at 9600 baud
  while (!Serial) {
    ; // wait for serial port to connect
  } // wait for serial port to connect
```

```

    establishContact(); // wait for incoming data
} /* setup */
void loop() {
    // put your main code here, to run repeatedly:
    // This is the Rotate 13 encryption algorithm
    // If you run the algorithm twice, you get back the original
    message
    // Example:   ABC -> NOP -> ABC
    if (Serial.available() > 0) // if you have data input
    {
        inByte = Serial.read(); // read one byte of input
        // A to M get converted to N to Z
        if (inByte >= 'A' && inByte <= 'M')
        {
            inByte += 13;
        } /* if upper case A-M */
        // N to Z get converted to A to M
        else if (inByte >= 'N' && inByte <= 'Z')
        {
            inByte -= 13;
        } /* if upper case N-Z */
        // Lower case a to m get converted to n to z
        else if (inByte >= 'a' && inByte <= 'm')
        {
            inByte += 13;
        } /* if lower case a-m */
        // Lower case n to z get converted to a to m
        else if (inByte >= 'n' && inByte <= 'z')
        {
            inByte -= 13;
        } /* if lower case n-z */
    }
}

```

```
        Serial.write(inByte);    // write the encrypted character
back
        } // if Serial.available() > 0
} /* loop */
void establishContact()
{
    // write 'A' repeatedly until you receive data from the host
    while (Serial.available() <= 0)
        {
            Serial.print('A');    // write 'A' to the host
            delay(1000);          // this delay is optional
        } // while Serial.available() <= 0
    Serial.println();
} // establishContact()
```