COL-215

Mini-Project: Encryption-Decryption

REPORT

INPUTS:

Through Switches:

1. mode: Switch 15
    - Use: For Encrypt or Decrypt mode
2. decrypt_memory_mode: Switch 14
    - Use: For selecting the keys to be used to decrypt:
        - Original Key
        - Currently added key
3. shuffle: Switch 13 & Switch 12
    - Use: For selecting the type of shuffle
4. key: Switch 11 to Switch 0
    - Use: keys to be used for taking XOR

Clock:

5. clk: On-board clock
    - Use: For timing all the processes

Through buttons:

6. add_key: Down button
    - Use: For adding multiple keys
7. reset: Left button
    - Use: For resetting the memory of the board
8. key_reset: Middle button
    - Use: For resetting the stored keys
9. tx_start: Right button
    - Use: For transmitting the data stored in th e memory
10. show_key_button: Up button
    - Use: To show all the keys currently stored in the memory

Through USB:

11. rx_in:
    - Use: For sending data from gtkterm to board

OUTPUTS:

Through USB:

1. tx_out:
    a. Use: For sending data from board to gtkterm

For LEDs:

2. decrypt_memory_mode_LED: LED 14
    a. Use: To show which keys are being used to decrypt:
        i. One with which encryption was Current or
        ii. The one that was added previously
3. shuffle_out: LED 13 & LED 12
    a. Use: To shoe the shuffle type being used currently to encrypt/decrypt
4. show_key: LED 11 to LED 0
    a. Use: To show the keys that are being used for decryption.

For Seven segment display:

Use: For displaying E or d on the seven segment display to show the mode in which we are now

5. seg: For segments

1. an: For anodes

What we have achieved through this project?

(I) Things that we proposed in the project proposal :-

1. Key for stream cipher will be provided through eight switches that will represent the eight bits of the key.
2. In encrypt mode:
    a. The file is read into the memory using the serial receiver made in the Lab Assignments.
    b. Encryption is performed while transmitting the data into the file:-
        i. The eight bits of the byte stored in memory are shuffled using any one of the 4 shuffle operations available (shuffle operation can be chosen using two switch buttons) and then XOR is taken with the corresponding bits of the key – this becomes the encrypted byte.
        ii. The encrypted byte is transmitted to the file using the serial transmitter that was made in the Lab Assignments.
3. In decrypt mode:
    a. The file is read into the memory using the serial receiver made in the Lab Assignments.
    b. Decryption is performed while transmitting the data into the file:-
        i. The eight bits of the byte stored in memory are XORed with the corresponding bits of the key and then shuffled using the corresponding shuffle of used in the encryption part – this becomes the decrypted byte.
        ii. The decrypted byte is transmitted to the file using the serial transmitter that was made in the Lab Assignments.
4. Available shuffle operations:-
    a. Shuffle 0 (Switch buttons 00):
    While Encryption: bits 01234567 become 02461357
    Corresponding shuffle for Decryption: bits 01234567 become 04152637.

b. Shuffle 1 (Switch buttons 01):
   While Encryption: bits 01234567 become 45670123
   Corresponding shuffle for Decryption: bits 01234567 become 45670123.
c. Shuffle 2 (Switch buttons 10):
   While Encryption: bits 01234567 become 76543210
   Corresponding shuffle for Decryption: bits 01234567 become 76543210.
d. Shuffle 3 (Switch buttons 11):
   While Encryption: bits 01234567 become 23456701
   Corresponding shuffle for Decryption: bits 01234567 become 670123456.

We achieved all the above things that we had proposed before starting the project. Further we have added the following special functionalities to the project that we had earlier proposed.

(II) Special Functionalities

1. Advancement in Encryption/Decryption algorithm –
   In the algorithm proposed earlier we were doing encryption byte-by-byte (stream cypher) by first shuffling the 8-bits of the byte and then taking an XOR with a 8-bit key provided by the switches.
   In the final project we are carrying out the following steps while encryption and decryption:-
   Encryption:
   a. We are taking a variable number of 12-bit inputs as key from the user and storing it in memory.
   b. We are also taking a 2 bit input for shuffle select (as was proposed earlier).
   c. Based on the shuffle selection inputs first we shuffle the bits of the byte and then take an XOR with a key that is made using:
      i. The bytes index in the plain text and
      ii. The multiple 12-bit key inputs provided
   d. For the i^{th} byte of the plain text we first choose the (i mod number of keys added)th 12-bit key added. Then we select the 8-bit key that will be used to take XOR with the shuffled bits of the i^{th} byte has its 8-bits as follows:
      bit 7: ((i+7) mod 13)th bit of the 12-bit key chosen and similarly select other bits till
      bit 0: ((i+7) mod 13)th bit of the 12-bit key chosen.
   e. Then we take XOR with the key selected and transmit the encrypted byte.

   Decryption: The process of decryption is similar to that of encryption and we need to do the following things:
   a. Take XOR with the same key that was used while encryption. To get the same key we use the exact same procedure of key generation so that we get the same key.
   b. Then we using the shuffle select variables since we know the type of shuffle that would have been done while encryption we can do the corresponding shuffle to reverse the effect of the encryption time shuffle to get back the original byte as the decrypted byte which will be transmitted.

2. Variable number of keys – We can take any number of keys and use them for encryption and decryption. Also , each key is of 12 bits rather than the usual 8 bits. This makes it almost impossible to crack the encryption in reasonable time. Also, the bits undergo rotation to make it undetectable even when no keys are input.

3. Forgot key feature – the keys are being stored in a separate memory and with a button press, we can see all the stored keys on the leds in front of the switches. They are displayed sequentially for one second time for each key.

4. Additional decrypt memory – Used for testing that any other key wont work in decrypting the encrypted text. If the key inserted in the decrypting mode is the same, the decrypted text will be same, otherwise it will be junk. To test this, we can just change the mode while decrypting to use decrypting memory, with a slide switch.

5. Erasable key memory – The key memory can be erased and rewritten on will, without resetting the input, so that we can try another attempt to put a proper key if first input is wrong or incomplete. This is again highly helpful in testing.

Applications -

- Used to encrypt important and confidential files. This module can work with any key generator with any number of bits in key. Thus, it provides extremely high level of security in encryption.

- This can also act as a Password Protected file generator. We can give any file and input a password for that particular file and then save it. Then we can clear the key memory with a push button. The file will be stored as an encrypted file, while will be readable only when someone inputs the same password and then decodes it.