
# Fachhochschule Kiel

## Faculty of Computer Science and Electrical Engineering

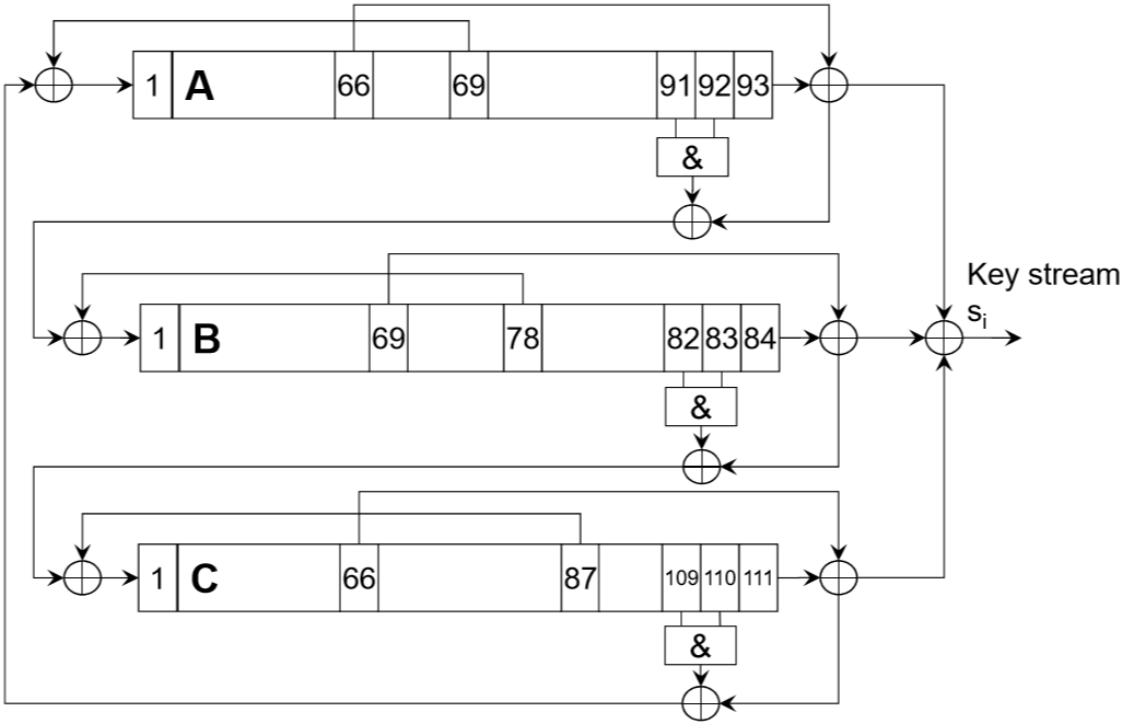
## Information Technology

## Advanced Cryptography (MK 105)

## Project Report
## on
## Trivium

**Subash Dawadi(926382)**

# 1. Introduction




Trivium is a relatively new stream cipher which uses an 80-bit key. It is based on a combination of three shift registers. Registers A, B and C are the heart of this registers. The length of the registers are 93,84 and 111 respectively. Specific feature of this cipher is that output of each register is connected to input of another register. Register are arranged in circle like fashion. Each register has feedback bit and feed forward bit. Each register two bits are performed AND operation. Final output is XOR sum of these bits. Table below shows all specification.

| | Register length | Feedback bit | Feedforward bit | AND inputs |
|---|---|---|---|---|
| A | 93 | 69 | 66 | 91,92 |
| B | 84 | 78 | 69 | 82,83 |
| C | 111 | 87 | 66 | 109,110 |

It has two input parameters a key K and Initialization Vector(IV). IV acts as a randomizer and takes new value for every session keys.

Initialization

Initially, 80 leftmost bit of register A is loaded with 80-bit Initialization vector and 80 leftmost bit of register B is loaded with 80-bit key. All other register bits are set to zero expect register C bit 109,110,111 which are set to 1.

Output of stream cipher is produced only after cycle of 1152.

## 2. Implementation in Matlab

I choose Matlab to implement this cipher into software. I had divided my coding into two phase. They are

a) Implementation

b) Operation

a) Implementation

In this phase, I implemented entire Trivium cipher into matlab code. At first I built register A,B,C and loaded with Initialization vector ,keys and required values and display as per requirement and then I finally implemented all total operation.

Description of register formation is given below

A=zeros(1,93); %% Register A of length 93 all assign to zero at first

B=zeros(1,84); %% Register B of length 84 all assign to zero at first

C=zeros(1,111); %% Register C of length 111 all assign to zero at first

At first all registers are assigned to zero value at first. For loading leftmost 80 bit of register A with Initialization vector, I choose randi command from matlab. Below code shown is for randi command.

# k=randi([0,1],[1,80])

randi command is used to generate random numbers. As Initialization vector servers as random number.

Similarly for key k , that is loaded into leftmost 80 bit of register B, I choose again randi command. For setting register C bit109, 110, 111 to one. I simply assigned 1 value to position of this register.

So I simply set Register A with Initialization vector IV at leftmost 80 bit. Similarly I set Register B leftmost 80 bit with key and 109,110,111 bit position of Register C with value 1 and all others register value are set to zero.

After then I focus on core implementation. So I loaded all build register into another function named “Operation”. Here I performed all operation as explained in above Introduction. For XOR operation I used “mod” command of matlab and for AND operation I used “bitand” command of matlab.

For easy understanding of code, I had simply comment of code which I have provided with this report.

b) Operation phase

In this phase, I run the entire code and check output. I simply check for output manually with two to three output stream bits. I found it correct at last.

## 3. Problem Faced

As we know coding is simply awesome if we don't encounter any bugs. But it is hectic if we encounter bugs. Same happened to me also. I simply faced a lot of problem in shift of register bits. Below is command for shift of bit position for register A.

```
for i=1:92
  K(1,i+1)=A(1,i);
  K(1,1)=t3;
end
 for i=1:93
   A(1,i)=K(1,i);
    End
```

Initially, I did

```
for i=1:92
  A(1,i+1)=A(1,i);
  A(1,1)=t3;
end
```

So it gave me a lot of problem. What was actually happened was that, I was shifting register bit within itself but not with added feedback value. I overcome this problem by simply doing shift of bits in another small code, observe result for many times and found the error.

## 4. Output

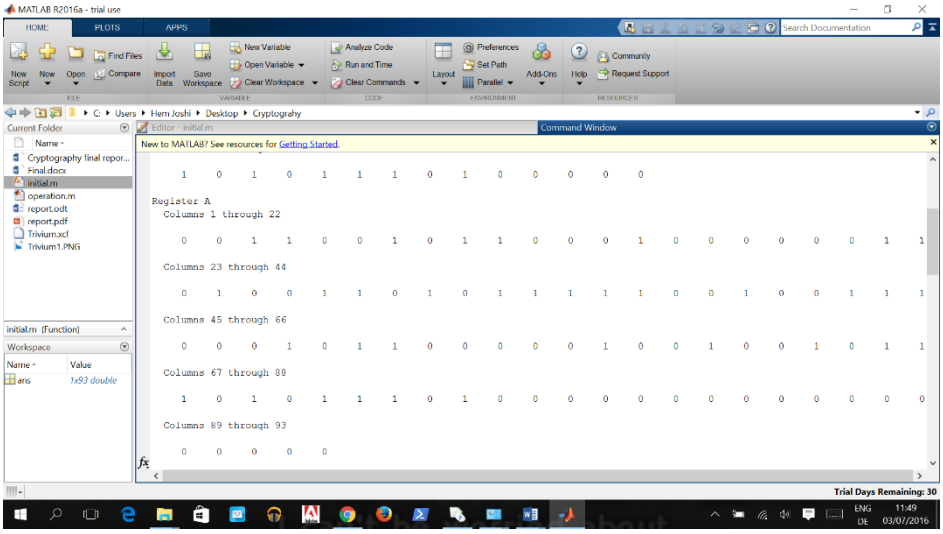



Figure 1: initial setup of Register A

```
Register B
  Columns 1 through 22

    0     0     1     0     1     1     1     0     0     0     1     1     0     1     1     1     1     0     0     1     1     0

  Columns 23 through 44

    0     0     0     0     0     1     0     1     1     1     1     1     0     1     0     1     1     0     0     0     1     1

  Columns 45 through 66

    0     1     1     0     1     0     1     0     0     0     0     1     0     1     0     0     0     1     0     0     0     1

  Columns 67 through 84

    0     1     0     1     0     1     0     0     0     1     1     1     0     1     0     0     0     0
```

Figure 2 : Initial setup of Register B

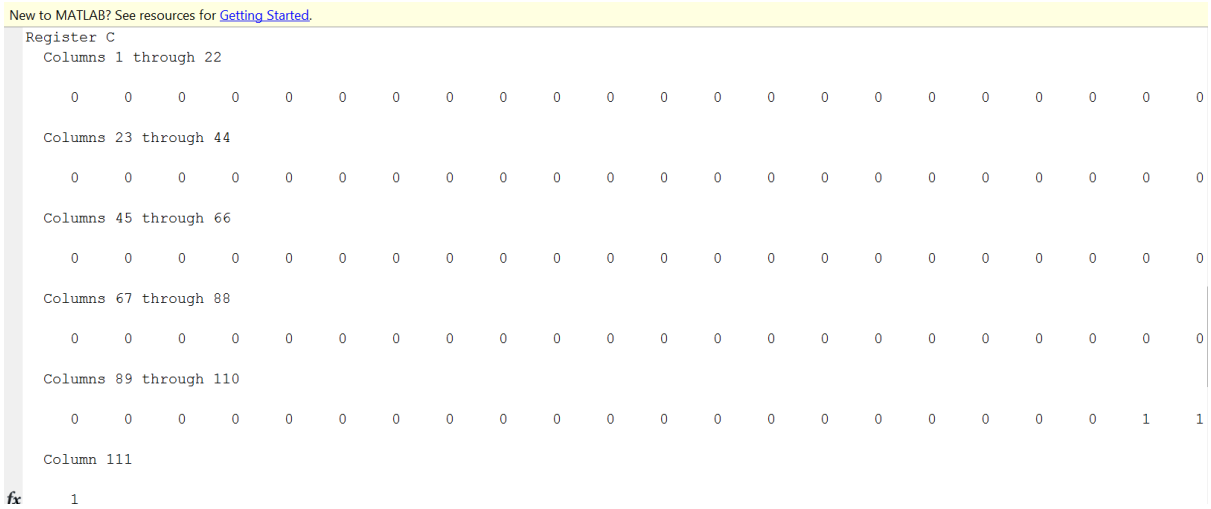



Figure 3: Initial setup of register 3

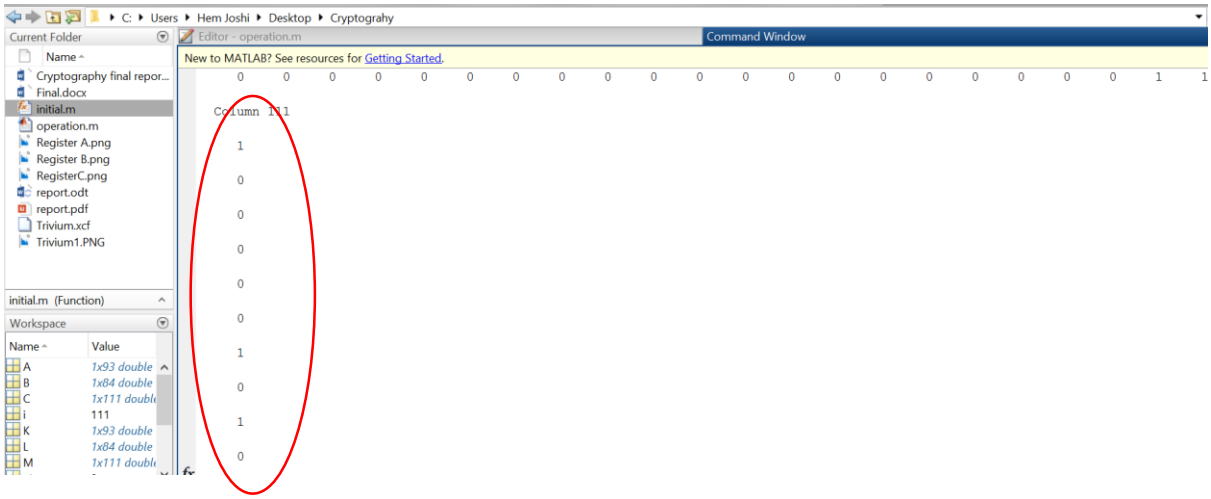



Figure 4: Final output of stream bits

The highlighted portion in red oval in above figure 4 is final output of stream cipher.

## 5. Conclusion

I was successful in implementing Trivium cipher in matlab and observed the result. I understood this cipher working mechanism very clearly.