Assignment No.1

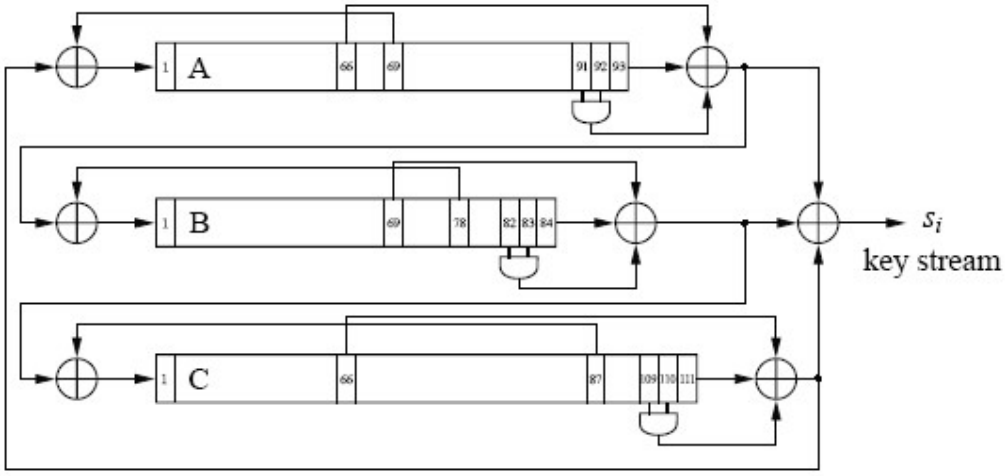
Submission Deadline: 10:30 AM Wednesday 21th October on Google Class Room

Total Marks: 100

NOTE: you can implement it any programming tool but code should be 100% original and this is an individual assignment

Implementation of Trivium Stream Cipher

Trivium is a relatively new stream cipher, which uses an 80-bit key. It is based on a combination of three Linear Feedback shift registers. Even though these are feedback shift registers, there are nonlinear components used to derive the output of each register, unlike the LFSRs that we studied in the class.



As shown in Figure above, at the heart of Trivium are three shift registers, *A*, *B* and *C*. The lengths of the registers are 93, 84 and 111, respectively. The XOR-sum of all three register outputs forms the key stream S_i. A specific feature of the cipher is that the output of each register is connected to the input of another register. Thus, the registers are arranged in circle-like fashion. The cipher can be viewed as consisting of one circular register with a total length of ***93 + 84 + 111 = 288***. Each of the three registers has similar structure as described below.

The input of each register is computed as the XOR-sum of two bits:

- The output bit of another register according to Figure above. For instance, the output of register *A* is part of the input of register *B*.
- One register bit at a specific location is fed back to the input. The positions are given in Table below. For instance, bit 68 of register *A* is fed back to its input.

The output of each register is computed as the XOR-sum of three bits:

- The rightmost register bit.
- One register bit at a specific location is fed forward to the output. The positions are given in Table below. For instance, bit 66 of register *A* is fed to its output.
- The output of a logical AND function whose input is two specific register bits. Again, the positions of the AND gate inputs are given in Table below.

	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

Note that the AND operation is equal to multiplication in modulo 2 arithmetic. If we multiply two unknowns, and the register contents are the unknowns that an attacker wants to recover, the resulting equations are no longer linear as they contain products of two unknowns. Thus, the feed forward paths involving the AND operation are crucial for the security of Trivium as they prevent attacks that exploit the linearity of the cipher, as the one applicable to plain LFSRs shown in the previous section.

For more detail, you can consult any source from internet or a textbook. However, you code should be 100 % original without any copying or plagiarism.

Implement Trivium and perform encryption and decryption using it.

Best of Luck ☺