INF 240 2019

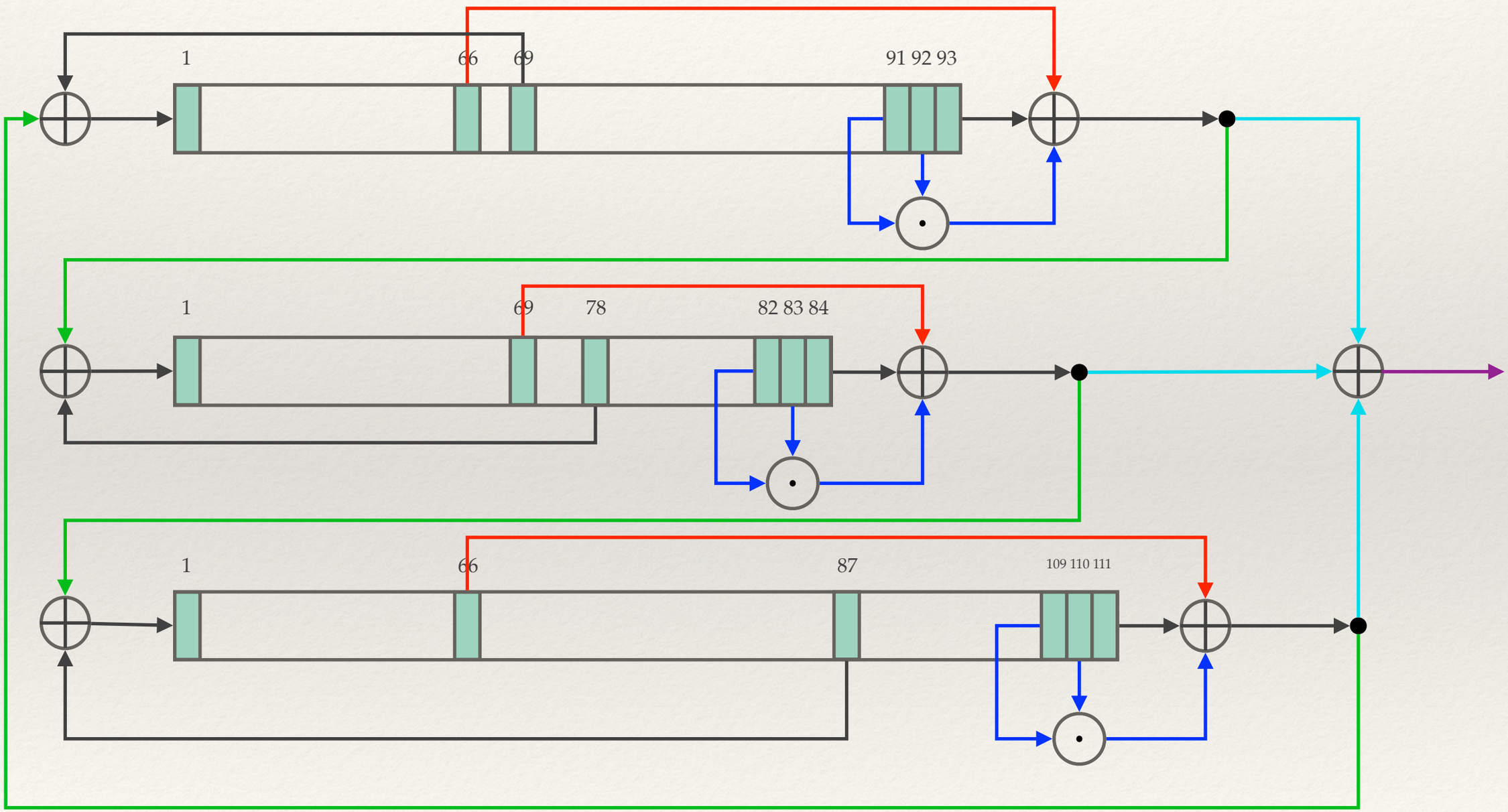
Trivium

Nikolay Kaleyski

The Trivium stream cipher

- Stream cipher designed by Christophe De Cannière and Bart Preneel from 2005
- The output of three LFSR's is combined to generate the keystream
- The LFSR's are interconnected, and non-linear components are added to them to improve security
- No efficient cryptanalytic attacks are known to date

Structure



Structure

- Trivium consists of three LFSR's of degrees 93, 84, and 111 for a total of 288 bits in the state
- Only one cell from each LFSR is connected to the feedback line, but it is combined (XOR-ed) with the output of another LFSR before replacing the bit at the left-most position
- Two bits of each LFSR are combined (AND-ed) and then XOR-ed with the output
- This is XOR-ed with another cell from the LFSR
- The results of the above steps for each LFSR are XOR-ed together and their value is the output at the current step

Initialization

- Most ciphers use both a *key* and an *initialisation vector* (IV) which does not need to be kept secret, but must change after every encryption so as to prevent known plaintext attacks
- Trivium uses an 80-bit IV which is loaded into the first 80 bits of the 93-degree LFSR
- An 80-bit key is loaded into the first 80 bits of the 84-degree LFSR
- The rightmost three bits of the 111-degree LFSR are set to 1
- All other bits are set to 0

Keystream generation

- Prior to encryption, the cipher performs 1152 = 4 x 288 iterations of "warm-up" without producing output
- The goal of the warm-up phase is to guarantee that the IV and key are sufficiently "mixed" together
- Starting with step number 1153, the keystream is generated by taking the output of the device