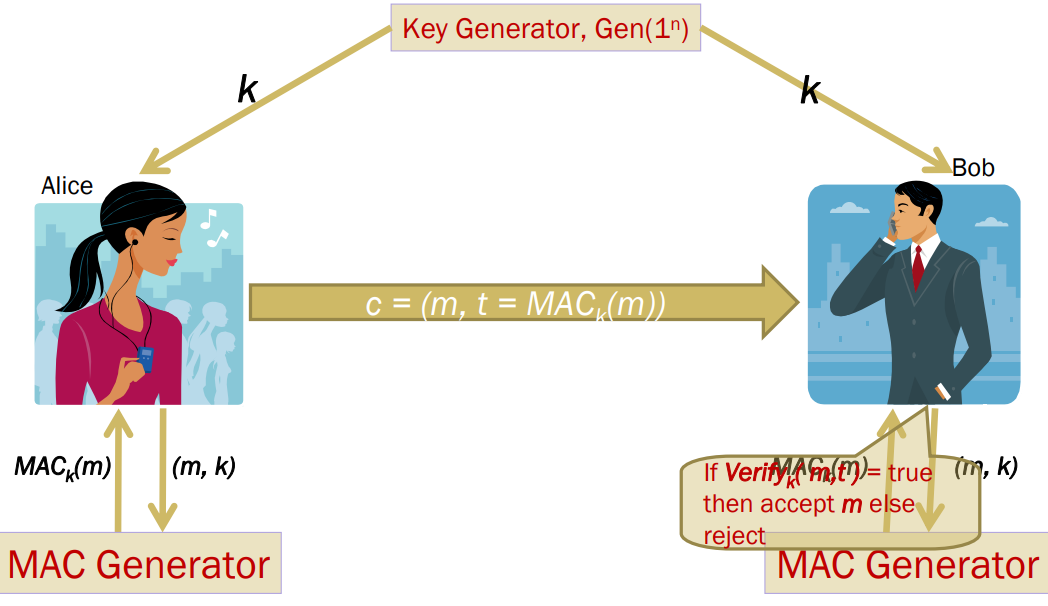
# Message Authentication Code (MAC)

## Theory:



* A Key Generation Algorithm that returns a secret key k
* A MAC generating algorithm that returns a tag for a given message m. Tag t = MACk (m)
* A Verification algorithm that returns a bit
* b = Verifyk (m1, t1), given a message m1 and a tag t1
* If the message is not modified then with high probability, the value of b is true otherwise false

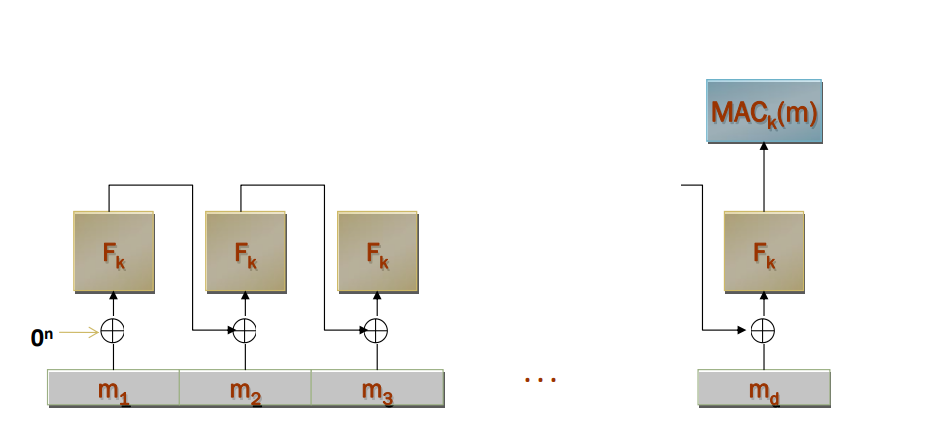
**Generating MAC:**

* Partition the message m to n sized blocks m1m2...mq
* Calculate MACk(m) = MACk(m1 ⊕ m2 ... ⊕ mq)

**Is this method secure?**

NO! We are authenticating the xor of the message blocks but not the message itself. So we can always choose a message whose xor value is the same as some other message.

**CBC-MAC:**



**Task:**

You are given 3 information: a message, key, and CBC-MAC signature. Your task is to verify whether the received message is valid or not.

| **Message** | **Key** | **MAC Signature** | **Validity** |
| --- | --- | --- | --- |
| I met an interesting turtle while the song on the radio blasted away | b'\x01\xd8i\xa1^0\x9a<\x0f\xf0\r\xc1\xdd\xd5\x89\xa6' | ba4ecb8db45c6ae0 |  |
| I like to leave work after my eight-hour tea-break | b'\xa6+\x16\x9d-1\xda\x8aV\xed\xf5\xf0cv\x04\x88' | f47e78c537fa1435 |  |
| Her daily goal was to improve on yesterday | b'[\xc5\xbd\xe4z\xd1=E\x17-ku\x02=|=' | ddaf3152edbe868a |  |
| He found the chocolate covered roaches quite tasty | b'5"k\xff\x81a\x9b 7\x8c>\xb7\xb9\xdcu\xaa' | 9d30d856f84489a8 |  |
| After fighting off the alligator, Brian still had to face the anaconda | b'\xa1\xfcw"?3\x91\x1c\t\x9c\x91\xe2He\x935' | b9d173e05bbf7738 |  |
| He decided to count all the sand on the beach as a hobby | b'\xa7\x83@\xde\xbf\xb494\xee\x84\x1e-\xc8A\xf9:' | 6355e471bd9930a1 |  |
| The sign said there was road work ahead so he decided to speed up | b'2\xcbv\xdcU6\x99\xb6.\xa7\xea\xeb\xaf\x10\xc7\x90' | 9fbafc75e0a5056a |  |
| Send 500$ to this account - 6589415651548 | b'\xc3\xea\x99e\xaal\xab\xd4\x9b\xf9\xb4Z\x19\xed\xcf\xcb' | 35273149636aca35 |  |
| Garlic ice-cream was her favorite | b'\x05\xf9\x83\x9d\xb7\xb6\xc3\xb8\x9e\xc5\xd9\xd8\x07]\xc6\xb3' | dc2de1e07b71d391 |  |
| I'd rather be a bird than a fish | b'\x84YY\xf0\x02GU\xa4LD\xd5\x85!A\xc2c' | 5e191d02aa5fc0b1 |  |

## Procedure:

Colab Notebook Link for this lab:

<https://colab.research.google.com/drive/1y0Za5ASOThcuahg7mxysdnd7QOEszlxj?>

1. Create a cmac object as shown using **key**
2. Update() the created object with your received message
3. Generate the MAC signature using finalize() function
4. Finally, print the decoded version of the signature and match it with your given signature.

# A5/1

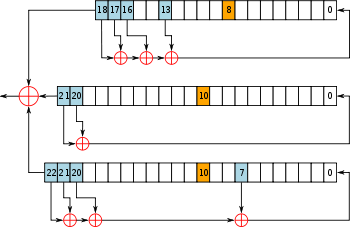
## Theory:

A5/1 consists of 3 shift registers.

X: 19 bits

Y: 22 bits

Z: 23 bits



## 

## Procedure:

Colab Notebook Link for this lab:

<https://colab.research.google.com/drive/1y0Za5ASOThcuahg7mxysdnd7QOEszlxj?>

1. **Complete** the encrypt() and decrypt() functions
2. Test your work by encrypting any plaintext and decrypting the found ciphertext

Encrypt the following plaintext:

| **Plaintext** | **64-Bit Key** | **Ciphertext** |
| --- | --- | --- |
| It is alive | 0011000001111111011110001011011010011010000001000011101101101011 |  |
| Snap out of it | 1110011010101010011101101010000001001100111011011000001011001010 |  |
| I am as mad as hell and I am not going to take this anymore | 0011101011001100010001111101110001110011001011011010100111001011 |  |
| Bond James Bond | 1101000000111100110011110000001100111001001100000010110011100111 |  |
| Love means never having to say you're sorry | 0111110000101011000010101111100111000001110000000001001101110110 |  |

Decrypt the following Ciphertext:

| **Ciphertext** | **64-Bit Key** | **Plaintext** |
| --- | --- | --- |
| 101100001100101111010100100100100010111010100001 | 1000000001111001000000111010110000111100100101001011001111000111 |  |
| 0110000101000001000000101110101100010100011001100111111100100011 | 1100010111111111101000001000110011010100100100111011001000001100 |  |
| 10100111010010010110101101110111 | 0010110010011001111001011100010100111010111011111000010100010010 |  |
| 100011110111100000010000111001000000110011111000101011111001010000100110101011101001010001011001111110101100100 | 0001101101111111100111011101010100000100111111100011010010111010 |  |
| 10101111111110011001010001000001111100100110000000110100110100001110010100010010 | 1110000000011000101110110000101011101101011100001110110110000100 |  |