**Module: CST4136 Cybersecurity & Cloud Systems**

**Lab Week: 06**

**Objectives:**

- Implement an XOR based stream cipher Method in MATLAB
- Implement a block cipher in MATLAB

## Implement XOR based stream cipher Method

- Check the RC4 cipher example provided in lab learning materials, experiment with it (explanation is here: https://www.youtube.com/watch?v=1UP56WM4ook )
- Design your own XOR based stream-based cipher module and implement, ready for final project.

**Guidance:** The provided example doesn't include a **nonce or IV**, so it always generates the same keystream for the same key. You can modify it to **add a nonce value** (e.g., mix it into the key setup) so each encryption run produces a different output. That simple change makes it your own stream cipher version.

## Implement a block cipher Method

- Example codes for **AES** and **DES** are provided in weekly lab learning materials. AES one is broken version.
- You can either fix the broken version or implement your own working version.
- **Choose only one algorithm — AES or DES**, whichever you prefer.
- Test your implementation with a sample plaintext and key, and show that encryption and decryption work correctly.

**By the end of the session, you should:**

- Understand stream cipher design. Have working stream cipher module.
- Understand DES/AES in practice. Have developed DES/AES modules from example.

**Resources:**

https://www.youtube.com/watch?v=1UP56WM4ook
https://uk.mathworks.com/matlabcentral/fileexchange/37847-data-encryption-standard-des
https://uk.mathworks.com/matlabcentral/fileexchange/53768-des-str-key-mode
http://freesourcecode.net/matlabprojects/59871/data-encryption-standard-%28des%29-in-matlab