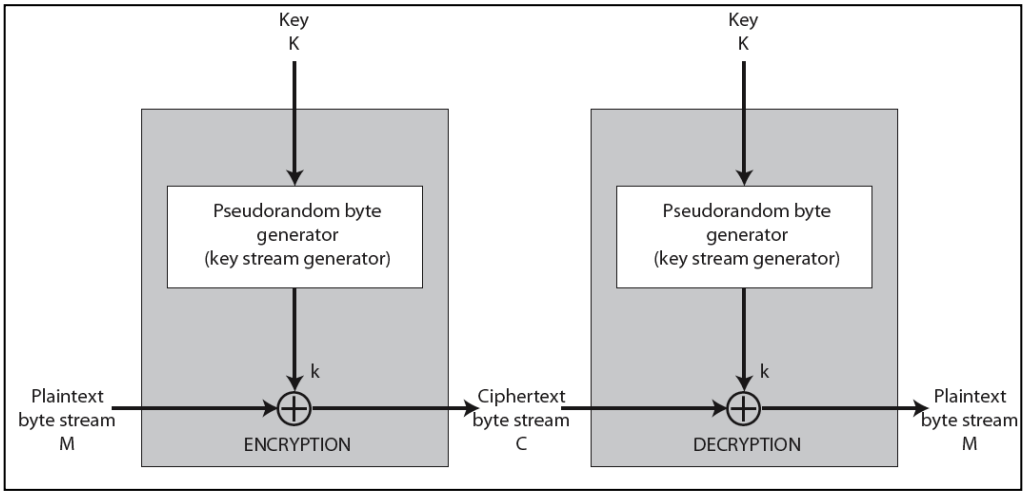
Cairo University
Faculty of Engineering
Computer Engineering Department

Spring 2025
CMP3050
(Computer Security)

Term Project - One Time Pad Stream cipher

- Project Objectives
  - Develop a strong and fast stream cipher using pseudo-random generators.
  - Ensure secure seed transmission using a strong symmetric key cipher.
  - Facilitate key exchange using Diffie-Hellman or ElGamal protocols.
  - Apply message authentication protocols for message integrity and authentication.
  - Analyze available design choices and select the best possible options.
  - Follow modular programming principles for clarity and maintainability.
  - Practice parallel programming communication.
  - Develop clear and detailed documentation skills.
- Introduction:

  This project allows you to explore cryptography by implementing a one-time pad stream cipher, which securely and quickly transmits sensitive data. As shown in the Figure below, this cipher generates a new random keystream for each transmission session, requiring identical keystream generation on both the sender's and receiver's sides. For the encryption and decryption process to work correctly, an initial seed exchange using secure encryption and authentication techniques is required. Through this project, you will gain experience with key cryptographic concepts such as pseudo-random generation, data encryption, authentication, and secure key exchange.



- Project modules
  - Stream Cipher for fast message encryption
    - The stream cipher will use the One-Time Pad (OTP) methodology, where the plaintext is XORed with a generated keystream based on a shared seed.
    - The plaintext is read from an input text file at the sender side and should be decrypted at the receiver side into an output text.
    - To speed up the project testing process, the transmission is done using a maximum number of 10 characters instead of one character per message.
    - Initial LCG parameters should be carefully selected to achieve a good pseudo random stream and these parameters are known constants to each side before transmission.

- Both sender and receiver must initialize with a new seed for each transmission session to ensure synchronization.
- The seed should be a long, proper random number generated at the sender side.

- Seed transmission
  - The seed should be encrypted using a strong asymmetric key cipher of your choice.
  - The asymmetric cipher's keys should be generated and exchanged using Diffie-Hellman or ElGamal Protocol.
  - In addition, a message authentication mechanism, like HMAC (Hash-based Message Authentication Code) should be used for ensuring message integrity and authenticity.

- Implementation Details
  - Project language
    - The project language is Python, and you are free to use any cryptography package provided that you understand in detail how it works.
  - Students should use "Modular Programming" i.e. divide the project into modules/functions like but not limited to:
    - Linear Congruential Generator (LCG) Module: generates the keystream.
    - Stream Cipher Module: Generates the keystream and performs XOR operations.
    - Seed Encryption Module: Manages seed encryption and decryption.
    - Key Exchange Module: Handles key generation and exchange.
    - Seed Authentication Module: Ensuring the message integrity and authenticity.
    - Communication Module: Manages data transmission between sender and receiver.
  - For each module or functionality of the requirement, students should print the input and output pairs of the function for verification.
  - The plaintext can be read from a text file, and the seed can be modified as required during the project delivery and discussion.
  - Provide one additional configuration text file to specify the parameters of the applied algorithms.
  - Print an output file at the receiver side for the decrypted plaintext.
- Project document
  - Provide a fully detailed report stating all the design choices, chosen algorithms, and print each detail of the functionalities of the project, including illustrative test cases.
  - Include the workload distribution between the team members in a table on the cover page.
  - Project documents will carry a considerable amount of the grade.
- Teams and delivery:
  - Work on pairs, and each student should participate in the code and the report.
  - Delivery and discussion will be on week 12.
  - Delivery details: upload one zipped file including your project code and report, and name the file with your team number like "1.zip". Send the zip file to email salma.cufece@gmail.com with subject "team number_one time pad project".
  - Plagiarism from a peer, the internet, LLMs, or any other resource will cause severe penalty.
  - The design choices are graded since they should be reasonable and justified towards achieving the project goals of implementing a strong and fast cipher.