Digital Signal Encryption/Decryption System

I. PROJECT DESCRIPTION

Design and implement an encryption/decryption system that reads a binary signal, convert the binary signal into a random analog signals (noise) distorting the content of the original signal data. The generated random signal will be encrypted/decrypted using the following crypto system (see Fig. 1).

![Encryption/Decryption System Diagram](image)

**Encoder/Decoder (64-bit binary-to-analog converter)**

The encoder/decoder (see Fig. 2) takes a binary string of 64-bits and maps each bit in the string into a unique 8-bits decimal value. The encoding technique is based on a randomly generated user’s profile. A user’s profile is established by (i) randomly selecting 64 integer values (8-bit decimal values from a set of 256 integer values) (ii) randomly picks one of the 64! available permutations and assigned it to both sender and receiver. A profile can be represented by the vector, Profile [64]. Output of the encoder will be composed of 64 values, each output value will represent a number that is taken from the profile vector and be polarized based on the bits contents of the encrypted binary string. For example, mapping \( b_i \rightarrow \text{Profile}[i] \) will be done as follows:

- If the content of bit \( b_i = 1 \), the output of the encoder will be the value (+Profile [i]).
- If \( b_0 = 0 \), the encoder will output (-Profile [0]).
Encryption/Decryption Algorithms:

You will be implementing a block cipher algorithm that is based on Tiny Encryption Algorithm (TEA). The plaintext is divided into equal sized blocks (64-bits). Each block is encrypted by applying TEA on the raw binary data block. The output of TEA (64-bits) will be used to initialized a 64-bit shift register R. The content of the register R is shifted to left by 5 bits using circular shift. The result of the shift operation will be XORed with (IV+i), where i represents the current block index. The decryption process is the inverse of the encryption function (see Fig. 3)
II. PROJECT REPORT AND SUBMISSION REQUIREMENTS (30 points)

- Write and submit a final report that outline your design. Describe all the classes that you design and developed for this project.
- Each team will require to submit one executable application. Your application must include all the .class files, executable files.
- Submit a hardcopy of the user manual and the project report
- In you report include graphs for the plaintext data (binary signal) and the encrypted data (analog signal)
- Encryption keys and Initialization vectors should be user input data.
- On the project due date. Each team will be asked to give a live demo during class time.
- No libraries or dependencies classes are allowed during the design and implementation for this work. All algorithms must be coded by the developers.

Reward points for your work:
- All teams with GUI implementations will get 5 rewards points. (optional)

III. PROJECT TEAMS:
- Team 1: Adeniji Adedamola, Calibuso Leonila Antonette, Cook Billy, Jaimes Edgar, Waldron Dylan
- Team 2: Bidwell Nicholas, Jordan John, Patel Jaykishan, Wissing Coby, Keplinger Matthew, Futch Lisa
- Team 3: Rana Krupa R, Rodgers Devin, Younker Jacob, Wilcox Cameron, McNeil Andre, Peoples Romeo

IV. GROUP MEETING (10 points)

Weekly meeting at the instructor office will be conducted with each group to monitor your progress.

Note: The authenticity of your work will be evaluated. Copying materials/code from online resources is prohibited and will not be tolerated.