ECG-based Encryption/Decryption System

I. PROJECT DESCRIPTION

Design and implement an encryption/decryption system that reads a file contains raw ECG data samples, quantized the ECG signal (convert the signal’s from analog to digital or vice versa using an 8-bit encoder/decoder), and then encrypt/decrypt the digital form of the ECG signal. Figure 1 depicted our ECG-based encryption/decryption system’s data processing blocks.

Quantizer (8-bit Encoder/Decoder)  (40 points)

The quantizer will read an ECG raw data signal and convert it from analog to digital. Each ECG pulse contains 104 data samples. You will be designing and implementing an 8-bit encoder/decoder that maps each analog sample in the ECG raw data into an 8-bit binary code. Each unique analog data sample must be mapped into a unique 8-bit binary code. In this project, you will be given a data file that contains 4 ECG pulses with a total of 416 samples. The data samples was collected from a single human subject (see Figure 2).
Encryption/Decryption Algorithms:
In this project, two encryption engines are integrated into the ECG-based crypto system:

I. **Asymmetric Encryption**

   **Knapsack Algorithm with ECB Mode**

   The Encryption/Decryption engine will be based on the Knapsack public/private asymmetric cipher scheme. Encryption and decryption will be performed on the ECG digital signal.

   - You will need to generate private and the public keys for the Knapsack crypto system
   - Private key will include SIK and a conversion factor
   - Public key will include the GK and a value \( n \)
   - Your scheme must be able to encrypt/decrypt 8-bit taken from the ECG-digital signal at once.

   For more detail see your Knapsack lecture slides

II. **Symmetric Encryption**

   **A5/1 Algorithm with Cipher-Feedback Mode**

   Block cipher can also be implemented as a self-synchronizing stream cipher; this called *Cipher-Feedback Mode* (CFB) mode. In CFB mode, data can be encrypted in units smaller than the block size. For Example, in our ECG-based encryption/decryption engine, each ECG pulse contains 13 blocks of 64-bit data (using an 8-bits code for each sample in the ECG-pulse). In this project, you will be implementing an 8-bit CFB mode working on a 64-bit block of ECG data. A block algorithm in CFB mode operates on a queue the size of the input block. Initially, the queue is filled with an IV, as in CBC mode. The queue is encrypted and the left-most eight-bits of the result are XORed with the first 8-bit ECG data code from the digital signal to become the first 8-bit cipher code. The same eight bits are also moved to the right-most eight bits positions of the queue, and all other bits move eight bit to the left. The eight left-most bits are discarded. To generate a 64-bit ciphertext block, CFB will take 8 rounds. The generated 64-bit ciphertext block \( B_{i-1} \) will be used to initialize the CFB’s queue of the next plaintext block \( B_i \). Decryption is the reverse of the above process. The encryption scheme is based on A5/1 (see Figure 3).
II. PROJECT REPORT AND SUBMISSION REQUIREMENTS

- 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 Java application. Your Java application must include all the .class files, executable file, and the .java files.
- Submit a hardcopy of the user manual and the project report
- In your report include graphs for the encrypted data
- 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 class are allowed during the design and the implementation for this works. All algorithms must be coded by the developers.

Reward points for your work:
- There will be a project contest among all teams. The team with the best GUI implementation will get 20 rewards points.
- Presenting your project at Techfest (April 14th) (10 points)

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

III. PROJECT TEAMS:
- Team 2: Brawner William A, Ericksen Steven, Newton Daniel J, Grant Darrell, Elias Tricia S
- Team 3: Ondriezek Shane, Bernitt Alden C, Mallard Preston, Chisholm Dajon S
- Team 4: McCarthy Tyler, Komula Dillon D, Gray Benjamin, Nelson Dylan, Bell Adrian J

IV. GROUP MEETING (10 points)
Weekly meeting at the instructor office will be conducted with each group to monitor your progress.