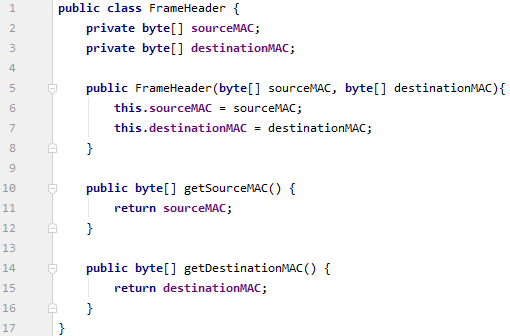
**БЕЗБЕДНОСТ НА КОМПЈУТЕРСКИ СИСТЕМИ**

**ЛАБОРАТОРИСКА ВЕЖБА 1**

Фисник Лимани, 151027

Во следните слики ќе ги прикажеме составните делови на кодот на првата лабораториска вежба и ќе го видиме/коментираме што им е нивната функција.



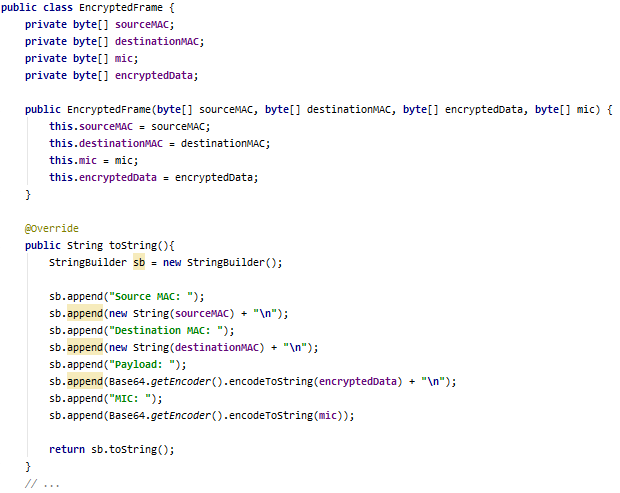
1. Класата **FrameHeader**



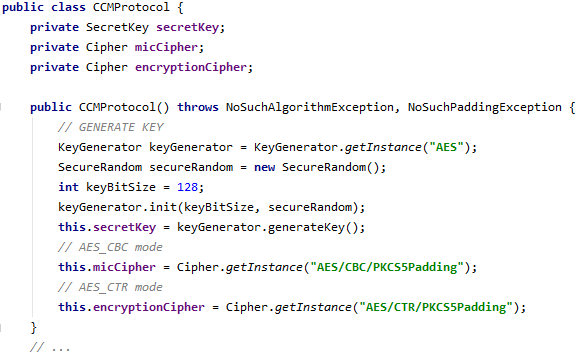
1. Класата **ClearTextFrame**

“// …” на крај на сликата значи дека оваа класа содржи уште некои функции кои ќе бидат прикажани подолу.

1. Класата **EncryptedFrame**



1. Класата **CCMProtocol**



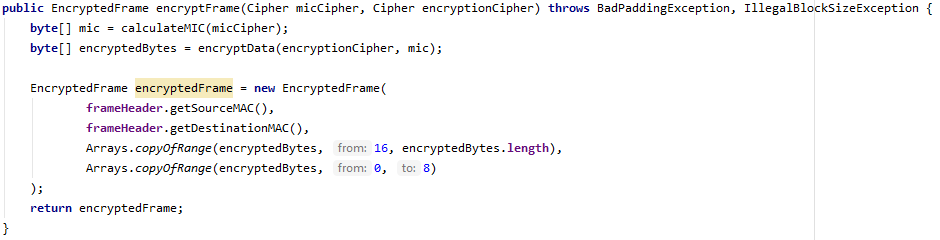
* CCMProtocol класата го симулира CCM протоколот.

Во оваа класа го изгенерираме:

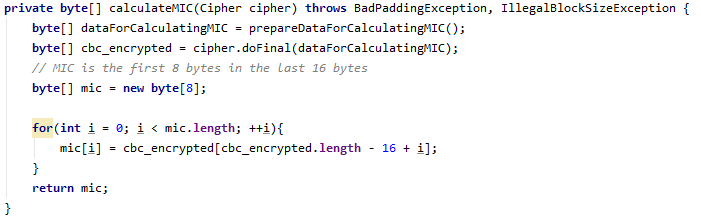
* + Клучот (Key-то)
  + AES/CBC шифрувачот
  + AES/CTR шифрувачот

1. Во класата ClearTextFrame ги имаме уште следните функции:

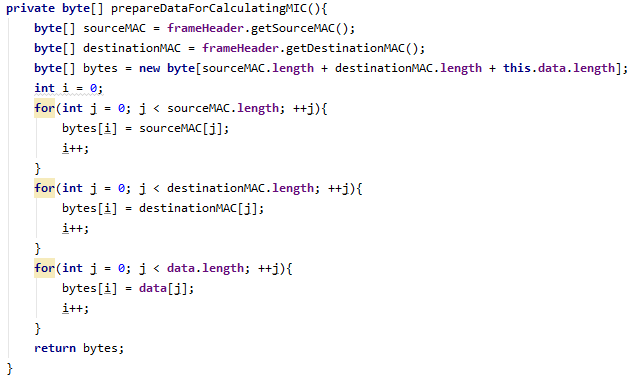
* **encryptFrame(Cipher micCipher, Cipher encryptionCipher)**
  + Оваа функција е централната функција во која се врши енкрипција на рамката
  + Прво се прави пресметување на MIC со посебна функција
  + Потоа се шифрира пораката



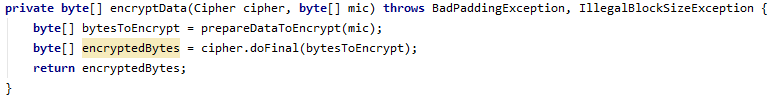
* **calculateMIC(Cipher cipher)**
  + Функција за пресметување на MIC (Message Integrity Check)



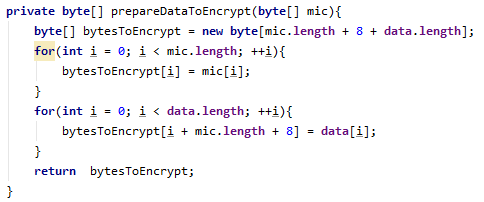
* **prepareDataForCalculatingMIC()**
  + Функција каде ги спремаме податоците за пресметување на MIC-от
  + Тука вршиме подредување на потребните податоци:
    - првите бајти ни се од изворниот MAC,
    - следните бајти ни се од дестинацискиот MAC,
    - и последните бајти ни се од пораката што сакаме да се испраќа



* **encryptData(Cipher cipher, byte[] mic)**
  + Централна функција за шифрирање на пораката што се испраќа



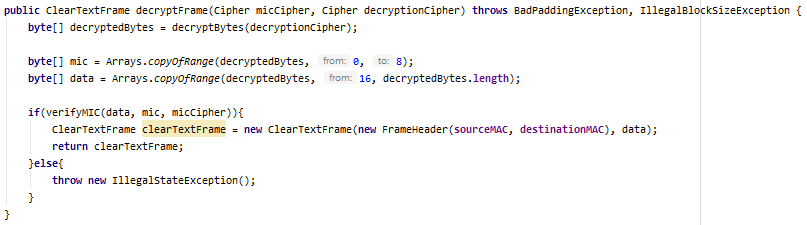
* **prepareDataToEncrypt()**
  + Функција за спремање на податоците за шифрирање
  + Првите 16 бајти ни се:
    - Првите 8 од MIC-от
    - Следните 8 може да бидат било какви бајти бидејќи ни се потребни само за да имаме еден блок од 128 битови (кои потоа ќе се игнорираат)
  + Следните бајти ни се од пораката што сакаме да се испраќа



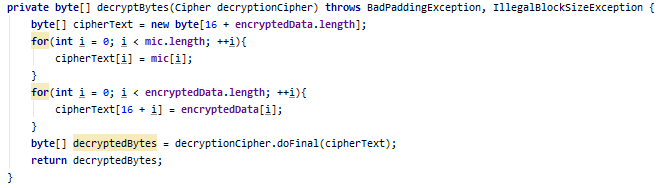
* + Го ставиме MICот како прв блок во низата од блокови што ќе се шифрираат бидејќи така ќе постигнеме MICот да се шифрира со CTR0, а потоа другите блокови кој ќе се креираат од пораката ќе се шифрират со: CTR1, CTR2, … CTRm.

1. Во класата **EncryptedFrame** ги имаме уште следните функции:

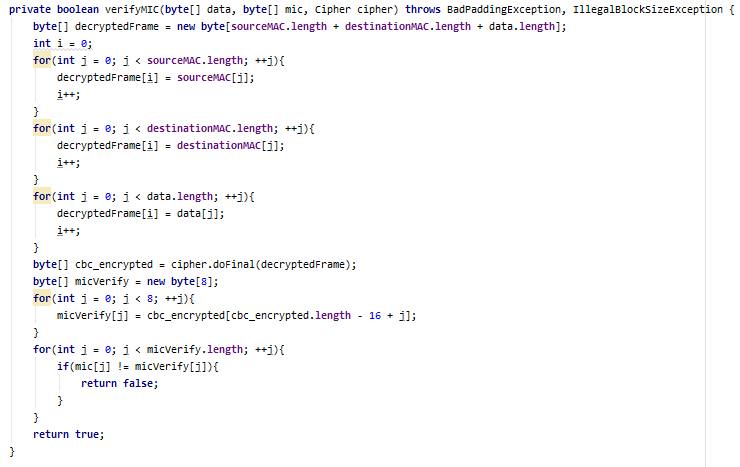
* **decryptFrame(Cipher micCipher, Cipher decryptCipher)**
  + Прво го дешифрираме рамката
  + Потоа го провериме MIC-от
    - Ако проверката на MIC-от поминува, тогаш се враќа инстанца од класата ClearTextFrame
    - Ако не, тогаш се фрли IllegalStateException исклучок



* **decryptBytes(Cipher decryptionCipher)**
  + Со оваа функција се прави дешифрирање на рамката

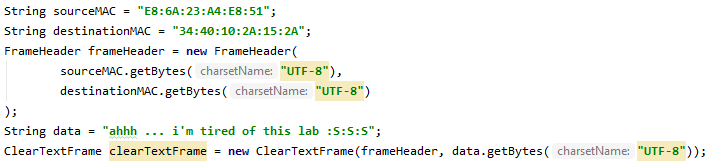


* **verifyMIC(byte[] data, byte[] mic, Cipher cipher)**
  + Со оваа функција се прави проверката за интегритетот на рамката



1. **ДЕМО:**Вредностите со кои ќе ги тестираме нашите функции ќе бидат:

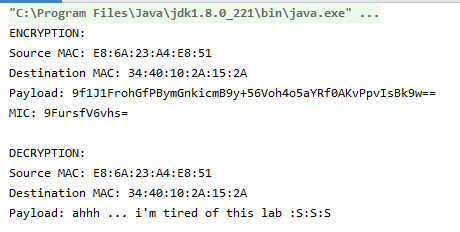
* Source MAC: **E8:6A:23:A4:E8:51**
* Destination MAC: **34:40:10:2A:15:2A**
* Data: **ahhh … i’m tired of this lab :S:S:S**



Main функцијата:



Output-от кога шифрирањето и дешифрирањето се извршуваат успешно:



Output-от ако дешифрирањето не успее:

