Segurança de Sistemas – Trabalho 2

Igor S. Brehm1

1 Escola Politécnica – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)  
Caixa Postal 1429 – 90619-900 – Porto Alegre – RS – Brazil

igor.brehm@edu.pucrs.br

**Abstract.** This paper aims to report to the reader the work done by this student in order to solve the problem scenario which was given by teacher Avelino Zorzo as the second evaluative task for the Security Systems discipline of the Software Engineering course at PUCRS university.

**Resumo.** Este relatório tem como objetivo relatar ao leitor o trabalho realizado por este aluno para solucionar o cenário problemático apresentado pelo professor Avelino Zorzo como segunda tarefa avaliadora da disciplina de Sistemas de Segurança do curso de Engenharia de Software da PUCRS.

# 1. Cifrar e Decifrar com AES, modo CBC com PKCS5

Os 16 primeiros bytes, tanto na cifragem quanto na decifragem, são considerados como sendo o IV. Na cifragem o IV é gerado randomicamente e adicionado na frente da mensagem cifrada a ser enviada. Uma biblioteca/dependência externa foi utilizada no código e mais informações estão disponíveis sobre ela no arquivo Readme.

## 1.1. Resultados Obtidos

1. CBC key: 140b41b22a29beb4061bda66b6747e14   
   CBC Ciphertext: 4ca00ff4c898d61e1edbf1800618fb2828a226d160dad07883d04e008a7897ee2e4b7465d5290d0c0e6c6822236e1daafb94ffe0c5da05d9476be028ad7c1d81  
   **Resultado**: Basic CBC mode encryption needs padding.
2. CBC key: 140b41b22a29beb4061bda66b6747e14   
   CBC Ciphertext: 5b68629feb8606f9a6667670b75b38a5b4832d0f26e1ab7da33249de7d4afc48e713ac646ace36e872ad5fb8a512428a6e21364b0c374df45503473c5242a253  
   **Resultado**: Our implementation uses rand. IV
3. CBC key: 140b41b22a29beb4061bda66b6747e14   
   CBC Plaintext: 4e657874205468757273646179206f6e65206f66207468652062657374207465616d7320696e2074686520776f726c642077696c6c2066616365206120626967206368616c6c656e676520696e20746865204c696265727461646f72657320646120416d6572696361204368616d70696f6e736869702e  
   **Resultado**: 

# 2. Cifrar e Decifrar com AES, modo CTR

Os 16 primeiros bytes, tanto na cifragem quanto na decifragem, foram novamente considerados como sendo o IV. Na cifragem o IV é gerado randomicamente e adicionado na frente da mensagem cifrada a ser enviada. Uma biblioteca/dependência externa foi utilizada no código e mais informações estão disponíveis sobre ela no arquivo Readme.

## 2.1. Resultados Obtidos

1. CTR key: 36f18357be4dbd77f050515c73fcf9f2   
   CTR Ciphertext: 69dda8455c7dd4254bf353b773304eec0ec7702330098ce7f7520d1cbbb20fc388d1b0adb5054dbd7370849dbf0b88d393f252e764f1f5f7ad97ef79d59ce29f5f51eeca32eabedd9afa9329  
   **Resultado**: CTR mode lets you build a stream cipher from a block cipher.
2. CTR key: 36f18357be4dbd77f050515c73fcf9f2   
   CTR Ciphertext: 770b80259ec33beb2561358a9f2dc617e46218c0a53cbeca695ae45faa8952aa0e311bde9d4e01726d3184c34451  
   **Resultado**: Always avoid the two time pad!
3. CTR key: 36f18357be4dbd77f050515c73fcf9f2   
   CTR Plaintext: 5468697320697320612073656e74656e636520746f20626520656e63727970746564207573696e67204145532061 6e6420435452206d6f64652e  
   **Resultado**: 0f416b123990ef705458a8d63d1a360fa94a7b03fafd6f6bd4b2f011a4394396fa908039e4e850b2b6d29587fcb0b129331c4cfca91e1b4cc711e9f722f27b20ff0aeec0cf47244256aae5337123703116471f037863db5212b13a6066c9de7007cb5f34aed61ac539a4bdd69b91447dc310950366c8271986a47ffc555d88d6d23886db

# References

1. Java Code Geeks “AES Encryption and Decryption in Java(CBC Mode)”, https://www.javacodegeeks.com/2018/03/aes-encryption-and-decryption-in-javacbc-mode.html, Abril.