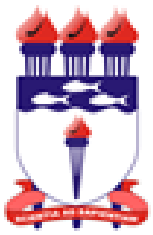

Universidade Federal de Alagoas
Instituto de Computação
Redes de Computadores
31 de agosto de 2021

Paloma Lacerda e Yanka Raíssa

Multithread Blowfish Server

1. Objetivo

Este trabalho teve por objetivo a construção de um sistema capaz de criptografar e descriptografar uma mensagem utilizando conceitos como socket, arquitetura de redes, protocolos de transporte entre outros assuntos pertinentes à disciplina. Foi construído inteiramente com a linguagem de programação Python (versão: 3.9)

2. Funcionalidades

MBS funciona utilizando a definição de conexão TCP, logo, para ser executado como planejado, deve-se primeiramente ativar um servidor no terminal e só então os clientes devem enviar suas requisições, sendo possível escolher qual porta e/ou host será utilizada durante a conexão do servidor, escrevendo '--port ou '--host', respectivamente. Ambos os parâmetros deverão ser passados no momento da execução do código. Quando a conexão é aceita, será solicitado ao cliente que informe qual ação deseja realizar enquanto o servidor escuta na porta.

No menu há 3 opções: Criptografar, Descriptografar e Sair. Escolhendo a primeira opção, o servidor irá solicitar a mensagem a ser codificada e em seguida uma chave de no mínimo 4 caracteres. Quando recebe essas informações, o servidor então utiliza o algoritmo de criptografía (módulo Pycryptodome, técnica blowfish) e retorna a mensagem criptografada.

Escolhendo a opção 2, será solicitado uma mensagem criptografada e em seguida a chave que foi usada para codificá-la. Caso seja identificado um erro no procedimento, existirá 2 possíveis causas para o ocorrido, primeiro: a sua chave pode está errada ou segundo: sua mensagem pode está em uma base incorreta (a base padrão deverá ser a de 64 bits). Caso tudo esteja correto, o servidor então realizará o procedimento inverso da opção 1 e devolverá a mensagem legível.

E por fim, a opção 3 encerra a conexão do socket.

3. Processo de desenvolvimento e futuras implementações

Para propósitos meramente estéticos, utilizamos o módulo Rich para personalização do terminal. As cores e as barras de progresso foram feitas utilizando este módulo.

Como o código foi desenvolvido majoritariamente no Linux, algumas dependências do Rich podem não funcionar em outros sistemas operacionais, e aí está uma possível melhoria, tornar o projeto multiplataforma.

Como todo projeto, surgiram alguns empecilhos quanto o funcionamento correto do algoritmo, entre eles podemos citar o bug da opção 2, no qual não podíamos digitar uma mensagem criptografada que já não tivesse sido codificada pelo sistema anteriormente, ou seja, só podia selecionar a opção 2 depois de passar pela 1.

Link do repositório

Palavras-chave: criptografia; blowfish; servidor; cliente-servidor; python;