TRABALHO CRIPTOGRAFIA E SEGURANÇA

Nomes: Leonardo Osvald de Souza / Lucas Amaral
RAs: 1121661 / 1117265

Relatório técnico:

- Tecnologias e algoritmos utilizados: Para o chat foi utilizado um console app com .NET 6 para o cliente e outro console para server.
  Para rodar o projeto é necessário SDK do .net 6.

  Server side: rodar o projeto utilizando o comando dotnet run - - - -server, onde ele irá subir um console que receberá as mensagens encriptadas e fará a descriptografia da mesma utilizando AES (simétrico) e o algoritmo SHA-256(HMAC), que é recomendado para mensagens. Para rodar o Client Side é preciso rodar o comando dotnet run (sem os args) no mesmo caminho do projeto.

- Descrição do funcionamento do sistema: Ao subir os 2 consoles, um lado irá receber o texto da mensagem e criptografá-la e em seguida enviar a mensagem através de um Network stream onde o console com o Server side estará fazendo a a leitura desse stream em um intervalo de 100ms (utilizando Thread.Sleep)

```
while (true)
{
    if (Console.KeyAvailable)
    {
        // Envia mensagem
        Console.Write("Você: ");
        var msg = Console.ReadLine() ?? "";

        aes.GenerateIV();
        var encrypted = Encrypt(msg, aes, hmacKey);

        // Exibe a mensagem criptografada (em Base64 para legibilidade)
        Console.WriteLine($"[INTERCEPTADO] Mensagem criptografada (Base64): {Convert.ToBase64String(encrypted)}");

        stream.Write(encrypted, 0, encrypted.Length);
    }

    if (stream.DataAvailable)
    {
        // Recebe mensagem
        var buffer = new byte[1024];
        var bytesRead = stream.Read(buffer, 0, buffer.Length);

        // Exibe a mensagem criptografada (em Base64 para legibilidade)
        Console.WriteLine($"[INTERCEPTADO] Mensagem criptografada (Base64): {Convert.ToBase64String(buffer[..bytesRead])}");


        var decrypted = Decrypt(buffer[..bytesRead], aes, hmacKey);
        Console.WriteLine($"Outro: {decrypted}");
    }

    Thread.Sleep(100);
}
```

Para fins de visualização, foi incluída a visualização da mensagem criptografada tanto no lado do cliente quanto do servidor.

A chave HMAC é gerada através de um número de 32 bits que é publicado no stream, para tanto o cliente quanto o servidor .
Agora que os 2 lados possuem a chave, é feita criptografia/descriptografia das mensagens.

- Escolha das tecnologias: por se tratar de um chat simples, para fins de aprendizado acadêmico, não achamos necessário utilizar algoritmos de chave assimétrica por ex. Os algoritmos selecionados são extremamente seguros e até hoje não foram quebrados, apesar da simplicidade. Visando também a simplicidade foi escolhido um console APP em .NET, que funciona como o esperado e atende o propósito do trabalho.