UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

ENTREGA 5

Documentação do Protocolo

IF837 - Protocolos de Comunicação

Nicholas Henrique Justino Ferreira
Rafael Sutil Pereira

RECIFE, 30 DE JULHO DE 2021
Professor: Divanilson Rodrigo de Sousa Campelo

Namespace VoteProtocol

Fornece serviços a fim de manter uma comunicação segura entre cliente e servidor para que nenhum invasor em uma rede WiFi consiga decifrar, alterar ou retransmitir dados capturados. Este Namespace é constituído por duas classes, uma para ser usada com o servidor (PServer) e outra para ser usada com o cliente (PClient).

Classe PServer

Representa uma instância do servidor com métodos e propriedades que auxiliam uma comunicação segura.

Handshake

Método estático e público que realiza a troca de certificados entre cliente e servidor, calcula e envia a chave secreta a fim de gerar o HMAC das mensagens. Este método não possui retorno.

Parâmetros

Socket current: Soquete do cliente no qual desejamos fazer a troca de mensagens.

PackMessage

Método estático e público que empacota uma mensagem adicionando o número de sequência, tipo e um número randômico, criptografando e concatenando com o HMAC da mensagem criptografada.

Parâmetros

string message: A mensagem que deve ser empacotada.
bool endMessage: Indica se o pacote é de encerramento de conexão.

Retorno

Retorna o pacote em um vetor de bytes pronto para ser transmitido com segurança.

Comentários

O tipo que será a mensagem é apenas para informar se é uma mensagem de encerramento ou não, ele foi adicionado para evitar possíveis *truncation attacks.*

UnPackMessage

Método estático e público que desempacota a mensagem recebida do cliente, verifica a integridade comparando o HMAC com o gerado localmente, verifica se o número de sequência é o esperado e o atualiza.

Parâmetros

byte[] package: Pacote a ser desempacotado.

Retornos

Retorna a mensagem descriptografada sem o número de sequência, sem o tipo e sem o HMAC. Caso o HMAC ou o número de sequência não coincida, o retorno será uma string "ERROR".

ReceiveResponseBytes

Método estático e privado que recebe um vetor de Bytes de um Socket conectado.

Parâmetros

Socket socket: Soquete do cliente que vamos receber os bytes

Retorno

Retorna os bytes recebidos.

Comentários

Este método, por ser privado, só será usado para receber o certificado do cliente no Handshake.

Classe PClient

Representa uma instância do cliente com métodos e propriedades que auxiliam uma comunicação segura entre cliente e servidor.

Handshake

Método público que inicia a comunicação com o servidor a fim de realizar a troca de certificados e receber a chave secreta para gerar o HMAC das mensagens. Não possui parâmetros e retorno.

Exceções

Exception Handshake Error: Esta exceção pode ser lançada por diversos erros que possam acontecer dentro do Handshake, dentre os possíveis erros temos a falha em tentar converter os bytes recebidos do servidor em um certificado, falha em extrair a chave pública deste certificado e falha na tentativa de descriptografar a mensagem que contém a SecretKey.

CertValidate

Método estático e privado que checa se o certificado recebido do servidor é válido. Não possui retorno, apenas lança exceções e desconecta o cliente caso haja erros.

Parâmetros

X509Certificate2 certificate: Certificado recebido do servidor.

Exceções

ArgumentNullException certificate: Esta exceção é lançada quando o certificado possui um valor nulo.
Exception Certificate was not issued by a trusted issuer: Esta exceção é lançada quando o certificado enviado não foi emitido pelo ElectionServer.

SendString

Método privado usado no Handshake para dar início a troca de mensagens. Não possui retorno.

Parâmetros

string text: Mensagem a ser codificada e enviada sem nenhuma proteção.

ReceiveResponseBytes

Método privado que é usado exclusivamente para receber o certificado do servidor. Não possui parâmetros.

Retornos

Retorna um vetor de 2048 bytes correspondente ao certificado do servidor.

ReceiveResponseKey

Método privado que é usado exclusivamente para receber a chave secreta do servidor no Handshake. Não possui parâmetros.

Retornos

Retorna um vetor de 288 bytes que representa a chave secreta.

ReceiveResponsePackage

Método público usado para receber o pacote que foi enviado pelo servidor. Não possui parâmetros.

Retornos

Retorna um vetor de 288 bytes que representa o pacote, 256 bytes para o mensagem criptografada e 32 bytes do HMAC.

PackMessage

Método público que empacota uma mensagem adicionando o número de sequência e um número randômico, criptografando e concatenando com o HMAC da mensagem criptografada.

Parâmetros

string message: A mensagem que deve ser empacotada.

Retornos

Retorna o pacote em um vetor de bytes pronto para ser transmitido com segurança.

UnPackMessage

Método público que desempacota a mensagem recebida do servidor, verifica a integridade comparando o HMAC com o gerado localmente, verifica se o número de sequência é o esperado e o atualiza.

Parâmetros

byte[] package: Pacote a ser desempacotado

Retornos

Retorna a mensagem descriptografada sem o número de sequência, sem o tipo e sem o HMAC.

Exceções

Caso o HMAC não coincida, o retorno será uma string "Message without integrity". Caso o número de sequência não coincida, o retorno será uma string "Sequence number does not match".

CalculateSHA256

Método público que calcula o valor da função SHA256 sobre um string.

Parâmetros

String value: string a ser codificada em SHA256.

Retornos

Sb.ToString(): string do valor SHA256 calculado