Relatório Técnico: Funcionalidade dos Arquivos Python

1. Introdução

Este relatório fornece uma análise detalhada dos arquivos Python fornecidos: server.py, gpu_crypto.py, e client.py. Eles implementam uma comunicação segura entre cliente e servidor utilizando criptografia AES (Advanced Encryption Standard).

Relatório Técnico: Funcionalidade dos Arquivos Python

2. Descrição dos Arquivos

Este arquivo implementa um servidor que aceita conexões de até dois clientes simultâneos. O servidor gerencia a comunicação entre os clientes, encaminhando mensagens criptografadas de um cliente para outro.

- Inicialização do Servidor:

O servidor é configurado para escutar na porta 12345 no endereço IP local (127.0.0.1). Ele aceita até duas conexões de clientes.

- Gerenciamento de Clientes:

O servidor recebe o nome do cliente e armazena o socket de conexão em um dicionário. Ele envia uma chave de sessão aleatória de 32 bytes para o cliente.

- Comunicação:

O servidor recebe o tamanho e o conteúdo das mensagens criptografadas dos clientes, decifra e reencaminha as mensagens para outros clientes conectados.

Relatório Técnico: Funcionalidade dos Arquivos Python

Este arquivo contém a classe GPUCrypto responsável pela criptografia e descriptografia das mensagens utilizando AES em modo CBC (Cipher Block Chaining).

- Inicialização:

A classe é inicializada com uma chave de criptografia.

- Criptografia:

O método encrypt adiciona padding aos dados, gera um vetor de inicialização (IV), e realiza a criptografia.

- Descriptografia:

O método decrypt separa o IV do texto cifrado, decifra os dados e remove o padding.

Relatório Técnico: Funcionalidade dos Arquivos Python

Este arquivo implementa um cliente que se conecta ao servidor, envia e recebe mensagens criptografadas.

- Inicialização do Cliente:

  O cliente se conecta ao servidor, envia seu nome e recebe a chave de sessão.

- Recebimento de Mensagens:

  O cliente escuta continuamente por novas mensagens do servidor, decifra e as exibe.

- Envio de Mensagens:

  O cliente lê mensagens do usuário, as cifra e as envia ao servidor.

Relatório Técnico: Funcionalidade dos Arquivos Python

3. Fluxo de Dados entre Cliente e Servidor

1. Conexão:

- O cliente se conecta ao servidor e envia seu nome.
- O servidor aceita a conexão e envia uma chave de sessão.

2. Comunicação:

- O cliente cifra a mensagem usando a chave de sessão e a envia ao servidor.
- O servidor recebe a mensagem cifrada e a reencaminha para o outro cliente.
- O cliente destinatário recebe a mensagem cifrada e a decifra.

Relatório Técnico: Funcionalidade dos Arquivos Python

4. Segurança e Criptografia

A comunicação entre cliente e servidor é protegida utilizando criptografia AES em modo CBC. A chave de sessão de 32 bytes é compartilhada entre o servidor e os clientes no início da conexão, garantindo a confidencialidade das mensagens trocadas. O uso de um vetor de inicialização (IV) diferente para cada mensagem aumenta a segurança, evitando padrões previsíveis.

Relatório Técnico: Funcionalidade dos Arquivos Python

5. Conclusão

Os arquivos fornecidos implementam uma comunicação segura entre cliente e servidor, utilizando criptografia AES para proteger as mensagens. O servidor gerencia a troca de mensagens criptografadas entre os clientes, garantindo que apenas os destinatários possam decifrá-las. A abordagem utilizada oferece uma camada robusta de segurança para a troca de informações.