Relatório Grupo 66

Introdução:

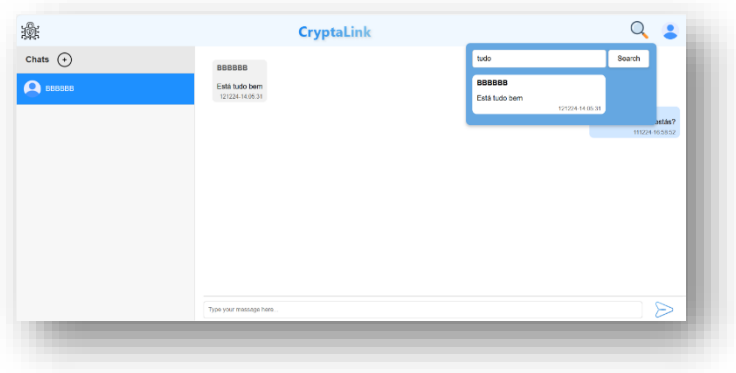
Este projeto tem como objetivo desenvolver uma aplicação de mensagens peer-to-peer (P2P) segura, que permite a comunicação direta entre utilizadores, garantindo a privacidade e a integridade das mensagens. Inspirada em plataformas como WhatsApp e Signal, a aplicação foi desenhada para oferecer uma experiência de comunicação descentralizada e segura, onde apenas o remetente e o destinatário têm acesso ao conteúdo das mensagens, protegendo-as contra interceções e modificações

A aplicação foi desenvolvida em Java e utiliza comunicação baseada em SSL Sockets para estabelecer uma camada de segurança sobre o protocolo de transporte. Com SSL (Secure Sockets Layer), é possível assegurar que as mensagens trocadas entre os utilizadores estão encriptadas e protegidas contra acessos não autorizados. Esta abordagem permite que a aplicação forneça criptografia de ponta a ponta, autenticação dos utilizadores e verificação de integridade das mensagens, cumprindo assim os requisitos de segurança fundamentais de um sistema de mensagens P2P.

Novas funcionalidades implementadas:

Para além das funcionalidades implementadas na primeira entrega foi adicionada a funcionalidade de pesquisa de mensagens e, para além disso, as mensagens deixaram de ser guardadas localmente e passaram a ser guardadas na cloud. Para isto criamos 4 replicas na AWS, 4 na Firebase e 4 na Microsoft. Abaixo está uma imagem do frontend com esta nova funcionalidade implementada.

Figura 1 - Funcionalidade de pesquisa de mensagens



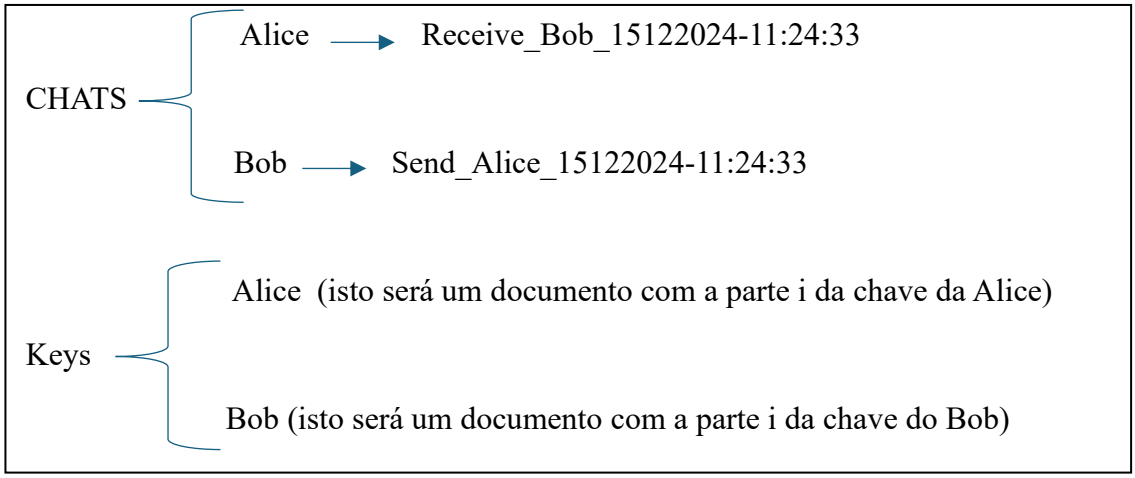
Para além desta funcionalidade decidimos também implementar um comando que permite limpar todas as mensagens que existam na cloud dum certo user (tal comando está explicado no documento HTR.txt enviado no zip da entrega). Achamos esta funcionalidade interessante, visto que o professor não tem acesso à nossa conta da cloud e, desta maneira será mais fácil de testar o nosso projeto.

Estrutura da Cloud:

Imaginemos o seguinte exemplo em que o user Bob envia uma mensagem à Alice no dia 15/12/2024 às 11:24:33 e que a cloud nesse momento estava vazia. O conteúdo que estará na réplica i será o seguinte:

Réplica-i:

Nome da réplica i:



Garantias de segurança oferecidas:

Para garantir a segurança nas comunicações, utilizamos SSL Sockets para estabelecer conexões diretas e encriptadas entre utilizadores, permitindo que as mensagens trocadas sejam protegidas contra intercepções. Cada utilizador (peer) cria um SSL Socket para comunicar diretamente com outro utilizador, garantindo assim a privacidade dos dados enviados e recebidos e uma interação descentralizada, além disso no momento do login cada user cria uma keyStore para guardar a sua chave pública e certificado e uma trustStore para guardar os certificados de outros user que considere confiáveis. Para se guardar os dados de cada user criou-se um servidor que consegue comunicar com cada user, de maneira a fornecer informações de outros utilizadores que estejam ativos e, para além disso, gera também um ficheiro com o seu próprio certificado.

A cipher suite utilizada na aplicação é a TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, configurada com o protocolo TLSv1.2. Ela combina o algoritmo ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) para troca de chaves segura, garantindo a criação de uma chave secreta compartilhada; RSA (Rivest-Shamir-Adleman) para autenticação das partes, assegurando que cliente e servidor (ou cliente e cliente) sejam confiáveis; AES-256 em modo GCM para criptografia eficiente dos dados, oferecendo também verificação de integridade; e SHA-384 (Secure Hash Algorithm) para garantir que as mensagens trocadas não sejam alteradas.

Em termos de segurança na cloud, visto que os providers não são confiáveis, decidimos criar uma chave por cada user, que será responsável pela encriptação (no caso de enviar algo para a cloud) e desencriptação (no caso de ir buscar algo à cloud). A chave foi criada utilizando o algoritmo AES (Advanced Encryption Standard). Este algoritmo opera em blocos de dados e permite diferentes tamanhos de chave (128, 192 ou 256 bits), sendo que neste caso optámos por uma chave de 256 bits. Para além da criação da chave o utilizador divide a mesma em 12 partes e envia cada parte para uma réplica em específico. Deste modo, cada réplica contém apenas uma parte da chave (1/12). Quando um user volta a entrar na app (assumindo que já esteve

lá antes e enviou mensagens), este fará a recuperação da chave criada anteriormente com apenas 8 partes recuperadas, pois através do algoritmo de Shamir podemos definir em quantas partes dividimos a chave e quantas partes precisamos para a recuperar totalmente. Para o nosso projeto decidimos dividir em 12 partes (visto que existem 12 réplicas) e recuperá-la em apenas 8, conseguindo assim implementar secret sharing.

Benefícios de Segurança:

● Privacidade e Confidencialidade: A utilização do algoritmo AES-256 em modo GCM para criptografia assegura que todos os dados trocados entre os peers sejam encriptados, garantindo que apenas as partes envolvidas possam acessar o conteúdo das mensagens.

● Autenticidade: O uso do algoritmo RSA para autenticação, juntamente com a verificação de certificados via KeyStore e TrustStore, garante que os peers estão a comunicar com entidades confiáveis, evitando interações com agentes maliciosos.

● Integridade dos Dados: O modo GCM de operação do AES, aliado ao hash seguro SHA-384, protege contra alterações nos dados durante a transmissão, assegurando que qualquer tentativa de modificação seja detetada.

● Privacidade dos Dados na cloud – Como todas as mensagens enviadas pelos utilizadores da app para a cloud são encriptadas, então os seus dados vão estar seguros vistos que só o user que enviou a mensagem possui tal chave, logo será o único a conseguir desencriptar as mesmas.

Conclusão:

O projeto desenvolvido demonstra uma abordagem robusta e de acordo com o pedido para a implementação de uma aplicação de mensagens peer-to-peer (P2P) que prioriza a privacidade, a segurança e a descentralização. A utilização de SSL Sockets e da cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 assegurou comunicações seguras, oferecendo confidencialidade, autenticidade e integridade nas mensagens trocadas. Além disso, a implementação de mecanismos como o armazenamento de dados na cloud com a encriptação AES-256, combinado com o uso de secret sharing, proporcionou uma camada adicional de segurança para as informações sensíveis dos utilizadores.

As funcionalidades implementadas, como a pesquisa de mensagens e a capacidade de limpar dados de um utilizador na cloud, reforçam a utilidade e a usabilidade da aplicação. Ao longo do projeto, destacamos o compromisso com boas práticas de segurança, garantindo que tanto a comunicação quanto o armazenamento de dados sejam protegidos contra ataques ou acessos não autorizados.

Concluimos assim que este projeto atendeu aos objetivos iniciais, de criação de uma aplicação para comunicação de mensagens descentralizada e segura.