Criptografia de Senha

Tatiane Corrêa

MÉTODOS E METODOLOGIA

- **Bouncy Castle** (org.bouncycastle) como provedor de criptografia. Bouncy Castle é uma biblioteca Java para criptografia que suporta uma ampla gama de algoritmos e protocolos.

- **PBKDF2** (Password-Based Key Derivation Function 2) para gerar uma chave a partir de uma senha e um salt. É frequentemente utilizado para hashing de senhas

MÉTODOS E METODOLOGIA

- **AES (Criptografia Simétrica):** Métodos como generateAESKey(), encryptAES() e decryptAES() lidam com a geração de chaves e criptografia/decriptografia usando o algoritmo AES. AES é um padrão de criptografia simétrica, o que significa que a mesma chave é usada tanto para criptografar quanto para descriptografar os dados.

- **RSA (Criptografia Assimétrica):** Métodos como generateRSAKeyPair(), encryptRSA() e decryptRSA() lidam com operações RSA. RSA é um algoritmo de criptografia assimétrica, onde uma chave pública é usada para criptografar os dados e uma chave privada correspondente é usada para descriptografar.

PBKDF2

Benefícios:

- Derivação Segura de Chaves: PBKDF2 foi projetado para transformar senhas, que muitas vezes são de baixa entropia, em chaves criptográficas seguras.
- Resistente a Ataques de Força Bruta: Devido ao uso de iterações, é mais resistente a ataques de força bruta em comparação com funções de hash simples.

Desafios:

- Velocidade: PBKDF2 é deliberadamente lento para torná-lo resistente a ataques, mas isso pode ser um desafio quando usado em cenários onde o desempenho é uma preocupação.
- Possíveis Ataques Futuros: Embora seja seguro atualmente, as técnicas de ataque continuam evoluindo e podem eventualmente encontrar vulnerabilidades.

AES (Criptografia Simétrica)

Benefícios:

- Desempenho Rápido: O AES é eficiente e pode criptografar ou descriptografar grandes volumes de dados rapidamente.
- Padrão Amplamente Aceito: É o padrão de criptografia simétrica adotado pelo NIST (National Institute of Standards and Technology) e é usado globalmente.

Desafios:

- Gestão de Chaves: A chave usada para criptografar é a mesma usada para descriptografar. Portanto, o compartilhamento seguro dessa chave com partes interessadas é um desafio.
- Não Autentica Dados: O AES padrão não garante a integridade ou autenticidade dos dados. Geralmente é combinado com modos de operação ou técnicas adicionais (como HMAC) para garantir a autenticidade.

RSA (Criptografia Assimétrica)

Benefícios:

- Gestão de Chaves Simplificada: Não é necessário compartilhar a chave privada. Apenas a chave pública é distribuída, e a chave privada permanece segura.

- Flexibilidade: Pode ser usado para criptografia de dados, bem como para autenticação (por exemplo, assinaturas digitais).

- Confiabilidade: O RSA tem resistido ao teste do tempo em termos de segurança, desde que chaves de tamanho suficiente sejam usadas.

RSA (Criptografia Assimétrica)

Desafios:

- Desempenho: Criptografar dados diretamente com RSA é muito mais lento do que com algoritmos simétricos. Isso é uma desvantagem quando se trata de grandes volumes de dados.

- Limitação do Tamanho de Dados: O RSA só pode criptografar dados até um certo limite de tamanho, geralmente menor do que o tamanho da chave.

- Gestão de Chaves: Embora seja mais fácil em termos de compartilhamento, gerar, armazenar e renovar pares de chaves pode ser complexo.

Autenticação

Registro de Usuário:

- Quando um usuário se registra:
- Uma chave AES é gerada.
- A senha do usuário é criptografada usando esta chave AES.
- Um par de chaves RSA (pública e privada) é gerado.
- A chave AES é criptografada usando a chave pública RSA.
- As chaves RSA (pública e privada) e a senha criptografada são armazenadas no banco de dados após serem codificadas em Base64.

Autenticação

Processo de Login:

- O usuário fornece um nome de usuário e senha.
- O sistema busca no banco de dados pelo nome de usuário. Se não encontrar, a autenticação falha.
- Se o usuário for encontrado:
- A chave privada RSA do usuário é decodificada do Base64 e convertida de volta para um objeto PrivateKey.
- A chave AES criptografada é decifrada usando a chave privada RSA, retornando a chave AES original.
- A senha criptografada é então decifrada usando esta chave AES, retornando a senha original.
- A senha original é comparada com a senha fornecida pelo usuário durante o login. Se elas coincidirem, a autenticação é bem-sucedida.

Estrutura do banco de dados

- encrypted_password: Em vez de armazenar a senha do usuário em texto simples, o sistema armazena uma versão criptografada da senha.
- public_key: Usada para criptografar a chave AES.
- Private_key: Usada para descriptografar a chave AES, permitindo assim que a senha do usuário seja descriptografada.
- encryptedAESKey: A chave AES é criptografada usando a chave pública RSA do usuário, garantindo que apenas o detentor da chave privada correspondente possa descriptografá-la.

Resumo:

A autenticação é baseada em uma combinação de criptografia simétrica (AES) e criptografia assimétrica (RSA). A senha é inicialmente criptografada com AES. A chave AES é então criptografada com RSA. Durante o processo de login, essa chave AES criptografada é decifrada usando a chave privada RSA, e a senha criptografada é decifrada usando a chave AES recuperada para verificar a autenticidade do usuário. Se tudo estiver correto, o usuário é autenticado.

Referências

- https://www.bouncycastle.org/
- https://blog.mailfence.com/pt/criptografia-simetrica-x-assimetrica-qual-e-a-diferenca/
- https://www.baeldung.com/java-bouncy-castle