Aplicação Criptografia

Trabalho prático 1

Criptografia e segurança de sistemas informáticos

Nome: Vasco Rodrigues Ribeiro

Número: al70638

**Introdução:**

Neste relatório, analisamos a aplicação desenvolvida para criptografar e descriptografar mensagens e arquivos, além de assinar digitalmente mensagens para garantir a sua autenticidade. A aplicação utiliza diferentes técnicas e algoritmos de criptografia, incluindo criptografia simétrica, criptografia assimétrica e funções de hash. As principais características, vantagens e desvantagens de cada algoritmo serão apresentadas a seguir.

**Criptografia Simétrica:**

A aplicação utiliza o algoritmo Advanced Encryption Standard (AES) para criptografia simétrica. O AES é um algoritmo de criptografia de bloco que usa a mesma chave para criptografar e descriptografar os dados.

**Vantagens:**

* Velocidade: O AES é mais rápido do que os algoritmos de criptografia assimétrica.
* Eficiência: O AES é eficiente em termos de recursos, tornando-o adequado para dispositivos com capacidade de processamento limitada.

**Desvantagens:**

* Distribuição de chaves: A maior desvantagem do AES é a necessidade de compartilhar a chave secreta entre as partes envolvidas na comunicação segura. A distribuição segura da chave é um desafio.

**Criptografia Assimétrica:**

A aplicação utiliza o algoritmo RSA para criptografia assimétrica. O RSA é um algoritmo de criptografia de chave pública que usa um par de chaves (uma chave pública e uma chave privada) para criptografar e descriptografar os dados.

**Vantagens:**

* Segurança na distribuição de chaves: A chave pública pode ser compartilhada abertamente sem comprometer a segurança dos dados criptografados.
* Autenticação: O RSA também pode ser usado para autenticação, pois apenas o detentor da chave privada correspondente pode descriptografar os dados criptografados com a chave pública.

**Desvantagens:**

* Velocidade: O RSA é mais lento do que os algoritmos de criptografia simétrica.
* Tamanho da chave: O RSA requer chaves de maior tamanho para fornecer o mesmo nível de segurança que os algoritmos simétricos.

**Funções do Hash:**

A aplicação permite ao utilizador gerar hashes de mensagens usando o algoritmo de hash seguro (SHA) na versão SHA-256. As funções de hash são usadas para criar uma representação de tamanho fixo de um conjunto de dados, que é praticamente única para cada conjunto de dados.

**Vantagens:**

* Integridade dos dados: As funções de hash podem ser usadas para verificar a integridade dos dados, pois qualquer alteração nos dados resulta em uma alteração no hash.
* Eficiência: As funções de hash são geralmente rápidas e eficientes em termos de recursos.

**Desvantagens:**

* Unidirecionalidade: As funções de hash são projetadas para serem unidirecionais, o que significa que não é possível recuperar os dados originais a partir do hash.

**Assinatura Digital:**

A aplicação utiliza o algoritmo RSA em conjunto com o algoritmo de hash SHA-256 para assinar digitalmente mensagens. Isso garante a autenticidade, integridade e não repúdio das mensagens trocadas entre as partes.

**Vantagens:**

* Autenticidade: A assinatura digital confirma a identidade do remetente, garantindo que a mensagem seja originada de uma fonte autêntica.
* Integridade: A assinatura digital garante que a mensagem não foi alterada durante o trânsito, garantindo a integridade dos dados.
* Não repúdio: A assinatura digital impede que o remetente negue ter enviado a mensagem, pois apenas o detentor da chave privada correspondente pode criar uma assinatura válida.

**Desvantagens:**

* Complexidade: A assinatura digital envolve o uso de algoritmos de criptografia assimétrica e funções de hash, aumentando a complexidade da implementação em comparação com a criptografia simétrica.
* Desempenho: A assinatura digital pode ser mais lenta do que a criptografia simétrica devido à complexidade dos algoritmos de criptografia assimétrica e funções de hash envolvidos.

**Documentação de funcionalidades da aplicação:**

Como documentação das funcionalidades da aplicação, existe a classe "**CryptographyMethods**" responsável por executar todas as funções de encriptação,desencriptação e verificação de dados. De forma sucinta, esta classe servirá como uma classe model onde serão armazenadas todas as operações que se realizarão ao longo da utilização da app.



* Este método "hashText" tem como função realizar o hashing de uma string (como parâmetro de entrada) e através do algoritmo SHA-256 irá criar um digest que posteriormente será usado para mostrar na UI da app.



* Este método terá como função de gerar uma chave simétrica para encriptação de dados de forma simétrica. A chave pública gerada neste método será responsável por encriptar os dados.



* Este método terá como função gerar um vetor de inicialização para ser usado no algoritmo de encriptação simétrica CBC (Advanced Encryption Standart) na operação XOR do primeiro byte da string a encriptar.



* Este método terá como função gerar um par de chaves que é utilizado para algoritmos de encriptação asimétrica. Este método retorna uma chave privada e uma chave pública.



* Este método terá como função encriptar uma string de forma simétrica usando a chave secreta gerada e o vetor de inicialização.



* Este método terá como função desencriptar dados de forma simétrica usando a chave secreta usada para encriptar os dados e o vetor de inicialização.



* Este método terá como função verificar o formato do texto que o utilizador introduza na UI. Caso seja base64 continuará com uso da aplicação,caso contrário avisa o utilizador que o input está num formato incorreto



* Este método terá como função desencriptar dados de forma asimétrica recebendo texto cifrado (de forma asimétrica) e desencriptando com a chave privada, trazendo confidencialidade á aplicação.



* Este método terá como função realizar assinatura digital de texto usando uma chave privada (chave privada do par de chaves gerado) e retornar uma string assinada (num array de bytes).



* Este método terá função de verificar a assinatura feita, ou seja, irá verificar a autencidade da assinatura, se o texto que vem assinado vem da origem (quem assina) correta e verifica se o conteúdo foi manipulado ou não através do hashing, trazendo também integridade á app.



* Este método terá função de juntar 2 ou mais arquivos escolhidos pelo utilizador. Este método recebe os dois arquivos e copia o conteúdo de cada um para um terceiro arquivo.Este terceiro arquivo terá formato ".tar" com o nome "output\_joined.tar" localizado na diretoria da aplicação.

**Nota:** O utilizador poderá juntar mais que dois arquivos para o ficheiro "output\_joined.tar" mas apenas poderá fazer de 2 em 2 arquivos selecionados através da componente JfileChooser presente na interface gráfica.

A classe "CriptografiaApp" servirá como View e controller simultâneamente onde irá ser implementada a interface gráfica da aplicação com as componentes disponibilizadas no java (Jbutton,JtextField,JLabel,Jpanel,Jframe, etc...). Os métodos definidos na classe Model irão ser chamados nos actionListeners dos botões existentes na aplicação, que irão ser tratados em blocos try catch para tratamento de exceções.

**Output dos métodos:**

Como forma de debug, o resultado dos métodos de encriptação de dados (hash,encriptação simétrica e asimétrica (AES,CBC,RSA,etc...) são mostrados também na consola do projeto).

**Conclusão:**

A aplicação desenvolvida utiliza diferentes técnicas e algoritmos de criptografia para fornecer funcionalidades de criptografia simétrica e assimétrica, hashing e assinatura digital. Essas técnicas e algoritmos oferecem vantagens em termos de segurança, autenticidade e integridade dos dados. No entanto, também apresentam desafios em termos de distribuição de chaves, desempenho e complexidade.Temos que saber qual algoritmos de encriptação de dados temos que usar consoante os recursos presentes no sistema (se está a ser implementado num sistema que requer um elevado nível de segurança e rapidez,o poder computacional das máquinas que realizam a encriptação e desencriptação, etc...). O uso de algoritmos e técnicas apropriadas para cada cenário específico é crucial para garantir a segurança e a eficiência das comunicações seguras.